Моё разочарование в софте

Войти Регистрация
http://tonsky.me/blog/disenchantment/

Разработка под Android,
Разработка под Linux,
Разработка под Windows,
Софт

Перевод



Суть разработки программного обеспечения
— Нужно проделать 500 отверстий в стене, так что я сконструировал автоматическую дрель. В ней используются элегантные точные шестерни для непрерывной регулировки скорости и крутящего момента по мере необходимости.
— Отлично, у неё идеальный вес. Загрузим 500 таких дрелей в пушку, которые мы сделали, и выстрелим в стену.

Я занимаюсь программированием уже 15 лет. Но в последнее время при разработке не принято думать об эффективности, простоте и совершенстве: вплоть до того, что мне становится грустно за свою карьеру и за IT-отрасль в целом.

Для примера, современные автомобили работают, скажем, на 98% от того, что физически позволяет нынешняя конструкция двигателя. Современная архитектура использует точно рассчитанное количество материала, чтобы выполнять свою функцию и оставаться в безопасности в данных условиях. Все самолёты сошлись к оптимальному размеру/форме/нагрузке и в основном выглядят одинаково.

Только в программном обеспечении считается нормальным, если программа работает на уровне 1% или даже 0,01% от возможной производительности. Ни у кого вроде нет возражений. Люди даже гордятся, насколько неэффективно работает программа, типа «зачем беспокоиться, компьютеры достаточно быстрые»:

> @tveastman: Я каждый день запускаю программу на Python, она выполняется за 1,5 секунды. Я потратил шесть часов и переписал её на Rust, теперь она выполняется за 0,06 секунды. Это ускорение означает, что моё время окупится через 41 год, 24 дня :-)

Наверное, вы слышали такую мантру: «Время программиста дороже времени компьютера». Это означает, что мы тратим компьютерное время в беспрецедентных масштабах. Вы бы купили машину с расходом 100 литров на 100 километров? Как насчёт 1000 литров? С компьютерами такое происходит постоянно.

Всё невыносимо медленно

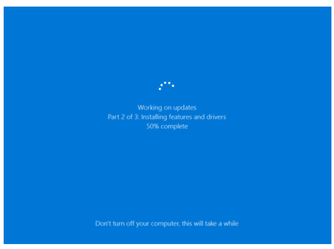
Оглянитесь вокруг: портативные компьютеры в тысячи раз мощнее тех, что привели человека на Луну. Тем не менее, каждый второй сайт не может обеспечить плавную прокрутку страницы на 60 FPS на последнем топовом MacBook Pro. Я могу комфортно играть в игры, смотреть видео 4K, но не прокручивать веб-страницы! Это нормально?

Почтовому приложению Google Inbox в браузере Chrome от той же Google, требуется 13 секунд, чтобы открыть письмо среднего размера:

> This is, in real time, how long it takes for Google Inbox running in Google Browser to open an email. Not the shortest one, but still, it's just text and pictures! Go Web Stack go! pic.twitter.com/CvqsFilUuc
> — Nikita (@nikitonsky) February 28, 2018

Он ещё анимирует пустые белые формы вместо того, чтобы показать их содержимое, потому что это единственный способ анимировать что-то на веб-странице с приличной производительностью. Нет, не 60 FPS, а скорее «настолько быстро, насколько возможно на этой странице». С нетерпением жду, что же веб-сообщество предложит, когда дисплеи 120 Гц станут мейнстримом. Они еле справляются с 60 Гц.

Обновление Windows 10 занимает 30 минут. Что можно делать так долго? Этого времени достаточно, чтобы полностью отформатировать мой SSD-накопитель, загрузить свежий билд и установить его примерно 5 раз подряд.



> Павел Фатин: Набор текста в редакторе — относительно простой процесс, поэтому даже 286 могли обеспечить довольно плавный процесс набора.

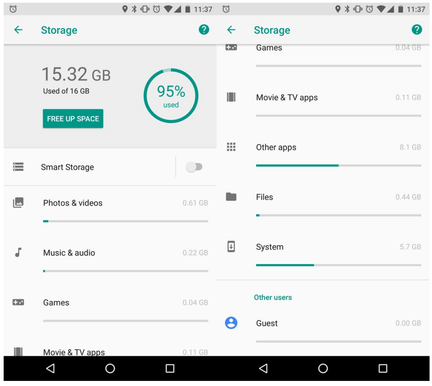
В современных текстовых редакторах задержка при наборе больше, чем в 42-летнем Emacs. Текстовые редакторы! Что может быть проще? На каждое нажатие клавиши, нужно всего лишь обновить крошечную прямоугольную область на экране, а современные текстовые редакторы не могут сделать это за 16 мс. А это много времени. МНОГО. 3D-игра заполняет экран сотнями тысяч (!!!) полигонов за те же 16 мс, а также обрабатывает ввод, пересчитывает мир и динамически загружает/выгружает ресурсы. Как так?

Тенденция такова, что софт вовсе не становится быстрее и функциональнее. Мы получаем более быстрое оборудование, на котором софт с теми же функциями ворочается медленнее, чем раньше. Всё работает намного медленнее максимальной скорости. Никогда не задумывались, почему ваш телефон загружается от 30 до 60 секунд? Почему он не может загрузиться, скажем, за одну секунду? Здесь нет никаких физических ограничений. Лично мне бы такое понравилось. Хочется, чтобы разработчики достигли предела, используя каждый бит для производительности.

Всё ОГРОМНОЕ

И ещё это раздутие. Веб-приложения могут открываться в десять раз быстрее, если просто заблокировать рекламу. Google умоляет всех прекратить тормоза с помощью инициативы AMP — технического решения, для которого не нужны какие-либо технологии, просто немного здравого смысла. Если удалить раздувание, интернет станет работать на сумасшедшей скорости. Неужели это сложно понять?

Система Android без приложений занимает почти 6 ГБ. Просто задумайтесь на секунду, насколько неприлично огромное это число. Что там, фильмы в HD-качестве? Думаю, в основном код: ядро, драйверы. Ещё какие-то ресурсы, конечно, но они не могут быть такими большими. Сколько же драйверов вам нужно для телефона?

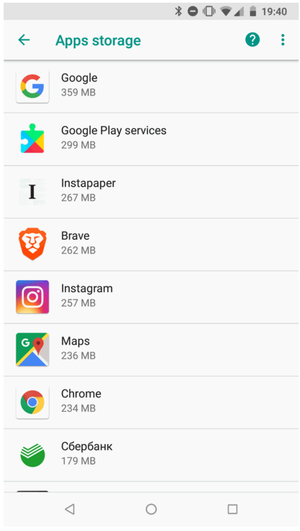


Windows 95 занимала 30 МБ. Сегодня у нас есть веб-страницы тяжелее, чем эта ОС! Windows 10 уже 4 ГБ, то есть в 133 раза больше. Но разве она в 133 раза лучше? Я имею в виду, функционально они практически одинаковы. Да, у нас появилась Кортана, но я сомневаюсь, что она весит 3970 МБ. Но это Windows 10, неужели Android должен быть ещё в полтора раза больше?

Приложение клавиатуры Google как ни в чём не бывало съедает 150 МБ. Эта программа рисует 30 клавиш на экране — она правда в пять раз сложнее, чем вся Windows 95? Приложение Google app, в основном, просто пакет для Google Web Search, занимает 350 МБ! Сервисы Google Play, которыми я не пользуюсь (я не покупаю там книги, музыку или видео) — 300 МБ, которые просто сидят здесь и которые нельзя удалить.

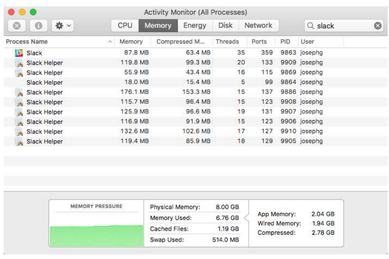
Всё, реклама кончилась.
Покажите что-нибудь

ЧИТАЮТ СЕЙЧАС
Senior Engineer в поисках работы. Как я прошел 20 собеседований с HR и что я об этом думаю
5,2k 22
Китайское искусственное солнце...
17,5k 43
Суд обязал Роскомнадзор заблокировать «пиратские» научные порталы на территории России
23,6k 154
Почему «Юному технику» не удастся построить лазер
34,7k 112
Собираем собственный аудио усилитель профессионального уровня не (очень) дорого
11,9k 33
NASA объявило о новом полете на Луну
2,7k 9



После установки всех необходимых приложений (социальные сети, чаты, карты, такси, банки и т. д.) на телефоне остался всего 1 гигабайт для фотографий. И это вообще без игр и музыки! Помните времена, когда ОС, приложения и все ваши данные помещались на дискету?

Ваша программа для заметок наверняка написана в Electron и, таким образом, поставляется с драйвером для контроллера Xbox 360, умеет показывать 3D-графику, воспроизводить аудио и фотографировать с помощью веб-камеры.



Простой текстовый чат всегда славился скоростью и малым потреблением памяти. Так что Slack — это пример очень ресурсоёмкого приложения. Я имею в виду, что чат и текстовый редактор — это самые базовые вещи, они должны потреблять меньше всего ресурсов. Добро пожаловать в 2018 год.

Вы можете сказать, что они хотя бы работают. Но увеличение размера — не значит улучшение. Это значит, что кто-то потерял контроль. Мы больше не знаем, что происходит. Увеличение размера — это повышение сложности, снижение производительности и надёжности. Это ненормально и не должно считаться нормой. На раздутый размер нужно сразу обращать внимание — и держаться от них подальше.

Всё гниёт

Android-телефон на 16 ГБ был прекрасен три года назад. Сегодня под Android 8.1 он еле работает, потому что каждое приложение увеличилось минимум вдвое без видимых причин. Дополнительных функций нет. Они не стали быстрее и внешний вид не изменился. Они просто… раздулись?

iPhone 4s вышел с iOS 5, но едва может работать под управлением iOS 9. И это не потому, что iOS 9 намного лучше — в основном, система не изменилась. Но новое оборудование быстрее, поэтому они сделали программное обеспечение медленнее. Не волнуйтесь — вы получили захватывающие новые возможности, например… работа тех же приложений с той же скоростью! Не знаю.

iOS 11 прекратила поддержку 32-разрядных приложений. Это значит, что если разработчик не готов вернуться и обновить приложение, скорее всего, вы не увидите снова эту отличную программу.

> @jckarter: Программу DOS можно заставить работать без изменений практически на любом компьютере, сделанном после 80-х годов. Приложение JavaScript может прекратить работу из-за завтрашнего обновления Chrome.

Сегодняшние веб-страницы не будут работать в любом браузере через 10 лет (а может и раньше).

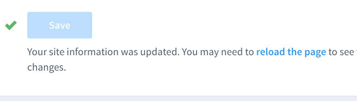
«Нужно бежать со всех ног, чтобы только остаться на том же месте». Но смысл? Я могу постоянно покупать новые телефоны и ноутбуки, как все, но делать это лишь ради того, чтобы иметь возможность запускать все те же приложения, которые стали только медленнее?

Думаю, что мы можем и должны исправить ситуацию. Сейчас все разрабатывают программы для сегодняшнего дня, изредка для завтрашнего. Но будет неплохо делать вещи, которые работают немного дольше.

Хуже — значит лучше

Сейчас никто ничего не понимает. И не хочет понимать. Мы просто выпускаем полусырую ерунду, надеемся на лучшее и называем это «здравым смыслом для стартапа».

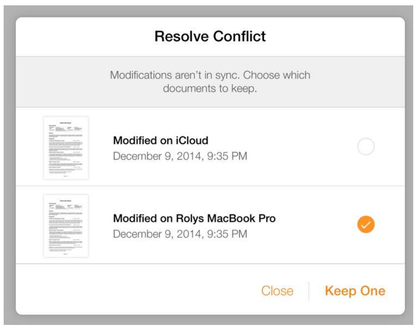
Веб-страницы просят обновиться, если что-то пошло не так. У кого есть время, чтобы найти причину неполадки?



Любое веб-приложение выдаёт постоянный поток «случайных» ошибок JS, даже на совместимых браузерах.

Вся архитектура баз данных веб/SQL построена на предпосылке (даже надежде), что никто не изменит данные, пока вы смотрите на открытую веб-страницу.

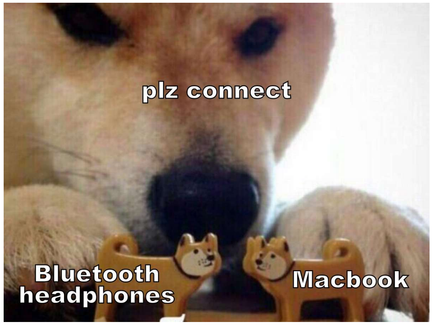
Большинство приложений для совместной работы сделали «как смогли», там масса типичных сценариев, когда они теряют данные. Видели диалог «Какую версию сохранить?» Сегодня планка так низка, что пользователи рады даже этому вопросу.



И нет, в моём мире не является нормальным приложение, которое говорит: «Я уничтожу часть твоей работы, только выбери какую».

Linux *намеренно* убивает случайные процессы. И всё же это самая популярная серверная ОС.

У меня каждое устройство регулярно выходит из строя так или иначе. Время от времени монитор Dell нужно аппаратно перезагружать, потому что в нём есть софт. AirDrop? Вам повезёт, если он обнаружит устройство, иначе что делать? Bluetooth? Спецификации настолько сложны, что устройства не будут устанавливать связь друг с другом, а периодические перезагрузки — оптимальный вариант.



И я даже не упоминаю об Интернете вещей. Это настолько за гранью разумного, что даже нечего добавить.

Я хочу гордиться своей работой. Я хочу делать рабочие, стабильные вещи. Для этого нужно понять, что конкретно мы разрабатываем, внутри и снаружи, а это невозможно сделать в раздутых, чрезмерно усложнённых системах.

В программировании такой же хаос

Кажется, что никто больше не заинтересован в качественных, быстрых, эффективных, долговечных, основательных решениях. Даже если давно известны эффективные решения, мы по-прежнему боремся с теми же проблемами: управление пакетами, системы сборки, компиляторы, конструкция языка, IDE.

Системы сборки по своей сути ненадёжны и периодически требуют полной очистки, хотя у них есть вся информация для инвалидации. Ничто не мешает сделать процесс сборки надёжным, предсказуемым и на 100% воспроизводимым. Просто никто не думает, что это важно. NPM уже много лет находится в состоянии «иногда работает».

> @przemyslawdabek: Кажется, что rm-rf node_modules является неотъемлемой частью рабочего процесса в проектах Node.js/JavaScript.

А время сборки? Никто не считает проблемой, что компилятор работает минуты или даже часы. А как же «время программиста дороже»? Почти все компиляторы, пре- и постпроцессоры значительно, иногда катастрофически увеличивают время сборки, не обеспечивая пропорционально существенных преимуществ.

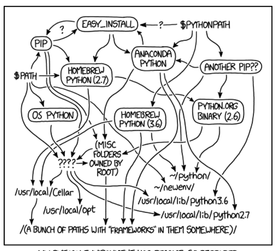


Вы ожидаете, что программисты будут принимать в основном рациональные решения, но иногда они делают прямо противоположное. Например, выбирая Hadoop даже если он медленнее, чем выполнение той же задачи на одном десктопном компьютере.

Машинное обучение и ИИ отбросили программное обеспечение к гаданию на кофейной гуще во времена, когда большинство компьютеров даже не были достаточно надёжными.

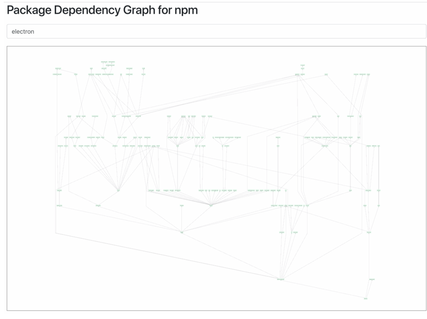
> @rakhim: Когда приложение или сервис говорит «под управлением ИИ» или «на основе машинного обучения», я читаю это как «ненадёжное, непредсказуемое поведение, которое не поддаётся объяснению». Я держусь подальше от «ИИ», потому что хочу от компьютеров противоположного: надёжности, предсказуемости и логики.

Мы засунули виртуальные машины в Linux, а затем засунули Docker в виртуальные машины, просто потому что никто не смог разобраться с бардаком, который производят большинство программ, языков и их окружений. Мы накрываем дерьмо одеялами, чтобы не убирать его. Например, «единый бинарник» остаётся ОГРОМНЫМ преимуществом Go. Нет бардака == успех.



Окружающая среда Python настолько загрязнилась, что мой ноутбук объявили зоной экологической катастрофы.
Примечание. Агентство по защите окружающей среды Python хотело залить его цементом и захоронить с картинкой на входе — предупреждением для будущих цивилизаций об опасности использовать sudo для установки случайных пакетов

А зависимости? Люди бездумно ставят переусложнённые «полные пакеты» для простейших проблем, не думая о последствиях. Из этих зависимостей растут новые. В конечном итоге вы получаете дерево, которое является чем-то средним между фильмом ужасов (огромное и полное конфликтов) и комедией (нет причин, по которым мы добавили сюда эти пакеты, но вот они):



Программы не могут работать несколько лет без перезагрузки. Иногда даже несколько дней — это слишком. Происходят случайные *г*люки, и никто не знает почему.

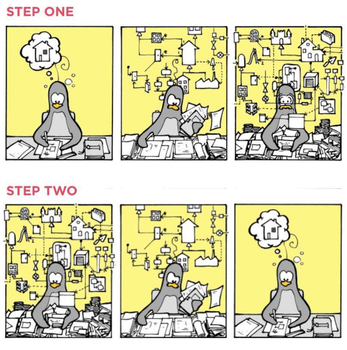
Что ещё хуже, ни у кого нет времени остановиться и выяснить, что произошло. Зачем беспокоиться, если всегда есть другой выход. Поднять новый инстанс AWS. Перезапустить процесс. Удалить и восстановить базу данных. Написать скрипт, который будет перезапускать ваше сломанное приложение каждые 20 минут. Включить одни и те же ресурсы несколько раз: тяп-ляп — и в продакшн. Двигайся быстро, не трать время на исправление ошибок.

Это не инженерная работа. Это просто ленивое программирование. Инженерная работа предполагает глубокое понимание производительности, структуры и ограничений того, что вы создаёте. Лепить халтуру из некачественного материала — совершенно противоположное занятие. Чтобы развиваться, мы должны понимать, что и зачем мы делаем.

Мы застряли

Таким образом, всё это просто куча едва работающего кода, добавленного поверх ранее написанного едва работающего кода. Он продолжает расти в размерах и сложности, уменьшая шансы на изменения.

Чтобы иметь здоровую экосистему, *необходимо* вернуться. *Необходимо* иногда выбрасывать хлам и заменять его лучшими альтернативами.



LUKEW IDEATION + DESIGN Image source: https://bit.ly/2wBCD66

Но у кого есть на это время? Новые ядра ОС не выходили сколько, 25 лет? Это сейчас стало слишком сложным, чтобы просто взять и переписать. В браузерах накопилось столько пограничных ситуаций и исторических прецедентов, что никто не осмелится писать движок с нуля.

Сегодняшнее определение прогресса — или подбросить топлива:

> @saltrizv: 2014 — нужно внедрить микросервисы для решения проблем с монолитами.
> 2016 — нужно внедрить Docker, чтобы решить проблемы с микросервисами.
> 2018 — нужно внедрить Kubernetes, чтобы решить проблемы с Docker.

или изобретать велосипед:

> @dr_c0d3: 2000: напишите 100 строк XML, чтобы «декларативно» настроить сервлеты и EJB.
> 2018: напишите 100 строк YAML, чтобы «декларативно» настроить микросервисы.
> В XML были хотя бы схемы…

Мы застряли, и никто нас не спасёт.

Бизнесу всё равно

Пользователям тоже. Они выучились принимать то, что мы делаем. Мы (инженеры) говорим, что каждое приложение для Android занимает 350 МБ? Хорошо, они будут с этим жить. Мы говорим, что не можем обеспечить плавную прокрутку? Окей, они свыкнутся с телефоном, который подтормаживает. Мы говорим: «Если не работает, перезагрузитесь»? Они перезагрузятся. Ведь у них нет выбора.

Конкуренции тоже нет. Все строят одни и те же медленные, раздутые, ненадёжные продукты. Случайный скачок вперёд по качеству даёт конкурентное преимущество (iPhone/iOS против других смартфонов, Chrome против других браузеров) и заставляет всех перегруппировываться, но ненадолго.

Наша миссия как инженеров — показать миру потрясающие возможности современных компьютеров с точки зрения производительности, надёжности, качества и удобства использования. Если нам не всё равно, люди потянутся. И никто кроме нас не покажет им, что такое возможно. Если только нам не наплевать.

Не всё так плохо

Иногда на пасмурном небосводе просвечивают лучики надежды.

Работа Мартина Томпсона (LMAX Disruptor, SBE, Aeron) впечатляет, она освежающе проста и эффективна.

Редактор Xi Рафа Левиена, кажется, построен на правильных принципах.

Джонатан Блоу для своей игры разработал язык компилирования, который компилирует 500 000 строк в секунду на ноутбуке. Это холодная компиляция, никакого промежуточного кэширования, никаких инкрементальных билдов.

Не нужно быть гением, чтобы писать быстрые программы. Здесь нет какой-то магии. Единственное, что требуется, — это не строить софт на базе огромной кучи дерьма, которую поставляют современные инструменты.

Манифест лучшего мира

Я хочу видеть прогресс. Я хочу перемен. Чтобы современное программное обеспечение совершенствовалось, а не стояло на месте. Я не желаю заново изобретать одно и то же, каждый раз выпуская всё более медленный и раздутый продукт. Я хочу во что-то верить — в достойную цель, в будущее, которое лучше, чем то, что мы имеем сегодня, и чтобы появилось сообщество инженеров, которые разделяют это видение.

Что мы имеем сегодня — это не прогресс. Мы едва достигаем бизнес-целей с этими плохими инструментами. Мы застряли в локальном оптимуме, и никто не хочет двигаться. Это даже не хорошее место, оно раздутое и неэффективное. Мы просто как-то привыкли к нему.

Поэтому я хочу заявить: **нынешняя ситуация — полное дерьмо**. Как инженеры, мы можем и должны, и сделаем лучше. У нас могут быть лучшие инструменты, мы можем создавать лучшие приложения, более быстрые, предсказуемые, более надёжные, использующие меньше ресурсов (на порядки меньше!). Мы должны глубоко понять, что мы делаем и почему. Мы должны выпускать продукты надёжно, предсказуемо, с самым высоким качеством. Мы можем и должны гордиться нашей работой. Не просто «учитывая то, что у нас было…» — никаких оговорок!

Надеюсь, я не одинок. Надеюсь, что есть люди, которые хотят того же. Я буду рад, если мы хотя бы начнём говорить о том, насколько абсурдно нелепа нынешняя ситуация в индустрии программного обеспечения. А потом, возможно, придумаем, как выбраться из неё.

Tags:
индустрия ПО,
прогресс,
ожирение софта

+443
812
735,5
2792,8
587
@m1rko
Пользователь

Поделиться публикацией

ПОХОЖИЕ ПУБЛИКАЦИИ

24 января 2018 в 23:48
Видеозаписи выступлений с лекционного дня по игровой индустрии в ВШБИ
+18 2,9k 37 1

30 ноября 2017 в 22:02
Кому и насколько нужно поспевать за прогрессом?
+1 4,3k 15 34

30 августа 2017 в 13:02
Работа со сложным софтом: как провести миграцию кассовой программы и не «сломать» магазин
+4 1,7k 13 0

ЗАКАЗЫ Фрилансим

Корпоративный сайт под ключ
6 откликов • 47 просмотров
12000
за проект

Абонент по поддержке Android-приложения (удаленно)
13 откликов • 71 просмотр
800
за час

Разработка кроссплатформенной игры – политический симулятор
15 откликов • 125 просмотров
200000
за проект

Доработать электронную читалку под Android
16 откликов • 63 просмотра
25000
за проект

Разработка Android приложения
16 откликов • 135 просмотров
800
за час

Все заказы Разместить заказ

Комментарии 2432

vireshet 20 сентября 2018 в 15:07

+39

Это всё, конечно, правильно, но, мне интересно, готов ли автор (да, я вижу что это перевод) платить вместо условной тысячи долларов за новый телефон — платить 50 000? И тогда действительно расширят штат разработчиков, чтобы у них было время писать ядро системы отдельно под каждый телефон. А ещё придётся платить за каждый веб-сайт, потому что разрабатывать каждый раз всё с нуля, без использования библиотек и фреймворков — будет очень и очень дорого. Вот поэтому, там где оно надо (космос, например, или военка) — там и оптимизируют. А в реальном мире пользователю важнее получить приложение за 1$ которое будет работать и на его телефоне, и на планшете, и даже на телевизоре…

DrZlodberg 20 сентября 2018 в 15:17

+8

Всё хорошо в меру. Только что попробовали открыть XML файл в пару сотен метров (не спрашивайте зачем) в 17 VisualStudio под десяткой. На относительно свежем компе с 4 пнём (из какой именно) и 16ГБ оперативки. Через 15 минут грохнули не дождавшись его выхода из медитации.

sopov 20 сентября 2018 в 15:18

+4

А в FARe за сколько откроет?

lieff 20 сентября 2018 в 15:27

+19

Мгновенно, он умеет обрабатывать кусочками из произвольного места. Только colorer плагин будет в фоне с начала расцветку считать.

QtRoS 20 сентября 2018 в 22:48

−10

'кусочками' научно называется memory mapped files :)

crozzle 21 сентября 2018 в 17:14

+1

К словам придираетесь

QtRoS 21 сентября 2018 в 19:08

+1

Я не придираюсь — даю детали для потенциально заинтересовавшихся реализацией.

sasha1024 22 сентября 2018 в 02:22

+4

Мне кажется, memory mapped files тут мало при чём. Речь о том, что некоторые редакторы способны отображать файл, лишь распарсив его **целиком**, а некоторым достаточно распарсивания **некоторых определённых частей** (в т.ч. той, которая должна быть отображена в данный момент, т.к. находится в видимой на экране части документа). Memory mapped files тут в принципе можно применить, но, по-моему, не обязательно, обычных lseek/read вполне достаточно, разница скорее в алгоритмах, а не в lseek+read vs. mmap.

IRainman 22 сентября 2018 в 08:23

+1

Memory mapped files тут не просто можно, а обязательно ибо иначе будет больше вероятность лагов ;)

popov654 22 сентября 2018 в 09:09

+1

Мне кажется, или в случае, если весь файл влезает в RAM (точнее даже не так, если все данные, необходимые парсеру/просмотрщику влезают в RAM), то это совершенно лишнее? Исходя из определения

IRainman 22 сентября 2018 в 10:41

+2

Не лишнее, memory mapped files позволяют убрать лишнюю прослойку и сразу использовать данные из файлового кэша системы. При использовании обычного подхода получится, что система в любом случае откроет файл и он в любом случае будет в кеше, но в память приложения данные придётся копировать. и для @sasha1024 тоже.

sasha1024 22 сентября 2018 в 10:54

0

Я так понимаю, в кэше будет не обязательно весь файл, а может быть часть файла (если он большой).
А что значит «но в память приложения данные придётся копировать»?
Upd.: А, всё, я кажется, понял. При read происходит перенос данных в уже выделенную пользователем память. При этом эта память уже контроллируется приложением, т.е. оптимизировать это каким-то pe'map'ингом страниц (или как там ОС это делать может) не выйдет.

IRainman 22 сентября 2018 в 12:15

0

Агу, именно так. Кроме того остаётся пусть и небольшой но оверхед на работу самих API файловой системой, а тут для приложения сразу доступен уже очищенный от всего связанного с ФС кусок памяти содержащий данные файла.

sasha1024 22 сентября 2018 в 12:23

0

Ну, оверхед на работу с файловой системой будет в любом случае — разве нет?

IRainman 22 сентября 2018 в 12:26

0

Этот оверхед конечно останется, но вся эта работа и так производится. Просто тут очень удобно получается, что данные уже лежат в памяти и воспринимать их как файл нет никакой необходимости.

sasha1024 22 сентября 2018 в 12:28

0

Ну, на скорости это практически не сказывается.

maxzhurkin 7 октября 2018 в 08:56

0

О чём вы спорите?
Far давно в Open Source — уже можно было пять раз посмотреть

sasha1024 7 октября 2018 в 10:07

0

Не совсем понял, как реализация Far'а что-то докажет.

IRainman 22 сентября 2018 в 12:20

0

Более того благодаря этому можно делать очень полезные вещи для многопроцессной обработки чего либо.

sasha1024 22 сентября 2018 в 12:26

+2

По сути, благодаря mmap мы избегаем лишнего копирования данных из одного участка оперативной памяти в другой. Что достаточно быстрая операция. А основной затык тут не в mmap vs. lseek+read, а именно в алгоритмах, которые либо довольствуются малыми кусочками файла, либо заставляют *вычитать с диска* (нагрузка на диск) и *распарсить* (нагрузка на процессор и опять же таки оперативную память) значительный кусок файла или вообще весь файл.

sasha1024 22 сентября 2018 в 10:29

0

Обосновать как-то сможете?

gearbox 22 сентября 2018 в 12:45

ЧТО ОБСУЖДАЮТ

Сейчас
Вчера
Неделя

Настолько ли стар твой Windows?
22,9k 113

Триумф и трагедия «Бурана»
44,9k 335

Китайское искусственное солнце…
17,5k 43

Полку ретроконсолей прибыло: PC Classic для тех, кому дороги DOS-игры
7,9k 36

NASA объявило о новом полете на Луну
2,7k 9

+3

редактор/ide отображает (при подсветке синтаксиса и прочих плюшах) не plain text (где да, memory mapping помог бы) а по сути AST дерево, которое надо где то хранить. И вот это AST во первых весит больше исходника, во вторых его еще создать надо — то есть распарсить весь исходник. vim худо бедно справляется потому что у него подсветка на регулярках, и она регулярно ломается.

sasha1024 29 сентября 2018 в 17:21

0

> И вот это AST во первых весит больше исходника…

Кстати, необязательно. Это AST может:

- во-первых, не содержать в себе сами строки, а лишь ссылаться на позиции в исходном файле;
- во-вторых, быть не полностью детализированным (т.е. условно говоря, мы распарсим первую функцию в файле-исходнике полностью на всю глубину, но если она выше экрана, то тут же забудем результат парсинга за исключением её собственно начала и конца (т.е. удалим все узлы глубже узла самой функции), а если она снова попадёт в экран — распарсим заново (т.е. детализируем этот кусочек дерева)).

Но, да, на практике там делают немногие, та и в любом случае всё, что выше экрана, придётся распарсить, а это время, даже если результат детально не хранить.

vikarti 22 сентября 2018 в 15:36

0

Похожий функционал был еще в некоторых DOS-приложениях (для просмотра БОЛЬШИХ изображений вроде растровых карт) а в DOS memory-mapped files не бывает. Руками можно все сделать. Сложно но можно.

aamonster 23 сентября 2018 в 11:54

0

Memory mapped files — это просто немного удобства для программиста. Под CP/M-80 (в которой ещё не было не то что mmf — банальных file handles) текстовый редактор wordstar спокойно правил файл больше размера оперативки.

DrZlodberg 20 сентября 2018 в 16:13

+5

Не знаю, как фар, а блокнот++ открывает за <1сек.
Вообще по опыту там дико тормозит подсветка кода, и такая беда не только там.

AlexMt 20 сентября 2018 в 19:00

+4

Ну конечно, ведь в этом случае надо разбить текст на токены, и хорошо если разработчик предусмотрел токенизацию с экран, но ведь в сложных больших структурах где, к примеру, есть один корень и много ветвей (тот же xml), это может быть нереально, соответственно нельзя определить где находится начало, откуда можно стартовать синтаксический разбор, и соответственно корректную подсветку создать просто не выйдет.

DrZlodberg 21 сентября 2018 в 08:22

0

Как раз в xml это проще простого. Токенов там мало и они все достаточно легко распознаются. Там не требуется делать разметку с корня, достаточно чуть отойти назад и найти начало любого токена. Проблема будет только с CDATA, но на него пофиг.

AlexMt 21 сентября 2018 в 17:30

+1

Возможно, но опять же имхо зависит от структуры xml, на моём текущем проекте xml, смёрженный из 5-10 xml может быть от 100-150 MiB и выше, как повезёт. В этом случае открытие файла в пр++ становится мучением, как и работа с ним.

DrZlodberg 22 сентября 2018 в 10:21

0

Раньше использовал EditPlus2 (не помню, почему отказался частично), до 50Мб он переваривал не морщась, больше не пробовал.

Ezhyg 23 сентября 2018 в 20:10

0

При открытии большой пачки файлов, пусть даже мелких, на сотню байт-пару килобайт, зависает на иногда до 5-10 секунд, на так и не посчитал каком по счёту файле, что-то после 120, но не всегда на одинаковом количестве. (вот сейчас проверил — завис, правда только на пару секунд, на 127-ом)

Я тоже отказался от него в больше части предыдущих операций и перешёл на AkelPad. Он оказался лучше Notepad++.

DrZlodberg 23 сентября 2018 в 20:45

0

Если не секрет, зачем нужно открывать 100+ файлов одновременно? А если какое-то однотипное массовое редактирование, то уж проще скриптами. Ну и пару сек — это не катастрофа. Многие редакторы один файл открывают дольше.

Ezhyg 24 сентября 2018 в 00:41

0

Пара секунд только сейчас, при открытии же «обычной пачки» из ~200-300 файлов — зависает на заметное время.

Для перевода делаю копию файлов. А им нужно иногда конвертирование между форматами (из Unix в Win) и изменение кодировки (последнее можно автоматизировать).
После покупки автором оригинала — огрызкобука, часть файлов он стал сохранять в формате Unix. Изредка он заменяет много файлов, бывают сбои синхронизации и резервного копирования. На самом деле такое редко нужно, но всё же бывало, такое приходилось делать массово.
Да, можно написать конвертер и я писал один (но только для смены кодировки), но он тоже не универсален, тем более учитывая, что иногда файлы для работы нельзя выделить из всей пачки :(.

khim 24 сентября 2018 в 01:49

0

Осталось понять чем вам perl-скрипт на 20 строк не угодил и зачем для конвертации вообще привлекать редактор.

Ezhyg 24 сентября 2018 в 03:23

0

«Осталось понять, чем…» вы читали предыдущий комментарий :(. В котором я достаточно полно описал, зачем мне нужен редактор, да, помимо двух «конвертаций»!
А ещё, помимо самого скрипта, мне внезапно(!) понадобится интерпретатор — сам perl. Офигенный «скрипт», угу.
А ещё, он делает только одно, довольно редко нужное действие, но не делает второе, самое важное конвертирование — смену кодировки.

khim 24 сентября 2018 в 04:30

0

> А ещё, он делает только одно, довольно редко нужное действие, но не делает второе, самое важное конвертирование — смену кодировки.

Смену кодировки делает iconv.

> А ещё, помимо самого скрипта, мне внезапно(!) понадобится интерпретатор — сам perl.

На любой современной Unix-системе он есть. А откуда у вас возьмутся файлы в Unix-кодировке если у вас нету Unix'а — для меня загадка.

В любом случае если такое происходит не один раз в жизни, то проще поставить CygWin, чем забивать гвозди микроскопом, если один раз — можно и потерпеть пока редактор эти файлы откроет…

Druu 24 сентября 2018 в 04:53

0

> В любом случае если такое происходит не один раз в жизни, то проще поставить CygWin, чем забивать гвозди микроскопом

Забивание гвоздей микроскопом — это как раз куча мусора в системе в виде cygwin'ов, перлов и прочей ненужной гадости для решения одной задачи.

Ezhyg 24 сентября 2018 в 05:32

0

Ваши комментарии оскорбляют мой разум! Даже не знаю, чего в них больше — издёвки, насмешки или просто оскорбления.

Вы второй раз не потрудились прочитать, видимо лозунг — «комментарий не читай, ответ строчи», вам весьма близок?

Ezhyg 24.09.18 в 00:41

> После покупки автором оригинала — огрызкобука, часть файлов он стал сохранять в формате Unix

khim 24.09.18 в 04:30

> А откуда у вас возьмутся файлы в Unix-кодировке если у вас нету Unix'а — для меня загадка.

Для меня загадка — диагноз, который можно поставить по этой цепочке комментариев.
Вдобавок даже в предыдущем комментарии вы, похоже, так же не потрудились прочитать комментарий на который решились ответить, с как бы «желанием» помочь.

> Смену кодировки делает iconv.

Я рад за него, чесслово! Мало того, изредка, даже использую его помощь.

> На любой современной Unix-системе он есть.

Не на любой. Но это и не важно, при чём тут юниксы ваще?
EditPlus2 — Windows программа, как я могу работать в ней, при этом конвертировать что-то скриптом питона, а затем айконом? (только не вздумайте мне рассказывать про вторую машину или виртуалки)

> В любом случае если такое происходит не один раз в жизни, то проще поставить CygWin, чем забивать гвозди микроскопом, если один раз — можно и потерпеть пока редактор эти файлы откроет…

Да уж… тащить огромный пакетище для мелких разовых вещей — очень по красноглазиковски!
Здесь я согласен с @Druu

Если уж вы решили комментировать, то будьте добры, потратьте своё, безусловно бесценное, время и прочитайте уже комментируемое!

opetrenko 24 сентября 2018 в 09:09

0

Не оскорбляйтесь так сильно. Я, например, долго пытался понять, автор какого именно оригинала купил огрызок, и почему он, автор, решил сохранять в Юникс-формате.

Опять же, программисты наверное предполагают, что у вас есть возможность хоть на чем-то написать маленький скрипт.
Наверняка, не пытаясь этим вас обидеть.

Ну а айконв, наверное, посоветовали, поскольку вы писали свой велосипед.

Никоим образом не пытаюсь оскорбить ваш разум. Может, я не совсем понял, что он писал и как думал, поэтому заранее прошу у него прощения.

Надеюсь, мой незазангажированый комментарий убедит вас, что вас не хотели обидеть.

Что до сотен файлов открытых в редакторе — то, очень интересное наблюдение. Прям как в том анекдоте, на любую бензопилу, найдется рельса которая её сломает. Интересно, сколько сотен или тысяч файлов повесят редактор окончательно.
Да, так и не понял каким именно вы пользуетесь.

Ezhyg 24 сентября 2018 в 09:49

0

Я никогда не оскорбляюсь, не вижу смысла. Чего и вам желаю :D

> долго пытался понять, автор какого именно оригинала купил огрызок, и почему он, автор, решил сохранять в Юникс-формате.

Автор оригинала (файлов), переводимого мной (слово перевод вам надеюсь понятно?)

Формат Юникс, потому что огрызкобук же, потому что там этот формат вообще-то по умолчанию в его редакторе. А не самого мак-а, потому что его редактор — *никсовый, а не яблочный.

> Опять же, программисты наверное предполагают, что у вас есть возможность хоть на чем-то написать маленький скрипт.
> Наверняка, не пытаясь этим вас обидеть.
> …
> поскольку вы писали свой велосипед.

Так я и написал. Хотя это не скрипт и уж точно не велосипед! Нормальная команда запуска редактора без гуя (консольно), с параметрами конвертирования из одной кодировки в другую. И я сразу же объяснил, что конвертировать надо «два раза», но нет… начинается снова предложение вначале одного, на перле, потом второго на пхп, да и вообще не под ту ось. Это же клиника!

> Что до сотен файлов открытых в редакторе — то, очень интересное наблюдение.

Я же говорю, это бывает нужно редко, ну какие буквы-то незнакомы? :(
Тем более, не важно — отдельно я сконвертирую или вместе, мне же всё равно эти файлы открывать и редактировать.
Всего файлов там около 3800, но ни разу ещё не было чтобы «испортилось» больше пары сотен.

И я описал почему я так же отказался, как и автор комментария, прокомментированного мной.

> Раньше использовал EditPlus2 (не помню, почему отказался частично)

@0xd34df00d в репах есть:

> безазотистые вещества (6,5 %), азотистые вещества (1,1 %), жиры (0,2 %), минеральные соли (в нём очень большое содержание кальция), витамины (А — 0,04 мг, С — 8—20 мг, В1 — 0,08—0,11 мг), значительное количество сахаров и витамина РР. Богата янтарной кислотой.

Никаких dos2unix там нет! Это ген такой или вещество?

0xd34df00d 24 сентября 2018 в 07:09

0

Осталось понять, зачем тягать какие-то непонятные скрипты, если в репах есть dos2unix.

vitaliy2 22 сентября 2018 в 20:53

0

Notepad++ не очень, т.к. не предназначен даже для самого базового редактирования кода (не поддерживает Tab indent space align). Зато зачем-то поддерживает подсветку кода. Бред.

DrZlodberg 23 сентября 2018 в 20:46

+1

Подсветка кода сильно нужнее. Хотя выравнивание не помешало бы, есть такое.

vitaliy2 23 сентября 2018 в 23:19

0

Но и подсветку значительно сложнее сделать. На несколько порядков.

InterceptorTSK 24 сентября 2018 в 10:26

0

Это чем сложнее?)
Какая разница какой фон под символом?
Красный, белый или зилёный?
Почему белый фон под символом рендерится за 10мс, а красный фон за 100мс?)))
Цыфры условные.
Вы вообще понимаете, что разницы в принципе нет никакой? И не должно быть.
Разница получается в таких вот поделиях исключительно потому, что поделие сделано из говна и палок. Подсветку прикручивают не на том уровне абстракций. Всего навсего…

VolCh 24 сентября 2018 в 15:16

+2

Для подсветки нужен хоть какой-то синтаксический анализатор конкретного языка. На регулярках или строящий AST, обрабатывающий весь файл целиком или потоковый, в основном потоке или в фоновом процессе, но анализатор. Для каждого языка.

InterceptorTSK 24 сентября 2018 в 17:00

-1

Ага ага…
Вы еще скажите, что отрисовка процессором символа с фоном медленней, чем определение того какой под символом фон нужен.

VolCh 24 сентября 2018 в 17:11

0

Причём тут это?

InterceptorTSK 24 сентября 2018 в 18:56

-2

Как это причем?
У вас решение какое-то дичайшее.
Вы говорите что парсить строчку дороже, чем символ с красным фоном рисовать.
Вы ОБЪЯСНИТЕСЬ ХОТЯ БЫ КАК-ТО??? Как такое может быть вообще?
Или вы что не понимаете, что вы люто бредите??

sumanai 24 сентября 2018 в 19:19

+3

Какую строчку? А если 900 строчками выше объявлен символ начала комментария?

picul 24 сентября 2018 в 19:25

+1

Вообще то тут сильно зависит от того, что парсить и как рисовать. И на каком железе. Парсинг строки кода на C++ может быть сильно медленнее, чем отрисовка символа 16x16 пикселей с заданным фоном.
А вообще мне кажется, Вы с @VolCh о разных вещах говорите. Вы — о времени выполнения, а он — о времени разработки.

0xd34df00d 25 сентября 2018 в 01:37

0

Особенно учитывая, что парсинг C++ неразрешим в смысле Тьюринга и требует полноценного интерпретатора C++.

VolCh 25 сентября 2018 в 09:40

0

Конечно парсить строчку, иной раз в несколько мегабайт длиной, дороже, чем рисовать символы с разным цветом фона. Как минимум в случае когда символы мы по любому рисуем и нам нужно лишь определять каким цветом и с каким фоном их рисовать.

InterceptorTSK 25 сентября 2018 в 10:53

0

В реальность вернитесь уже.
У вас строка бывает длиной в несколько мегабайт? И зачем подсвечивать эту строчку длиной в мегабайт?
Вы про какой парсинг сейчас говорите?
Давайте разбираться. Парсинг всего документа целиком — это одно. И да, совершенно прав человек, когда пишет, что если начало комментария 900 строчками выше, то надо парсить весь документ. Так символ комментария может быть не только на 900 строк выше, а выше еще и на 1800-ой строке. Комментарий кто-то попытался сделать вложенным например. И не надо говорить что где-то это запрещено, а где-то разрешено, и где-то документ будет валидным, а где-то нет. Суть не в этом. Парсить ДОКУМЕНТ в любом случае нужно. Но я не про парсинг документа) Я про парсинг строки. Т.е. где в строке подсветить символы. Я про графику. И только. Т.е. есть строка, уже распарсенная. Так вот вам в этой строке нужно подсветить исходя из распарсенного — символы 10,11,12, 34,35,36 и т.д. Вот это и тормозит. А тормозить не должно никак. Имея именно распарсенный документ любой, т.е. абсолютно любой — тормозит именно подсветка символов. Это же увидеть абсолютно легко. Берете документ на 10кб и на 10мб — подсветка тормозит в обоих случаях. И тормозит именно определение того, где какой символ подкрасить, т.е. фон поменять. А фон меняют через лютые костыли зачем-то. Вот это и тормозит больше всего.
Выходит так, что тормозит не парсинг документа, а графика. Это как с курсором выжирающим ядро. Типа того.

VolCh 25 сентября 2018 в 13:31

0

Документ состоит из строк. Распарсить документ = распарсить все строки. И это вам нужно вернуться в реальность, где строки никто не парсит снова, чтобы подсветить её части. Определение цветов символов в строке — это не парсинг.

InterceptorTSK 25 сентября 2018 в 15:12

0

Да что вы говорите) Т.е. если вы получили некую структуру данных о документе, то в ней что-то сказано о том где и как красить фон символов? Наложение раскраски тоже никто не отменял. Закомментированное что-то может быть тоже внутри раскрашено, но бледно. Перенос строки разве кто-то отменял? И как вы в такой строке красить собираетесь? А если идут 20 табов и строка уходит за экран? Тоже подкрашивать то что ушло за экран? Вам графическую часть парсить придется и еще как. И да, тут нужно будет тоже парсить. И это парсинг тоже.

VolCh 25 сентября 2018 в 18:40

0

Наложение — это наложение, проецирование если угодно, может разметка, но никак не парсинг. И до графической части там ещё далеко.

Mr_Rm 1 октября 2018 в 11:13

0

А до какого уровня парсить, насколько подробную подсветку мы хотим?
Рисовать буквы одним цветом, цифры другим, прочие символы третьим — элементарно и почти ничего не стоит. Подсветка ключевых слов, строк, комментариев — уже зависит от синтаксиса языка. Выделить имена классов, элементы перечислений, парные скобки — нужен полноценный парсер. Неиспользуемые параметры?
Неинициализированные переменные? Теперь рисование разноцветных символов требует транслятора с анализом dataflow…

vitaliy2 24 сентября 2018 в 20:07

0

Я имею ввиду реализовать подсветку синтаксиса в редакторе программного кода на порядки сложнее, чем реализовать в редакторе tab indent space align.

Во втором случае понадобится всего несколько строчек кода, а в первом месяцы или даже годы работы (если делать в одиночку, придумать кучу всяких тем, поддержать 100 языков программирования, для каждого из которых нужен синтаксический анализатор, сделать всё оптимайзно, красиво в архитектурном плане и т.д.).

Так вот, Notepad++ решил пойти по странному пути. Потратить годы работы на подсветку синтаксиса — это ок, а потратить 5 минут времени на Tab indent space align — не ок, хотя эти две фичи одинаково важны.

picul 24 сентября 2018 в 22:51

+2

Ну, на счет одинаково важны — это все таки субъективно. Я вот вообще почти не слыхивал про tab indent space align, а подсветка синтаксиса очень помогает.

Vlad_IT 20 сентября 2018 в 15:25

+1

Если отключить из редактора intellisence, подсветку синтаксиса, то вероятнее он откроется быстрее. Тот же sublime, если автоматом не поставит подсветку синтаксиса, то может открывать файлы в разы больше чем с подсветкой. Хотя Vim например и с подсветкой открывает достаточно шустро.

DrZlodberg 20 сентября 2018 в 16:14

+7

Блокнот++ открывает быстрее секунды с подсветкой. На самом деле там 2 проблемы
1. подсветка. У многих она тормозит. У некоторых — безбожно.
2. Весь файл в одну строку. Такая подстава многие редакторы ставит на колени. Как пример: PSPad с подсветкой справляется, а вот с однострочниками нет. Более того, там даже прокрутка будет безбожно тормозить (если вкл автоматический перенос)

domix32 21 сентября 2018 в 12:33

0

Помнится перестал использовать PSPad после того как надоели постоянные падение при попытке открыть хотя бы метровый файл

Azan 21 сентября 2018 в 17:14

0

Та же самая ситуация, сам по себе PSPad был отличным продуктом, но если бы не эти мелочи. Но зато я открыл для себя существование Notepad++ и Sublime Text

vitaliy2 22 сентября 2018 в 20:57

0

Если бы ещё в Notepad++ можно было бы редактировать программный код. Туда зачем-то впихнули подсветку синтаксиса, а какой в этом толк, если всё-равно не поддерживает tab indent space alignment.

sumanai 22 сентября 2018 в 21:01

+1

> tab indent space alignment

И это всё?

IRainman 23 сентября 2018 в 13:51

0

Подсветка синтаксиса в Notepad++ полезна тем чтобы глаза не вытекали и чтобы можно было, например, скрипт для cmd отредактировать. А для редактирования кода и разработки есть IDE, её и необходимо использовать.



Alexeyslav 24 сентября 2018 в 08:23

0

Имеющаяся IDE неповоротлива и бедна на подсветку синтаксиса, поэтому используется Notepad++



landarken 24 сентября 2018 в 15:55

0

А что мешает самим написать плагин?



vitaliy2 24 сентября 2018 в 20:23

+1

1. Написал бы, но я работаю с другими языками программирования.
2. Это странно, для такой базовой функциональности писать плагин. Давайте ещё напишем плагин нажатия клавиши Enter, чтобы перенести курсор на следующую строку, предварительно создав её, потом напишем плагин для стрелок — а чё, хочешь перемещаться по коду, ставь плагин! и т. д.
3. У notepad++ есть какие-то баги с плагинами — не сохраняются настройки.
4. Но конечно же всегда можно сделать Pull Request! Но даже если бы я писал на этом языке, то не знаю, стал ли бы я делать. Потому что когда языки совпадали, я пулл реквестил аж в 5 проектов. В двух из них авторы до сих пор рассматривают их уже 2 года (хотя перед pull requestом я с ними общался, они были не против изменений). Казалось бы, всё готово, им нужно только одну кнопку нажать и смержить изменения, но нет. В одном проекте исправил ошибку, но pull request тупо отклонили, сказав, что не хотим мы ваших исправлений. И только в двух проектах pull requestʼы без проблем приняли.

   Выяснять отношения и годами просить принять pull request поднадоедает. Хочется просто взять, реализовать фичу, и всё, а не заниматься не пойми чем. Хочется заниматься именно программированием.



EmmGold 26 сентября 2018 в 18:43

0

хм

- вы перемещаетесь по коду стрелочками
- «Выучить новый язык программирования» на уровне написания плагина непосильная задача

да практически герой статьи в топике.



vitaliy2 26 сентября 2018 в 19:59

0

Необязательно непосильная, но значительно усложняет процесс. Вместо того, чтобы взять и написать, нужно вначале изучить язык, потом разобраться как писать эти плагины, подключать и т. д.

И самое главное — ради чего? Ради того, чтобы следующий человек пришёл и сказал «А где tab indent space alignment?»? Ну да, наверное, найдёт плагин в гугле. По факту плагин уже есть давно. Но работает плохо. +В Notepad++ не сохраняются настройки плагинов.



Alexeyslav 26 сентября 2018 в 23:09

0

Не замечал такой проблемы. Может, это из-за того что программа с плагинами поставлена в папку Program Files запись в которую под правами пользователя по умолчанию системой пресекается.

sumanai 26 сентября 2018 в 23:18

0

Интересно, а куда ещё ставить программы, кроме как в Program Files? И по идее плагины должны хранить настройки в апплдате, там же, где хранят свои настройки другие нормальные приложения



Alexeyslav 26 сентября 2018 в 23:23

0

Должны, но приложение можно поставить и по старинке, оно будет всё хранить в своей папке получая каждый раз облом. А в новой винде есть ещё и прозрачный механизм переноса этих файлов в профиль для старых приложений, они не заметят подвоха. Может, механизм как-то неправильно работает — насройки исправно сохраняются, но плагины пытаются читать их по старому месту?

sumanai 27 сентября 2018 в 13:23

0

> Может, механизм как-то неправильно работает — насройки исправно сохраняются, но плагины пытаются читать их по старому месту?

Вряд ли. Это нужно сохранять настройки от пользователя, а запускать от админа, только тогда отключится виртуализация файловой системы.

DrZlodberg 22 сентября 2018 в 10:11

0

Странно. Сам активно им пользовался и падений особо не припомню. Отказался как раз из-за невозможности работы с большими однострочными файлами (дичайше тормозил). Но если файл порубить предварительно на строки — вполне норм. Для нарезки использовал EditPlus2, он без проблем пережевывал 50+метров (больше просто не пробовал)



domix32 22 сентября 2018 в 20:09

0

Я в читерскую бытность частенько открывал непотребные файлы текстовыми редакторами. Переживали такое немногие. Это было году в 2004 в расцвет WinXP и тогда много чего под винды отваливалось по разным причинам.



Goodkat 20 сентября 2018 в 21:04

+1

Ну сделай ты подсветку синтаксиса отдельными потоком, чтобы не мешать пользователю пользоваться твоей программой.
Сперва разработчики для iOS догадались (ОК, их заставили) переносить тяжёлые и долгие операции в фоновые потоки.
За ними подтянулись разработчики для Android.
А на десктопах всё так же по старинке всё одним потоком, последовательно — если у пользователя комп тормозит, то пусть купит себе новый, ага.

sasha1024 22 сентября 2018 в 12:03

+1

Интересно, за что Вас заминусовали…

VolCh 22 сентября 2018 в 13:16

0

Ну у джетбрейнс парсинг отдельным потоком или процессом



Goodkat 22 сентября 2018 в 14:05

+1

За это мы их и любим.
Но очень многие разработчики ожидают, что их творения будут работать в идеальных условиях, на быстрых процессорах, с достаточным количеством памяти, с быстрой и всегда доступной сетью и т.п.

freuser 22 сентября 2018 в 21:18

+1

… с правами админа (десктоп), с доступом ко всем возможным действиям, нужным и ненужным, но пусть будут (мобайл)…

Первое еще можно понять — сисадмин по своей лени выдал разработчику полные права для изменения либ и прочего, вот последний так же ленится релогаться туда-обратно и

пишет под админом. Ваше право, возьмите да укажите в требованиях к установке полные права. Неее, мы умолчим о том, что разработка у нас через одно место, а юзер пусть грешит на себя — то ли комп глючит, то ли ОС криво встала… Из последнего — надстройка Overwolf для TeamSpeak3: под юзером запускалась, все рисовала, но не коннектилась к ей же запущенному TS3.
Порой доходит до абсурда — пользовался плагином EveryHTTPS, и на одном из сайтов сертификат был выдан на айпишник локалхоста, то есть у админа всё работало…

aamonster 23 сентября 2018 в 12:03

0

Чтобы делать что-то в отдельном потоке — нужны на порядок более квалифицированные разработчики. Без этого ловим глюки и тормоза (потоки ждут друг друга — не раз наблюдал, как в несколько потоков оказывается медленнее, чем в 1).

sasha1024 29 сентября 2018 в 17:24

0

Ну, я думаю, такие вещи, как посветка синтаксиса, в любом случае квалифицированные разработчики делают, не?

DrPass 29 сентября 2018 в 23:40

+1

> нужны на порядок более квалифицированные разработчики.

Что значит «на порядок более квалифицированные»? Человек, если вообще способен быть программистом, способен и освоить многопоточность. Многопоточность может потребовать больше внимательности и времени на проектирование, но для её использования не нужно какое-то профессиональное просветление. Это такой же обязательный навык для любого программиста (кроме тех, кто поэтапно связался с какой-то принципиально однопоточной платформой), как и умение делать циклы. Просто проходится на чуть более поздних лекциях.

PsyHaSTe 30 сентября 2018 в 20:35

0

Я бы согласился, если бы не истории, что физбазз для людей просто рокет саенс.

aamonster 1 октября 2018 в 14:16

0

Не забывайте, что в коде неминуемо будут ошибки, которые потом придётся искать. Как по вашему, достаточно для этого «пройти многопоточность на более поздних лекциях»?

Конечно, сейчас есть методы, позволяющие кардинально уменьшить сложность многопоточного кода (например, делать все данные либо локальными, либо immutable — вроде современные языки даже научились жёстко устанавливать это ограничение), но всё равно сложности хватает.

CheatEx 25 сентября 2018 в 14:48

0

Проблема в том что эту подсветку и структуру данных которую читает процесс редеринга надо синхронизировать. И тут мы получаем два варианта: или отрисовка пишет непосредственно в структуру которой оперирует рендер и огромные процессорные ресурсы улетают на обработку блокировок или у синтаксиса своя копия отрисока с ней сверяется по необходимости и мы имеем примерно удвоенное потребление памяти.

DrPass 26 сентября 2018 в 00:48

+2

> у синтаксиса своя копия отрисока с ней сверяется по необходимости и мы имеем примерно удвоенное потребление памяти.

С одной стороны, да. С другой стороны, я помню, как четверть века назад притормаживала подсветка синтаксиса в Turbo Pascal 7 на компьютере «Поиск 1.04». Было примерно как сейчас у меня в браузере при комментировании этой ветки. Лагало, неприятно было, но работать было можно.
Checkit показывал, что быстродействие того «Поиска» было на уровне 0,5 от IBM PC XT (там был 16-битный проц на 5МГц, 608К ОЗУ, и программная эмуляция видеоадаптера IBM CGA).
И там подтормаживала подсветка синтаксиса. Блин, но я никак не могу понять, почему она в некоторых IDE подтормаживает сейчас, на восьми 64-битных суперскалярных ядрах по 4ГГц каждое.

kalininmr 22 сентября 2018 в 19:10

0

да. vim почти всегда шустро.
любопытно за счет чего

OlegMax 20 сентября 2018 в 18:08

+9

XML хороший пример, но в другом смысле. Среди читающих это есть некоторая доля программистов, некоторым из них приходилось парсить XML. Сколько из них пользовались XML DOM парсером, который по определению жрет память, а сколько гораздо более экономным XML SAX парсером? Вот то-то же, DOM настолько удобен для программиста, что в 99% процентов случаев ему прощали его цену. А из десятков таких нюансов и складывается общий оверхед.

DrZlodberg 21 сентября 2018 в 08:30

+4

На самом деле sax\dom — это фигня (хотя да, скорость там очень отличается). Недавно столкнулся с кодом, который на больших объёмах почему-то (только у клиента) дико тормозил. Практически экспоненциально! Т.е. для файлов в 100 кб вывод занимает секунд 10, а метр — уже к часу дело идёт. Начал разбираться. Оказалось файл формируется по шаблону, а шаблон тупо нарезается на блоки, из которых собирается файл. String.replace-ом. В итоге у разраба на машине с 16Гб оперативки всё работает, а вот у клиента с 2..4Гб моментально выжирается вся оперативка и дальше становится всё очень печально…

aamonster 23 сентября 2018 в 12:19

+3

Напомнило, как я в своё время всадил фееричную багу.
После нескольких часов работы программа начинала тормозить. Причина: в ней было отладочное окошко с текстбоксом лога, в который я при обработке очередного блока данных добавлял точку (что-то typ dbgForm.textboxLog.Text.Append(".") — дело было на дельфях). Что с каждой точкой оказывалось всё медленней...

evocatus 23 сентября 2018 в 21:06

0

В драйверах Qualcomm как-то нашли кусочек кода, где для того, чтобы найти максимум в массиве, его сортировали по неубыванию. Пузырьком.
Это драйвер, т.е. критичный к производительности код. Для мобильных устройств. От Qualcomm.

Vladimirsencov 1 октября 2018 в 11:13

0

Если массив маленький, то не критично. Но зачем так криво делать? Можно просто в цикле прогнать один раз.

Druu 2 октября 2018 в 10:13

0

Да возможно все гораздо проще — исходно надо было сортировать небольшой массив или его часть, сделали пузырек, потом требования изменились и оказалось, что нужен только один элемент, а пузырек остался, т.к. не стали все переделывать.

Почему-то многие забывают, что код не всегда пишется оп и сразу, зачастую он не раз рефакторится, прежде чем дойти до продакшена.

multiprogramm 21 сентября 2018 в 12:14

+9

Абсолютно точно.
То ли в 2014, то ли в 2015 у нас налоговая приняла законы, по которым XML-ины электронной отчётности по размеру стали допустимы до 10гб (вроде бы с изначального «копеечного» лимита 10-100мб), так все решения на DOMах очень красиво посыпались, когда пошли клиенты с громадными «контролируемыми сделками» и «книгами для ндс». Потому что DOM одной только памяти жрёт х10 от размера файла, и если раньше (100мб-1гб) на это закрывались глаза, то на 100гб глаза уже не закроешь, не говоря уже о платформе x86.
Делал в те времена в системе электронной отчётности проект по уходу с DOM (легаси было крепко на него завязано, XPath и прочие плюшки активно использовались) на SAX в один пробег, с «мультиков» плюшек или обоснованным отказом от них — веселья было много. Теперь в резюме указываю тег XML, а на собеседованиях периодически посмеиваются, мол, что там знать-то, древовидная структура, теги/атрибуты и всё. Так-то оно да, но, как и в физическом мире, когда масса объекта пересекает определённые границы, проявляется много интересных эффектов, о которых даже не думаешь, щупая маленькие объекты.

DrZlodberg 22 сентября 2018 в 10:13

0

Мы с этой проблемой лет на 10 раньше столкнулись. Объёмы тогда были не такие, но 50..100 метров на тех машинах тоже было не подарком.

0xd34df00d 21 сентября 2018 в 20:00

+1

Однопроходный алгоритм в ленивом DOM-парсере вполне может быть O(1) по памяти.

Да, я такое писал с одним из хаскелевских DOM-парсеров. Было очень удобно, приятно, естественно и эффективно.

OlegMax 21 сентября 2018 в 22:17

0

Но это уже не DOM.

0xd34df00d 21 сентября 2018 в 22:20

+3

Почему?

Надо разделять стратегию вычисления и API, тем более, что сайд-эффекты остаются все те же. Ну, да, если там невалидный документ, то я получу ошибку чуть позже, возможно, потратив ресурсы процессора на обработку валидного куска DOM-дерева. Ну так с SAX бы то же было бы.

DrZlodberg 22 сентября 2018 в 10:19

0

Это сильно зависит от структуры документа. Бывают форматы, где есть ссылки из начала в конец, т.е. для обработки самого начала уже надо иметь доступ ко всему файлу. И загружать его всё равно придётся полностью. Так что никакое O(1) не отменит того, что грузиться оно будет медленнее и памяти занимать сильно больше. Причём второе — бессмысленно, т.к. программы обычно всё-таки не работают с самим деревом, а просто загружают его в свою структуру просто дублируя данные.

0xd34df00d 22 сентября 2018 в 16:52

0

Ну, в этом случае можно просто сохранить ссылку на кусок в конце.

А вот если вам нужно посчитать, скажем, 0.95-персентиль по некоторым нодам в документе, это уже может быть болезненно, да.

DrZlodberg 23 сентября 2018 в 20:42

0

В смысле? Чтобы сохранить ссылку нужно сначала распарсить весь файл и найти, куда ссылаться, т.е. нужна доп логика над парсером. Проще sax-ом обработать в один проход. А ещё ссылки бывают не только вперёд, но и назад. DOM для того и нужен, чтобы работать с рандомным доступом, без этого смысла в нём нет.

0xd34df00d 23 сентября 2018 в 21:47

0

Ну это уже от конкретной задачи зависит, так сферически в вакууме трудно обсуждать конструктивно, но смысл в том, что встретили ссылку вперёд — сохранили в список соответствующий предикат. Тогда вы будете O(n) по числу предикатов/ссылок, а не по всему объёму документа.

А иногда оказывается выгоднее тупо распарсить документ два-три раза, зато каждый раз в один проход.

DrZlodberg 23 сентября 2018 в 22:10

0

Вот только вы, похоже, не учли один нюанс. В этом случае парсер(!) должен знать структуру конкретного файла, чтобы понимать, что является ссылкой, а что якорем. Что слегка убивает идею универсального псевдо-dom. Imho на этом этапе проще перейти к нормальному sax, это будет проще, чем морочиться со всей лишней логикой чтобы вкорячить сюда то, что тут не требуется. (троллейбус.jpg)

0xd34df00d 24 сентября 2018 в 00:09

0

Зачем это парсеру знать? Парсер просто выдаёт ленивое дерево нод, знать это нужно вам, когда вы по этому дереву проходите.

Druu 24 сентября 2018 в 03:38

0

> Парсер просто выдаёт ленивое дерево нод

А как парсер сможет выдать дерево нод (не важно лениво или нет), не зная структуры документа?

Вообще по такой логике можно любую задачу решить за O(1) — объявить что ф-я, которая делает то, что нужно, ленивая, и не запускается, т.к. оказывается не нужна.
В итоге можно мгновенно решать пр-задачи.

0xd34df00d 24 сентября 2018 в 07:12

0

> А как парсер сможет выдать дерево нод (не важно лениво или нет), не зная структуры документа?

А зачем парсеру знать структуру? Парсер — это синтаксис, структура — это семантика конкретных элементов. XML-парсеру пофиг, OOXML там, налоговые выверты или XHTML.

И если это ссылки, скажем, в смысле примечаний формата fb2 какого-нибудь, то не вижу проблем.

DrZlodberg 24 сентября 2018 в 08:28

0

Ок. Пусть fb2.
Есть док, у которого в самом начале ссылка на картинку с обложкой, которая лежит в конце. Когда клиент захочет сразу же отобразить обложку (ну он же работает с dom и уверен, что она уже загружена) вы предлагаете:
1. загрузить весь документ до обложки и работать как с обычным dom?
2. дропнуть всё до обложки а потом начать парсить сначала, когда клиент продолжит читать документ с начала первой главы?
3. ???

khim 24 сентября 2018 в 09:03

0

> Когда клиент захочет сразу же отобразить обложку (ну он же работает с dom и уверен, что она уже загружена)

Даже если она уже загружена — поиск в нём долог и дорог. И да, сложить две строки — это **тоже** дорого и сложно. Как и **выделить подстроку**. И как изменить регистр. И как многое другое. Лучше всего про эту банальность (как и почти любую другую банальности) описал Джоел…

Если вы будете помнить, о том, что поиск в DOM — это дорого и долго, то у вас всё будет работать быстро и с DOM'ом и, чуть медленнее (но со значительной экономией памяти) — с «ленивым» DOM'ом. Если же вы будете считать, что складывать строки легко и быстро — то вам, извините, никакие абстракции не помогут. Скорее помешают.

У меня есть ощущение, что вина всему — то, что начинающие программисты больше не начинают с Бейсика на C64. Когда вы писали на языке, который был в 100-500 раз медленнее (не в 100500, а в 100-500 — это довольно точная оценка), чем тот же алгоритм в машинных кодах и при этом процессор у вас был частотой 1 (один) мегагерц… то увидеть, что операции со строками — таки медленные можно было почти что невооружённым глазом. А сейчас — сеньор-программисты об этом не подозревают… а абстракции… абстракции могут быть любыми — важно только не забывать, что все они — дырявые…

DrZlodberg 24 сентября 2018 в 09:48

0

Собственно на вопрос вы так и не ответили. Те, кто понимает, что dom — это медленно и дорого — его по возможности не используют, так что речь не о них. И при повторном чтении там будет совсем не чуть медленнее (тем более что построение dom уже само по себе весьма затратно). Да и речь вообще была о том, что **просто так** заменить dom на ленивый нельзя. А не просто так — почему сразу не использовать sax?

0xd34df00d 24 сентября 2018 в 16:39

0

> А не просто так — почему сразу не использовать sax?

Для унифомности.

Впрочем, я тут ещё подумал и понял, что в ленивом сеттинге разницы между DOM и SAX нет вообще. Что там функция, паттерн-матчащаяся на ноды, что там функция, паттерн-матчащаяся на выплевки SAX-парсера.

И если вспомнить, что в ленивых языках всякие списки, ленивые деревья и прочее — это скорее управляющие структуры, а не структуры данных, то это начинает иметь всё больше и больше смысла.

DrZlodberg 25 сентября 2018 в 08:36

0

Для ленивых — может быть. Для остальных — наоборот, меньше смысла, т.к. логика обработки меняется.



0xd34df00d 24 сентября 2018 в 16:36

0

Если у меня недостаточно памяти, то я бы делал (2), в первый проход дропая всё, кроме обложки. SAX'ом вы тоже два прохода будете делать, впрочем.

DrZlodberg 25 сентября 2018 в 08:34

0

Вы меня упорно отказываетесь понимать. Какие 2 прохода? Вы работаете с dom! В sax всё прекрасно делается за 1 проход. Речь не о нехватке памяти, а о дублировании в памяти всего документа в dom.

А если вы работаете проходами, то на кой чёрт вам dom, если всё равно алгоритм работы полностью меняется. IMHO ленивый dom имеет смысл только при простой последовательной обработке, когда очень лень эту обработку писать. Да и то по факту сильно проще там не станет.



0xd34df00d 25 сентября 2018 в 18:53

0

Как вы с SAX решите вашу задачу с fb2 выше в один проход?

DOM мне затем, что, с учётом ремарки выше, это вопрос того, кто оборачивает поток XML-сущностей (тег/атрибут/etc) в управляющую структуру. Мне удобнее, когда это уже сделали за меня, чем строить дерево ручками.

DrZlodberg 26 сентября 2018 в 08:34

0

А в чём проблема? Мы же знаем, что картинка будет позже в добавим, когда дойдёт до неё дело. А вот работающие с dom такими вопросами обычно не заморачиваются, т.к. по идее всё дерево доступно полностью сразу.

Это всё хорошо, когда вы пишите редактор xml или сериализацию один-в-один. Но когда данные в памяти не соответствуют структуре файла (а обычно так и бывает) — их всё равно придётся преобразовывать. И когда это делать разницы нет. Хоть при обработке потока xml, хоть вытребая из дерева. Но первое быстрее и требует меньше памяти.

khim 26 сентября 2018 в 22:02

0

> А вот работающие с dom такими вопросами обычно не заморачиваются, т.к. по идее всё дерево доступно полностью сразу.

Никогда оно не бывает «доступно всё сразу», если всем известные числа вспомнить, то легко понять, что переход к следующему элементу — это что-то типа 0.5 нс (обращение к кешу L1), у куда-то «вдаль» — что-то типа 100 нс. Разница — **два порядка**. И да, это в полностью распарсенном и загруженном дереве.

DrZlodberg 27 сентября 2018 в 08:53

0

Всё-таки речь не о том, что они не бесплатны, а о том, что об этом никто не задумывается. Особенно с учётом того, что на загрузку там тратится на много порядков больше времени, чем на доступ. Собственно на фоне загрузки доступ уже можно считать условно бесплатным. Ну не считая случаев, когда загрузка засрёт всю память и мы будем работать со свопом. Это к обработке уже не относится.

khim 27 сентября 2018 в 18:52

0

> Особенно с учётом того, что на загрузку там тратится на много порядков больше времени, чем на доступ.

Ой ли? А если прикинуть? Если доступ к памяти занимает 100 наносекунд, то на 10Гбит/с канале вы за это время загрузите 120 байт. То есть «много порядков» у вас будет только если ссылки из одного места в другое случаются пару раз в файле на мегабайт. Иначе — эти величины очень даже могут быть сравнимы.

Собственно те самые числа «должен знать каждый» именно потому что часто вещи, которые «на много порядков больше времени» занимают вовсе не так много, как вам кажется — и соответственно «очень простые и быстрые» вещи часто оказываются вовсе не такими «простыми и быстрыми»…

> Ну не считая случаев, когда загрузка засрёт всю память и мы будем работать со свопом.

А почему, собственно? Если у вас система на SSD, то обращение к своп-файлу вполне может происходить на скоростях в несколько гигабайт в секунду… а вот если нужен произвольный доступ — то там числа совсем-совсем другие, конечно…

> Это к обработке уже не относится.

А у чему это, я извиняюсь, относится? Если вам дадут XL-файл гиг на 10 — что вы ним будете делать?



Ogra 27 сентября 2018 в 19:24

0

> 100 наносекунд, то на 10Гбит/с канале вы за это время загрузите 120 байт.

Да нифига вы за это время не загрузите. На один TCP-handshake куда больше времени уйдет.

DrZlodberg 28 сентября 2018 в 08:56

0

> Иначе — эти величины очень даже могут быть сравнимы.

В сферических условиях в вакууме всё может быть. В общем случае на больших файлах разница будет и приличная

> А почему, собственно? Если у вас система на SSD, то обращение к своп-файлу вполне может происходить на скоростях в несколько гигабайт в секунду… а вот если нужен произвольный доступ — то там числа совсем-совсем другие, конечно…

В основном потому, что изначально задача была в частности «не засрать всю память». А если это неизбежно — то не полагаться, что программа всегда будет работать на машине с топовым ссд неограниченного размера или как минимум с 128Гб оперативки.

Вот мы плавно и вернулись к началу этого разговора, когда вместо нормального проектирования разработчик полагается на то, что железо всё вытянет.

> А у чему это, я извиняюсь, относится? Если вам дадут XL-файл гиг на 10 — что вы ним будете делать?

Для начала я выясню, что, собственно с этим файлом требуется сделать. И если нет задачи написать клон excel — обычно требуется лишь загрузка из него значений, для чего пихать весь файл в память нет совершенно никакой необходимости.

Если вдруг имелся всё-таки XML файл, то тем более. SAX-парсеру размеры файла совершенно не важны.

0xd34df00d 29 сентября 2018 в 02:06

0

> Если доступ к памяти занимает 100 наносекунд, то на 10Гбит/с канале вы за это время загрузите 120 байт.

Это сравнение апельсинов с бегемотами. Сравнивать задержку доступа с устоявшейся скоростью передачи несколько странно.

> Если у вас система на SSD, то обращение к своп-файлу вполне может происходить на скоростях в несколько гигабайт в секунду… а вот если нужен произвольный доступ — то там числа совсем-совсем другие, конечно…

А если произвольный доступ не нужен, то можно снова лениво стримить всё дерево в DFS-порядке, и мы вернулись в начало.

khim 29 сентября 2018 в 02:31

0

> Если доступ к памяти занимает 100 наносекунд, то на 10Гбит/с канале вы за это время загрузите 120 байт.
>
> Это сравнение апельсинов с бегемотами. Сравнивать задержку доступа с устоявшейся скоростью передачи несколько странно.

Время есть время. Тот пёзdёц, который мы тут наблюдаем в Хроме вот на этой самой странице с вероятностью 99% связан с тем, что кто-то *исходил из логики: на загрузку там тратится на много порядков больше времени, чем на доступ…* но применение этой

логики достаточное число раз привело к тому, что… имеем то, что имеем то, что имеем: Chrome 54 — работает быстро, а «ускоренный» Chrome 69-70… вот так, как он работает…

> А если произвольный доступ не нужен, то можно снова лениво стримить всё дерево в DFS-порядке, и мы вернулись в начало.

Не совсем. На SSD «ленивое» построение дерева приведёт, скорее всего, к ускорению. В оперативке — вряд ли. Но в случае, если человек считает, что произвольный доступ не стоит ничего и не учитывает того, что он, в общем-то, не так далёк от скорости работы сети… не поможет ничего!



0xd34df00d 29 сентября 2018 в 02:03

0

Ну вот точно так же вы её лениво добавите, когда дело дойдёт до неё, и с DOM. Я перестал понимать ваше изначальное возражение о ссылках вперёд.

DrZlodberg 29 сентября 2018 в 11:17

0

Я пытался объяснить, что вы используете сложный компонент с неизвестной внутренней логикой вместо быстрого и простого как полено, и при этом всё равно вынуждены использовать его как простой потому, как иначе не получится. Ну не смог и ладно…

Druu 24 сентября 2018 в 09:23

0

> А зачем парсеру знать структуру? Парсер — это синтаксис, структура — это семантика конкретных элементов.

Парсер же возвращает АСТ (ну, условно, в общем АСТ может быть произвольным графом, а не деревом)? АСТ — это и есть структура документа, разве нет?



0xd34df00d 24 сентября 2018 в 16:35

0

Это синтаксическая структура документа (как подсказывает расшифровка аббревиатуры).

VolCh 25 сентября 2018 в 09:44

0

С трудом представляюю как на основе исходного кода можно построить синтаксический граф не в виде дерева. Внутренние ссылки, перекрёстные, циклические, рекурсивные и т. п. уже не относятся, как по мне, к синтаксическому разбору, это уже дальше этап разбора дерева.

Druu 26 сентября 2018 в 13:38

−1

> Внутренние ссылки, перекрёстные, циклические, рекурсивные и т. п. уже не относятся, как по мне, к синтаксическому разбору, это уже дальше этап разбора дерева.

В лиспе АСТ является произвольным графом.



0xd34df00d 29 сентября 2018 в 01:57

+1

А можно пример недерева?

Druu 2 октября 2018 в 10:24

0

Штука в том, что ридер-макросы не являются токенами, и в них вы можете производить произвольного вида вычисления. Например — взять кусок АСТ и замкнуть его самого на себя:
для
#1=(yoba . #1#)
получится бесконечный АСТ '(yoba yoba yoba yoba yoba ....)
можно сделать такой цикличный АСТ, отдать на евал и он повиснет, например, ну или построить граф любого вида.

OlegMax 22 сентября 2018 в 11:36

+1

Как был реализован IXMLDOMNode::parentNode?

InterceptorTSK 22 сентября 2018 в 12:53

0

Может быть и никак.
С чего вы взяли что оно ему надо было?
Он же не полную реализацию стандарта заваял. Полная реализация вообще никому не нужна, никогда.

OlegMax 22 сентября 2018 в 13:07

+1

Но это уже не DOM.

InterceptorTSK 22 сентября 2018 в 15:31

0

Еще раз?) Давайте отвечу по-другому. Ему не нужная реализация полного дома.
И вам не нужна) Она никому не нужна.
Человек отбросил лишнее, и алгоритмически оно у него полетело. При этом совместимость осталась.
В чем собственно проблема?

А проблема в том, что стремятся создать универсальную библиотеку, а это полная чушь. Потому что даже математически это невозможно. Если вы разгоняете что-то, то что-то просядет в другом месте.

В идеале же одна универсальная библиотека пусть будет, я не против, собрать проект не вдаваясь в подробности нужно быстро. Но для случаев кода дом не нужен полностью, должны быть другие реализации. И их должно быть много. Выбираете под себя вам нужную, т.е. ту которая максимально заточена под вашу конкретную задачу, а не ворох, а остальное там вообще не должно быть реализовано, ну потому что не нужно оно.

И кто так делает?) Никто. Потому что мозгов нет совершенно.
И поможет ли оно? Нет) Потому что нужно понимать из вороха узкозаточенных реализаций что они могут. А значит нужно все равно глубоко знать реализации. Так таких людей опять нет)

В итоге — вменяемых реализаций нет. Людей понимающих реализации тоже нет. Что остается в сухом остатке, впрочем на самом деле в жидком? Правильно: только гавно.

Druu 22 сентября 2018 в 15:37

0

> Но для случаев кода дом не нужен полностью, должны быть другие реализации. И их должно быть много.

Так а почему вы их еще не написали? Идите и пишите, вместо того чтобы в комментах говно обсуждать.



popov654 25 сентября 2018 в 19:30

0

Можно же просто написать свою. Я в начале этого года пытался писать свой браузер. Разбивку на токены и построение DOM реализовывал сам. И оно работает вполне шустро (правда, на очень больших файлах пока не тестил)



0xd34df00d 22 сентября 2018 в 16:53

0

Да чтоб я помнил, какой я библиотекой пользовался из тучи.

Мог быть просто завязанный узел, могло вообще не быть и делегировано на уровень выше (ну как в дереве тут).

OlegMax 22 сентября 2018 в 18:16

0

Хм, библиотеку не помните, но уверены, что она обеспечивала O(1) по памяти?
Звучит правдоподобно.

0xd34df00d 22 сентября 2018 в 19:56

0

А вы помните все применённые вами библиотеки, языковые возможности и тому подобное в каждом нетривиальном случае 5-7 лет назад?

Достаточно помнить своё удивление, в конце концов, что ты просто написал код с

DOM-стилем беготни по дереву, а он просто за счёт ленивости скомпилировался в O(1).



Berkof 21 сентября 2018 в 08:22

+2

Нелюблю аппл, но помню как хлопал в ладоши при подключении к клиенту пирса (файлообменник в Новосибе) с маковского Леонардо (или как-там) клиента, который сначала подключался, показывал чатик и позволял уже формировать запросы на поиск файлов, а в фоне отрисовывал всю чепуху с 30к пользователей онлайн… На старом компе под виндовым DC++ это занимало минуту, на новом — секунд 15… Даже оптимизировать ничего не надо, просто ненужную задачу выбросили в фон и нехай там хоть по 5 минут думает.

trublast 21 сентября 2018 в 18:14

+2

Или снимаешь галочку отображения списка клиентов в виндовом DC клиенте и получаешь тот же результат. При количестве пользователей в десятки тысяч этот список все равно смысла не имеет. Это не местечковый хаб с чатиком из начала 2000х, не нужен там этот список, там большая часть не люди, а приставки к ТВ.
Я ещё и чат отключал. Запуск приложения и подключение к хабу — секунда, ну может две.

serf 21 сентября 2018 в 16:24

0

А в vscode как на этом же файле? Это ведь vscode сейчас как раз модно и молодежно.

EvilFox 21 сентября 2018 в 16:38

0

Лучше чем Sublime (возможно от версии зависит). У меня саблим долго грузил и падал на файле 131МБ, а vscode свободно открыл (предупредил о возможных тормозах и предложил отключить подсветку, я не стал этого делать и так норм работает, только памяти выжирает не мало) и в отличие от n++ сворачивал и разворачивал огромные ветки/узды без особых тормозов (в n++ они были минутами).

aleksandros 25 сентября 2018 в 21:19

0

Невероятно, но факт. MC 20 лет (!), начиная с Windows 98 заявляет, что-де «переписали большую часть кода с нуля на ассемблере» и т.д. и т.п. Но с каждым разом «переписанная» система становится всё больше, а детские глюки как были, так и остаются.



Am0ralist 25 сентября 2018 в 21:48

0

> Невероятно, но факт. MC 20 лет (!), начиная с Windows 98 заявляет, что-де «переписали большую часть кода с нуля на ассемблере» и т.д. и т.п.

Где? Я хочу почитать про такое.

aleksandros 26 сентября 2018 в 19:02

0

Точно помню, что в те времена читал такое в журналах.

DMGarikk 26 сентября 2018 в 19:15

0

никогда они такого не говорили.
9х до самого своего упора, до ME, была допиленной 95 виндой на которую в 98 накатили IE4… да и закопали её потому что «переписывать» было себе дороже

aleksandros 27 сентября 2018 в 14:43

0

Ну, я и сам так считаю и спорить не буду. Но точно помню слова про переписанный код, ассемблер, рост производительности и всё такое.

muove 20 сентября 2018 в 15:42

+9

нет не 1$, а блин 1$ + стоимость нового телефона за XYZKRE фантиков, потому что на старом это не работает.

algotrader2013 21 сентября 2018 в 11:00

+3

Именно так. Лично мне пришлось пару лет назад купить новый телефон, потому, что полностью устраивающий меня galaxy s2 открывал сообщение в скайпе за 30 секунд. К слову, прямо сейчас при наборе этого сообщения и пришедший ему на смену s6 подтормаживает. Неужто разрабы хабра тоже вирусом поражены?

VolCh 22 сентября 2018 в 14:08

+5

Редкие посты получают 1000+ комментариев.

burzooom 23 сентября 2018 в 15:36

0

1000 комментов, каждый в среднем по 300 символов. Сколько это? 300 килобайт?

VolCh 23 сентября 2018 в 16:40

0

Килобайты особо непричём, скорее дело в количестве Dom нод и т. п.

evocatus 23 сентября 2018 в 21:13

0

600. UTF-8 же.



popov654 25 сентября 2018 в 20:18

0

Способ повесить ваш Chrome на минуту-полторы, если у вас x86:

1. Нажмите на этой странице Ctrl+U
2. Нажмите Ctrl+A на открывшейся вкладке
3. Кликните по выделенному тексту правой кнопкой мыши в любом месте
4. <>
5. <>
6. <>
7. Снимайте процесс, если надоело ждать.

evocatus 25 сентября 2018 в 21:33

0

У меня Firefox и он не подвесился, хотя реагировал не сразу (сотни миллисекунд).

masv 23 сентября 2018 в 09:39

0

У хабра вроде есть приложение, где (наверное) можно хорошо оптимизировать отображение только нужной информации, для плавности.

novice2001 20 сентября 2018 в 15:46

+21

А кто сказал, что сделать хорошее, быстрое и надежное приложение обязательно очень долго и очень дорого?
Вот тот же Web. Да, если каждый раз с нуля переписывать все эти библиотеки и фреймворки, то конечно. Только нужны ли они на 99% сайтов? Веб — это информация, текст, таблицы, картинки. Для этого всего вообще не нужен ни js, ни всякие библиотеки/фреймворки. уберите бессмысленное и бесполезное украшательство — и веб начнет летать на самых убогих компьютерах.
А там, где они нужны, как правило, хватило бы ресурсов и для написания «с нуля». Но незачем, ибо пипл и так хавает.

Beyondtheclouds 20 сентября 2018 в 16:51

+35

обычно люди которые так говорят, не занимались веб разработкой. Такие иллюзии достаточно быстро ломаются об реальность на больших проектах.

geisha 20 сентября 2018 в 21:15

+2

Реальность состоит в том, что никто не хочет и не будет переделывать всё с нуля на больших

проектах даже если это спасёт планету от метеорита. Ещё реальность состоит в том, что с первого раза может получиться только случайно, а в большом проекте — никогда.

Aingis 21 сентября 2018 в 11:31

+1

Ну, почему. Иногда переделывают. Это занимает от нескольких месяцев до нескольких лет, в зависимости от сложности и количества разработчиков. Скажем, потому что поддерживать сайт на XSLT и других древних технологиях уже нет смысла — проще переписать. Естественно, не обходится без багов (никто не совершенен), а к моменту что-то неизбежно уже устарело.

w0den 21 сентября 2018 в 12:17

0

Недавний пример: Github.com отказывается от использования jQuery и переходит на чистый JavaScript.

> По словам разработчиков, процесс ухода с jQuery занял годы.

saboteur_kiev 24 сентября 2018 в 18:18

0

Причем тут «хочет, не хочет».
Кто за это платит? По своему желанию переделываются только пет-проекты.
В случае больших проектов, переделать все это далеко не копейки.

w0den 20 сентября 2018 в 21:23

+15

Я никогда не работал в больших корпорациях, поэтому хотелось бы узнать, почему, например, когда я открываю Facebook.com как неавторизованный пользователь, страница занимает 6.16 MB и браузер отправляет 65 запросов? Наверное, дело в том, что форма регистрации использует AJAX-запросы и там всё «очень сложно»?! Ну, достаточно открыть страницу авторизации с «обычной» формой и полюбоваться «оптимизацией» — 3.75 MB и 37 запросов. Также, например, «оптимизирована» и главная страница Twitter.com — никаких форм с AJAX-запросами, однако размеры страницы составляют 2.41 MB, и браузер отправляет 18 запросов.

Забавно, что даже при редактировании файла на Github.com страница весит лишь 1.63 MB (а там ведь подсветка кода, AJAX-запросы, живой поиск, предпросмотр, и другие полезные функции).

yarric 20 сентября 2018 в 22:52

+18

«Вы ничего не понимаете, веб-программирование — это крайне очень супер сложно и вообще сперва добейся (закончи месячные курсы фулл-стека на пайтоне)».

SirEdvin 20 сентября 2018 в 23:15

+3

Представьте сколько аналитики, поведенческих метрик и рекламы с вас собирают facebook и twitter.
Ну и конкретно на facebook большинство запросов это статика. Как прикажите картинки то грузить?

w0den 21 сентября 2018 в 00:08

+18

На странице авторизации Facebook загружаются 6 картинок: два спрайта и 4 однопиксельных изображений. Общий размер картинок: 33.76 килобайт, когда страница занимает **3.75 мегабайт** и это только для того чтобы отображать форму с двумя полями и одной кнопкой.

Насчёт аналитики — даже Google Analytics и Яндекс.Метрика с Вебвизором вместе взятых не занимают больше 250 KB. Тут проблема не в аналитике, а в том, что разработчики банально не используют условные загрузки. По моему мнению, незачем загружать стили/скрипты комментариев, чата, списка друзей, редактирование профиля и других функций недоступны неавторизованных пользователей. *Но, возможно это лишь мои иллюзии, ведь я не работал в крупных корпорациях.*

Yakud 21 сентября 2018 в 11:26

0

Ну как же зачем?! Прогреть кэши перед входом пользователя на основной сайт, а не форму авторизации. Да, на одну страницу это слишком жирно, но зато после авторизации пользователь уже не будет ждать пока все скрипты загрузятся, особенно с медленным инетом, а сразу начнет поглощать контент.

w0den 21 сентября 2018 в 12:12

+3

А Вы точно веб-разработчик? Даже если Вы правы насчёт этой теории, это, мягко говоря, неправильно. Не говоря уже о том, что после авторизации, страница «нагревается» до 42 MB.

› Заголовок спойлера

esc 21 сентября 2018 в 13:51

0

Почему это неправильно? Если для ФБ ценны в первую очередь те, кто пойдет авторизироваться или регистрироваться, то логично сделать удобнее именно им.

А то, что там после загрузки еще больше, ни о чем не говорит. Вполне вероятно, что на странице логика грузится только те библиотеки, которые делают изначальное отображение интерфейса и общие для всех пользователей. А после авторизации уже происходит фактические наполнение данными, но пользователь сразу видит уже какой-то каркас интерфейса.

Alexeyslav 21 сентября 2018 в 14:07

-1

Если пользователь РЕШИЛ зарегестрироваться, то его не остановят мегабайты первой странички. Может, в первый раз отпугнёт и он передумает но через день-два-неделю он обязательно вернётся и зарегистрируется.
К сожалению, сам на мобильном интернете очень многих постов и картинок просто не увидел — тупо не дождался их загрузки, дольше 10 секунд загрузки — это уже бесполезная трата времени и вещь уходит куда-то далеко в бесконечную ленту событий, возврат к которой в будущем маловероятен.

w0den 22 сентября 2018 в 00:08

+2

Неправильно, по крайней мере потому, что нужно уважать пользователей и помнить, что не у всех топовые устройства и высокоскоростной + безлимитный интернет. Правильно, когда нужное загружается «при необходимости», а не по принципу «а вдруг пользователь захочет каждую минуту загрузить новый аватар, редактировать профиль, отправить сотни эмодзи в чате, сделать селфи, и многое другое, причём всё сразу».

Вы предполагаете, что разработчики заранее кэшируют данные, а я не думаю, что они отличаются от тех, кто подключают jQuery на всех страницах сайтах, когда данная библиотека будет использована только на странице контактов, и то, чтобы анимировать одну кнопку, если пользователь открыл страницу 29 февраля в полночь. Во-первых, если бы они заботились об активных пользователях, они бы использовали условные загрузки по крайне мере после авторизации. Во-вторых, они подключают мегабайтные скрипты на всех страницах. В-третьих, пару мегабайт до авторизации лишь «цветочки» по сравнению, с тем, что происходит после. Если бы Вы проверили Вашу «теорию» о кэшировании, то заметили бы, что после авторизации сайт совсем «не летает», даже если все скрипты загружаются из кэша. Он и не будет, ведь это не SPA, и мегабайтные скрипты выполняются каждый раз, когда пользователь переходит на другую страницу.

Facebook и Twitter лишь примеры, но они далеко не единственные кто не допускают, что пользователь не будет использовать даже 10% функционала. Посмотрите, сколько ненужных стилей, скриптов и медиа файлов загружаются на веб-сайтах; какие права требуют простые приложения; какие спецэффекты используются, дабы рассказать какие «мы современные»; сколько шрифтов подключаются лишь, чтобы «показать красивое название сайта». Практически все с каждым днём «улучшают» свои программы и сайты, но, к сожалению, это происходит только в пресс-релизах.

Напоследок, хотелось бы напомнить, что даже js-файл размером в 1 MB это довольно много для сайта, где пользователи пишут и читают обычные посты. И, тем не менее, Вы хотите оправдать разработчиков, которые подключают десятки мегабайт JavaScript. Боюсь, я не могу согласиться с Вами, поэтому, прошу Вас, не вздумайте переходить на сторону зла!

esc 22 сентября 2018 в 00:29

0

Я напомню, что Фейсбук непосредственно заинтересован, чтобы им пользовалось как можно больше людей и не будет специально делать так, чтобы ухудшить динамику регистраций или логинов.
Еще напомню, что эта компания стоит сотни миллиардов и каждый день ее сервисом пользуется больше миллиарда человек и этого бы не было, если бы они разгильдяйски относились к таким важным элементам, как главная страница сайта.

С вашей стороны очень самонадеянно считать что вы знаете лучше, как надо делать и что решение сделать именно такой механизм работы страницы авторизации у Фейсбука продиктован ленью или некомпетентностью разработчиков.

У них есть огромное количество данных о том, что происходит на сервисе на каждом этапе его работы, данные о сотнях миллионах пользователей (если говорить о веб версии). Равно как и есть возможность протестировать и сравнить множество альтернативных вариантов.

Догрузить пару мегабайт библиотек в фоновом режиме, пока юзер вводит логин и пароль, чтобы после авторизации интерфейс с базовом виде прорисовался как можно быстрее — вполне здравое решение.

Единственный сценарий, при котором такой подход может навредить, это если посетитель не будет авторизироваться, а просто зайдет на страницу. В этом случае загрузятся лишние мегабайты.

Но сервис вполне логично ставит интересы таких посетителей ниже тех, кто авторизацию пройдет.

popov654 22 сентября 2018 в 09:17

+4

> Еще напомню, что эта компания стоит сотни миллиардов и каждый день ее сервисом пользуется больше миллиарда человек и этого бы не было, если бы они разгильдяйски относились к таким важным элементам, как главная страница сайта.

Наивное утверждение. Сервис настолько огромен и популярен, что у существующих пользователей просто-напросто нет выбора (нет аналогов, куда перейти, ведь все друзья в сидят в фейсбуке). А с новыми пользователями то же самое — в мелкую соцсеть типа MySpace идти нет смысла, так как там почти никого нет.

А если вы почитаете, что люди пишут про скорость работы (что веб версии, что мобильного приложения в том же Google Play), то вы увидите, что негативных отзывов гораздо больше, чем позитивных (я правда читал только русскоязычные комментарии, но можно начать хотя бы с России как частного случая). Вы только вспомните: мобильное приложение фейсбука стало настолько тяжёлым, что из него выпилили мессенджер и выпустили отдельным приложением! А потом ещё выпустили Facebook Lite, к слову…

w0den 22 сентября 2018 в 13:07

0

Хотелось бы добавить, что лично я не считаю, что огромное количество денег и пользователей это показатель профессионализма. Скорее это говорит о том, что они хорошие бизнесмены и маркетологи.

Действительно, сверхпопулярные компании могут позволить себе очень многое. И если они изначально работают над привлечением новых пользователей, то став популярными, чаще всего в первую очередь они думают, как извлечь максимальную прибыль. Как монополисты они уверены, что пользователи никуда не убегут, а значит, они могут творить любую дичь. Правда, некоторые из них (например, IE, ICQ, MySpace, Yahoo) поплатились за это.

popov654 25 сентября 2018 в 19:50

0

А что в своё время натворил MySpace? Я туда заходил 2-3 года назад и был в полном восторге — и от дизайна, и от того, насколько сайт был шустрым и минималистичным.

khim 25 сентября 2018 в 21:15

+2

Это сейчас он «шустрый и реалистичный». Потому что у разработчиков нет ресурсов, чтобы рюшечек навешать. А когда Facebook «взлетел» примерно 10 лет назад — всё было наоборот.

Дорога ложка к обеду. Очень хороший пример тут — OS/2 и Windows. Когда в 1992м вышла OS/2, которая требовала 8MB, а на 4MB еле «шевелилась», то народ выбрал вышедшую тогда же Windows 3.1, которая сносно работала на 1MB, а на 2MB «летала». IBM вложила кучу ресурсов и в 1994м выпустил OS/2 Warp, которая более-менее работала на 2MB, а на 4MB — так вообще хорошо. Про неё начали потихоньку говорить, но через полгода — вышла Windows 95, которая требовала уже 4MB, а для приличной работы — 8MB… казалось бы должен был быть провал? Но нет: за 3 года ситуация с памятью изменилась и те 8MB, которые в 1992м казались чем-то запредельным в 1995м уже перешли в категорию «дорого, но терпимо»… а зато у Windows95 была куча программ, которых у OS/2 не было…

dimm_ddr 24 сентября 2018 в 10:05

+1

> Я напомню, что Фейсбук непосредственно заинтересован, чтобы им пользовалось как можно больше людей и не будет специально делать так, чтобы ухудшить динамику регистраций или логинов.
> Еще напомню, что эта компания стоит сотни миллиардов и каждый день ее сервисом пользуется больше миллиарда человек и этого бы не было, если бы они разгильдяйски относились к таким важным элементам, как главная страница сайта.

Я не встречал ни одного человека, который бы сказал что ему нравится дизайн и функциональность фейсбука. Ни в России, ни в Европе. Пользуются им все просто потому что им уже пользуются все и чтобы с него уйти нужно, во-первых, найти куда, а так как все пользуются ФБ, то адекватно развитого ничего и нет, во-вторых, переманить туда всех нужных знакомых, что тоже титанический труд, человеку проще ничего не менять и плеваться чем напрягать мозг и совершать какие-то действия, а в-третьих, нужно придумать что делать с новыми знакомыми и всякими группами которые тоже будут по умолчанию в ФБ. Так что пока этим поделием в принципе можно будет пользоваться — им будут пользоваться все. Это никак не говорит о том что они знают что делают. Более того я знаю больше одного человека, которые из-за всего этого бреда в ФБ отказались от него несмотря на все перечисленные пункты.

khim 24 сентября 2018 в 16:48

+1

На самом деле вы оба правы. Facebook, конечно, знает что делает, но… **не сегодняшний** Facebook. Хочу напомнить, что Facebook — далеко не первая социальная сеть, до него были MySpace, Orkut и *другие*. А знаете что с ними произошло? Их обошёл маленький и шустрый (в те времена) Facebook. А потом… потом менеджеры в Facebook выяснили что «рюшечки» привлекают пользователей! Процесс идёт по накатанной схеме. В результате в какой-то момент вы имеете кучу пользователей, которых привлекли на начальном этапе (и ненавидящих современный Facebook), пользователей которых «зачинатели» привели и тех, кто фанатеет от рюшечек и кому пофигу что они тормозят. Тут вы имеете максимальную капитализацию, аудиторию и прочее. Но эта ситуация неустойчива: в какой-то момент, после очередного раунда «добавления рюшечек» всё начинает тормозить настолько, что люди начинают уходить. И если этого не осознать и добавить «рюшечек» ещё — то всё начнёт рассыпаться: вначале уйдут первые (ценившие скорость), потом вторые (так как ушли первые), а потом и ценившие «рюшечки» не захотят в «пустом доме» жить. Но Facebook'у пока удаётся балансировать «на грани»: периодически он всё-таки занимается оптимизацией своих страниц, в результате такого коллапса, как, скажем, у Orkut'а не происходит.

xakep2011 21 сентября 2018 в 16:44

–2

Крупные корпорации это всегда бюрократия. Не исключено, что какие-нибудь маркетологи решили, что именно страница логина должна принять на себя удар по перформансу и сразу подгрузить всё, что нужно юзеру, чтобы после логина он на доли секунды быстрее увидел свою вожделенную ленту новостей.

aleksandros 25 сентября 2018 в 22:26

0

Можно ещё вспомнить «модемные» времена и Мейл.ру, чья страничка весила килобайт 50. И при этом работала, письма писались, отправлялись. Понятно, что сейчас там и форматирование и всякая живая загрузка. Но, страница выросла мегабайт до 5, если не больше, то раз в 100. Но удобство (как ни измеряй;) и скорость в 100 раз не выросли.

YemSalat 21 сентября 2018 в 08:01

+1

Че-то вы загнули… 1.1МБ на странице авторизации ФБ

▸ Заголовок спойлера

w0den 21 сентября 2018 в 09:07

0

Chrome показывает размер сжатых данных. Активируйте «Use large request rows», чтобы увидеть реальный размер.

▸ Заголовок спойлера

YemSalat 21 сентября 2018 в 12:44

+1

А зачем оценивать «разжатые» данные?
Это подмена понятий, ваш изначальный комментарий негодовал о:

> почему, например, когда я открываю Facebook.com как неавторизованный пользователь, страница занимает 6.16 MB и браузер отправляет 65 запросов

Вам эти данные приходят в сжатом виде и весят никак не 6 с лишним мегов.
1МБ на страничку, учитывая шрифты/иконки/картинки — это более менее.
На мобильной версии приходит около 300кб на всю страницу.

Так что не сгущайте краски.

Ogra 21 сентября 2018 в 12:46

+7

Ну так «занимает» же, а не «качается». И эти мегабайты яваскрипта еще надо распарсить и выполнить.

justboris 22 сентября 2018 в 14:19

0

Вот именно, HTML/JS/CSS еще парсится и интерпретируется. Результат займет в памяти еще больше места.

Цифра в 6 мегабайт не характеризует ничего, в отличие от сжатого размера, который реально передавался по каналу пользователю.

Alexeyslav 24 сентября 2018 в 08:46

0

Для пользователей с широким каналом от этого ни холодно ни жарко, сжатие разве что помогает разгрузить магистральные каналы. Имеет значение именно не сжатые данные, они ведь занимают реальную память. А интернет-канал это дело вторичное, и количество трафика играет роль только при жестких ограничений на сеть — мобильные сети и т.п. таким образом сжатие играет вторичную роль, решая только одну проблему из множества.

justboris 24 сентября 2018 в 10:53

0

> Имеет значение именно не сжатые данные, они ведь занимают реальную память.

Это нет так. Реальную память занимает "результат" исполнения кода, а не исходный текст. Короткий код может легко занять много памяти, достаточно насоздавать много объектов в цикле.

Alexeyslav 24 сентября 2018 в 13:52

+1

Если это делает ДАЖЕ короткий код, сложно представить на что способно большое количество кода.

khim 24 сентября 2018 в 16:58

0

На самом деле, как ни странно, сложно. В частности в одной из программ, в которой я переделал один модуль, сложную систему объектов была заменена на заранее просчитанную таблицу. Модуль на 200К стал модулем на 1М, но зато таблица на пару мегабайт перестала строиться при запуске — что улучшило время запуска (сильно), скорость работы (заметно), и потребляемую память (вдвое). Уменьшился даже запакованный образ дистрибутива, пострадало только место на диске (но это для нас был самый некритичный параметр).

justboris 24 сентября 2018 в 17:22

+1

> сложно представить на что способно большое количество кода

Еще как сложно! Там могут быть оптимизации, ленивые вычисления, которые сэкономят память.

Гадать о производительности кода по его размеру — такое себе занятие. Именно поэтому размер сжатого кода еще имеет какое-то значение, то разжатого – цифра ни с чем ни связанная.

w0den 21 сентября 2018 в 13:02

+4

Вы думаете, что все пользователи получают данные в сжатом виде? Или, что сжатие даётся пользователям даром и мегабайты хлама никак не влияют на производительность?

YemSalat 21 сентября 2018 в 16:35

0

> Вы думаете, что все пользователи получают данные в сжатом виде?

У 99% браузеры поддерживают сжатие.

> Или, что сжатие даётся пользователям даром

Да, почти даром.

> мегабайты хлама никак не влияют на производительность

Вот это уже под вопросом. Возможно, как написали выше, они грузят что-то заранее, чтобы потом не «лагало» и не погружало ничего.

popov654 21 сентября 2018 в 17:02

+2

Даже если разжатие не сильно нагружает процессор (скорее всего, это так), память всё-таки критична. Например, у меня в телефоне браузер крашится, если веб-страница пытается съесть больше 12-13 мегабайт (с учётом графики, шрифтов и всего-всего). А тут в мобильной версии сколько, говорите, после разжатия будет? Уже мегабайта 2-3, а это ещё без фотографий и текстов. А когда пользователь залогинится — тогда самый треш и начнётся. Там ещё и доп. стили и скрипты прилетят, и реклама с логикой её показа.

YemSalat 21 сентября 2018 в 18:54

–2

> Например, у меня в телефоне браузер крашится, если веб-страница пытается съесть больше 12-13 мегабайт

Он крашится если вкладка загружает больше 13 мегабайт? Чего? Диска или памяти?
Одни фотки сейчас весят по 2-3, если не больше. Может вам надо обновить телефон?

muon 24 сентября 2018 в 07:49

0

А не поздно писать обличающие статьи, если потребители уже считают происходящее нормой?..

popov654 25 сентября 2018 в 19:59

+1

Зря вас заминусовали. Да, телефон надо обновить, вы правы. Эта модель вышла летом 2011-ого. Но парадокс в том, что вплоть до весны-лета 2013-ого года всё работало более-менее сносно (в 2012-ом так вообще летало). Вы полагаете, тогда фотки весили сильно меньше? За счёт чего?

Оффтопик

esc 21 сентября 2018 в 13:48

0

FB скорее всего так делает, чтобы заранее прогрузить в кэш или даже запустить выполнение тех скриптов, которые потребуются после авторизации.

Видимо, посчитали что лучше пусть у тех, кто просто открыл и не авторизировался, делались лишние запросы, чем заставлять дополнительно ждать загрузки тех, кто таки авторизировался с этой страницы.

serf 21 сентября 2018 в 16:26

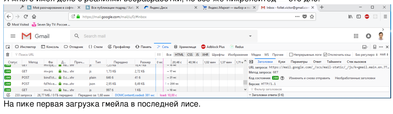
0

Полагаю потому что строчить блог посты и хайпиться другими способами, и решать реальные задачи это разные вещи.

DarthVictor 21 сентября 2018 в 00:11

+2

Я много имел дело с реалиями вебразработки, но все же пикрелейтед — это дно.



На пике первая загрузка гмейла в последней лисе.

dartraiden 21 сентября 2018 в 01:55

+2

Возможно, потому что в лисе нет поддержки ShadowDOM.

DarthVictor 21 сентября 2018 в 02:09

0

Вполне возможно, в Хроме 6MiB.
То есть полифил на 20MiB(!).

dartraiden 21 сентября 2018 в 02:11

–3

> Но ведь это не вина гугла, что в Firefox не реализовали ShadowDOM v0 (точнее, пытались, но он пока страшно далёк от завершения, а пока реализовывали, сама спецификация v0 стала deprecated, так что силы брошены на ShadowDOM v1).

Ogra 21 сентября 2018 в 05:11

+12

> Но ведь это не вина гугла, что в Firefox не реализовали ShadowDOM v0

Embrace, Extend, Extinguish нового поколения.

DarthVictor 21 сентября 2018 в 10:26

+6

Это либо вина либо менеджмента гугла в плохой поддержке браузеров с 10% рынка суммарно, либо разрабов которые пишут код на плохо полифилящихся технологиях, которые только один браузер поддерживает, либо и тех, и других.
А разгадка проста — из-за сильной интеграции с другими сервисами гугла, в первую очередь с андроидом, гмейлом пользуются все и писать они могут практически любой говнокод. Сравните даже хотя бы с яндексовой почтой.

НЛО прилетело и опубликовало эту надпись здесь

sumanai 21 сентября 2018 в 12:02

+5

> Во-вторых, гугл тем самым форсирует более быструю имплементацию стандартов у всех.

Используя устаревшую версию стандарта? Согласен с первым утверждением, но второе не верно.

popov654 21 сентября 2018 в 17:09

+4

Да не стояло там ничего на месте. Во-первых, была прекрасная и весьма шустрая Опера (2004-2008 если берём). Во-вторых, Firefox был не так уж плох (хотя и медленней, но не медленней, чем IE, а ещё намного безопаснее и функциональнее). Так что конкуренты не просто были, они развивались. И IE тоже развивался. Вспомните 5.5 в Win2k, и сравните с 7.0 и 8.0. Совсем разные вещи

НЛО прилетело и опубликовало эту надпись здесь

popov654 22 сентября 2018 в 09:24

+2

> Почему это был не его изобретатель

Изобретатель чего, AJAX?

Расскажите, это всё очень интересно. Мне тогда и в самом деле было 10-15 лет, и разработкой я тогда ещё не занимался, только читал книжки (года так с 2005-2006).

> на ряду с полным месивом в поддержке существующих

Имхо, не было там особого месива. У меня до сих пор стоит учебник по JavaScript 2004 года издания на полке. Так вот, 60-70 процентов вещей, которые там описаны — кроссбраузерны на сто процентов (да-да, тот самый ES3). Остальное — в двух вариантах, для IE и для Firefox. Opera же, заняв очень мудрую позицию, часто поддерживала оба варианта (окей, не оба, почти всегда вариант IE; но начиная с момента, когда появился Хром, а в Хроме и Firefox подходы были более близки к друг другу и к стандарту — ситуация поменялась, и Opera стала делать как Chrome и Firefox, а синтаксис IE иногда поддерживался в придачу, «для совместимости» — как пример, свойство document.all, которого кроме IE нигде больше не было, а в Opera до версии 11.50 оно всё ещё работало).

Source 22 сентября 2018 в 20:18

+1

Месиво всё-таки было и именно благодаря ему обрели популярность Prototype, MooTools и jQuery. Они брали на себя большую часть боли по поддержке особенностей браузеров. Отдельной историей было ещё и месиво с реализациями CSS.

popov654 25 сентября 2018 в 20:02

0

А что именно было с реализациями криво? Про CSS тех лет я почти ничего не знаю. Ну то есть знаю поведение IE7-8, немного — IE6 (тестил на нём один свой проект ради спортивного интереса). Остальное мимо меня прошло.

VolCh 26 сентября 2018 в 10:15

0

Больше всего запомнилось, что размеры блока разные браузеры по разному понимали. Современным языком говоря было разное значение box-sizing и управлять им нельзя было. А pixel perfect тогда было обычным требованием, буквально принимали вёрстку диффом скриншота и макета

НЛО прилетело и опубликовало эту надпись здесь

Goodkat 22 сентября 2018 в 14:27

+2

В лидеры Хром выбился благодаря монопольному положению Гугла на рынке поиска и агрессивному маркетингу.

НЛО прилетело и опубликовало эту надпись здесь

sumanai 23 сентября 2018 в 01:25

+4

> Хром на момент появления и в последствии действительно превосходил все, что было.

Не во всём и не для всех. У него как был дубовый, не настраиваемый интерфейс, так и остался.

ganqqwerty 23 сентября 2018 в 23:14

−1

значит они вовремя поняли, что интерфейс должен быть не настраиваемым, пошли так сказать по пути Эпла

sumanai 23 сентября 2018 в 23:16

0

Просто большинству плевать, а на меньшинство плевать большинству.

ganqqwerty 24 сентября 2018 в 19:05

−1

вроде ж справедливо

shurricken 24 сентября 2018 в 08:13

0

ээо… дефолтный интерфейс должен быть удобным… а все эти " в новой версии мы вам дадим мильон новых тем" вызывают тяжелый вздох…

domix32 21 сентября 2018 в 12:42

0

Так его уже объявили deprecated

popov654 21 сентября 2018 в 17:04

+6

Вопрос — а нафига он вообще нужен? Без него интерфейс просмотра писем не сделать?

muon 24 сентября 2018 в 07:54

0

Конечно не сделать. А SquirrelMail вообще никогда не существовало.

novice2001 21 сентября 2018 в 08:57

+12

Для того, чтобы судить о вкусе блюд, необязательно быть поваром.
И не надо про эти ваши «большие проекты». Реальность в том, что именно «большие проекты» в большинстве случаев нафиг не нужны, их лепят вместо простых сайтов с текстом и картинками

потому что это «стильно, модно, молодежно», а не потому что необходимо.

Druu 21 сентября 2018 в 12:15

+3

> а не потому что необходимо.

Если бы это не было необходимым, то простые сайты с текстом и картинками выигрывали бы в конкурентной борьбе, но мы наблюдаем обратную ситуацию — выигрывают те самые "большие проекты".

Так уж вышло, что в 2018 разница между десктопным приложением и сайтом в интернете стала бесконечна мала, можно сколько угодно по этому поводу плакаться, но фарш провернуть назад не удастся.

Можете себе свой гипертекстовый фидонет замутить и там делать свои страницы с простым текстом и картинками. В вебе поезд ушел.

novice2001 21 сентября 2018 в 12:26

+5

> Если бы это не было необходимым, то простые сайты с текстом и картинками выигрывали бы в конкурентной борьбе

Если бы были сайты с равноценным контентом, но в вариантах с рюшечками и без — то так бы и было. Но в погоне за модой сайты с хорошим контентом превратились в монстров. А одной скорости для выигрыша в борьбе мало.
И нет, разница между сайтами и приложениями никуда не делась.

Druu 21 сентября 2018 в 12:38

+3

> Если бы были сайты с равноценным контентом, но в вариантах с рюшечками и без — то так бы и было.

Так этих сайтов нет, потому что они проиграли конкурентную борьбу.

> А одной скорости для выигрыша в борьбе мало.

Ну вот видите, вы сами ответили. Производительность должна быть достаточной, прирост производительности выше чем нужно — уже не дает никакой пользы. И если клиента устраивает производительность на уровне 1% от возможной, то ПО работающее на уровне этого 1% будет иметь конкурентное преимущество над тем, что работает на уровне 90%. Потому что пользователю на эту разницу плевать, а вот затраты на разработку — возрастут.

novice2001 21 сентября 2018 в 13:14

+4

> Так этих сайтов нет, потому что они проиграли конкурентную борьбу.

Нет, они проиграли борьбу не потому что на них не было рюшечек, а потому что современные монстры раньше были нормальными легкими и быстрыми сайтами, их аудитория в значительной степени сформирована за много лет, а поведенческие факторы позволяют и сейчас удерживаться в топе поисковиков.

> Ну вот видите, вы сами ответили.

Нет, просто вы меня поняли так, как хотели понять.
Дело не в том, что если сайт грузится за 10 секунд, то это нормально и быстрее не надо. Дело в том, что между сайтом за 10 секунд с хорошим контентом и сайтом за 1 секунду с плохим любой нормальный человек выберет первое. Зато между сайтами с хорошим контентом но загрузкой в 10 и 1 секунду — второе. Это не та разница, на которую можно наплевать.
И дело, кстати, не только в загрузке, но и в последующем скроллинге и т.п.

Druu 22 сентября 2018 в 05:08

0

> Зато между сайтами с хорошим контентом но загрузкой в 10 и 1 секунду — второе.

Если бы это было так — мы бы видели в топе сайты с хорошим контентом и загрузкой в секунду. Но это не так.

> а потому что современные монстры раньше были нормальными легкими и быстрыми сайтами

Но их потом взяли и специально сделали ненормальными, тяжелыми и медленными? Зачем? Заговор рептилоидов?
Я серьезно спрашиваю. Как вы объясняете этот парадокс?



popov654 22 сентября 2018 в 09:30

0

Я бы объяснил чрезмерным переусложнением логики с одной стороны (но усложнение нужно и порой неизбежно), тем, что мониторы стали больше и графика потяжелела, плюс стало больше видеоконтента и анимаций — с другой (в этом тоже есть плюс, ибо красиво и людям нравится), и с третьей стороны — большой ленью разработчиков, которые часто вместо того, чтобы заюзать что-то самостоятельно, тянут сторонний фреймворк, половина возможностей которого им вообще не понадобится (может быть понадобится в перспективе, но не сегодня и даже не завтра). А во фреймворке ещё куча своих подкомпонентов. А фреймворков на крупных сайтах часто 2-3. Плюс на больших сайтах бэкенд тоже обычно отвечает далеко не молниеносно (взять тот же Twitch). Плюс объём пересылаемых на сервер и обратно данных, который растёт как минимум линейно вместе с усложнением логики.

Избежать всего этого можно — если делать простые и не очень красивые сервисы с небольшим количеством функций. «Дёшево и сердито». Но на такое меньше спрос, людям это уже приелось, им чего-то нового хочется. А за новое порой приходится платить тормозами, увеличившимся трафиком, необходимостью обновлять браузер и даже ОС...

novice2001 22 сентября 2018 в 09:35

+1

Тем, что модно и тем, что можно (в смысле «гляди, как мы могЁм»). Туда же и регулярные редизайны.
И не надо говорить, что «компании все просчитывают перед тем как сделать». Ярчайший пример эпик фейла — Кинопоиск 2.0. Просто если изменения не такие резкие, то пользователи просто потихоньку привыкают, но это вовсе не значит, что они в восторге от всего этого.

Druu 22 сентября 2018 в 09:39

0

> Тем, что модно и тем, что можно (в смысле «гляди, как мы могЁм»). Туда же и регулярные редизайны.

Модно и можно кому? Вы же понимаете, что решение принимается не программистами? Зачем бизнес в это деньги вкладывает, если это ничего не дает?

novice2001 22 сентября 2018 в 09:46

0

Потому что бизнесу кажется (ключевой момент), что так будет лучше («патамушта какнкуренты так сделали, давай и мы сделаем»). Но «кажется» и «будет» — разные вещи.

Druu 22 сентября 2018 в 15:17

+1

> Потому что бизнесу кажется (ключевой момент), что так будет лучше («патамушта какнкуренты так сделали, давай и мы сделаем»). Но «кажется» и «будет» — разные вещи.

Почему-то успешным оказывается именно тот бизнес, которому так кажется. А которому не кажется — успешным не оказывается.
Видимо, "кажется" обоснованно, в итоге.

Am0ralist 23 сентября 2018 в 13:48

0

попытка ошибкой выжившего оправдать неверную логику?

Druu 24 сентября 2018 в 09:24

0

> попытка ошибкой выжившего оправдать неверную логику?

При чем тут ошибка выжившего? И логика вполне верная.



Am0ralist 24 сентября 2018 в 11:54

0

Ну, тогда вы должны доказать непосредственную связь между кажется и успешным.
Потому что между успешным/неуспешным и кажется/не кажется могут любые пропорции быть.
Например, что большинство бизнеса, которому казалось — стало не успешным. Или что куча бизнеса, которому не казалось — стало успешным, после чего начало казаться.
А так, у вас есть условный успешный бизнес, которому сейчас что-то кажется. И он может в любой миг стать не успешным (яху, нокиа). И как они стали успешным? От того, что им казалось или нет? И сколько прочих бизнесов, которым казалось, не стали успешными

Druu 26 сентября 2018 в 13:41

0

Еще раз — у нас, фактически, нет успешного бизнеса, которому не кажется (или есть, но это какие-то единичные и малоизвестные случаи).
Если бы данный фактор не влиял на результат, то мы бы не наблюдали такую ситуацию.



Am0ralist 26 сентября 2018 в 14:56

0

> Еще раз — у нас, фактически, нет успешного бизнеса, которому не кажется (или есть, но это какие-то единичные и малоизвестные случаи).

Доказывайте.
Например, для начала докажите, что все те успешные бизнесы изначально были из тех, кому казалось. А не те, которым не казалось, а когда *достигли* успеха — стало казаться.

Druu 26 сентября 2018 в 15:29

0

> Доказывайте.

Зачем? Это очевидное для любого пользователя интернетов в 2018 наблюдение. С-но, именно это наблюдение стало причиной появления обсуждаемой статьи. И 2к комментов к ней. Вы сейчас будете придуриваться и делать вид, что всего этого нет? Ну если нет — то и обсуждать нечего, сайты быстрые, весят мало.

> А не те, которым не казалось, а когда *достигли* успеха — стало казаться.

А какая разница, стало или было? Это совершенно несущественный фактор.



Am0ralist 26 сентября 2018 в 16:24

0

> Почему-то успешным оказывается именно тот бизнес, которому так кажется. А которому не кажется — успешным не оказывается.
> Видимо, «кажется» обоснованно, в итоге.

> А какая разница, стало или было? Это совершенно несущественный фактор.

А давайте позовем дедушку КО?
Итак, если бизнес *достиг* не думая, сейчас сидя на достигнутом стал думать, то успех и думанье за — никак не связаны.
Более того, это совершенно не означает, что завтра эти думальщики не вылетят в трубу, а место их не займут другие.
Которые тоже не думали, но достигнув вершины станут думать и… цикл замкнется.

На что я вам и указываю. Вы видите кого-то, кто сидит на вершине и СЕЙЧАС поступает так. Как это доказывает, что он именно так поступал, дабы там оказаться? Никак. Более того, вы не можете утверждать, что своим поведением сейчас он не шагает к пропасти. Потому что куча контор, которые сидели на троне многие годы, вдруг пролетали мимо поворота и вылетали. И как это зависит от думанья или нет — у вас нет фактов однозначных.
То, что на рынке любят культ карго — не секрет. Только это не означает, что его почитатели смогут полететь вверх.

Druu 26 сентября 2018 в 16:51

–1

> Итак, если бизнес *достиг* не думая, сейчас сидя на достигнутом стал думать, то успех и думанье за — никак не связаны.

Так никто и не говорит, что успех связан. Речь не о том.

Речь о том, что либо те, кто занимается успешным бизнесом — умные люди, и, с-но, если они считают правильным вешать на сайты свистоперделки, а Am0ralist не считает, то это потому, что они знают что-то такое, чего не знает Am0ralist.
Есть второй вариант — на самом деле отбор отрицательный, с-но условный Ларри Пейдж — тупой. Именно потому что он тупой, он и стал миллиардером (и остальные тоже). Из-за своей тупости он любит свистоперделки. А вот Am0ralist — он умный. По-этому он свистоперделки не любит, он знает, что все это просто каргокульт. И по той же причине миллионов Am0ralist не заработал — для этого надо быть тупым. Как Ларри.
Ну и третий вариант — все это просто такое большое совпадение.

Вариант по душе выбирайте уже сами.



Am0ralist 26 сентября 2018 в 17:42

0

О, переход на личности так быстро.
Ларри Пейдж — умный, да. Вот только когда все это начиналось и когда он управлял напрямую — были давно.
Не говоря уже о том, что сейчас у него могут быть когнитивные искажения уровня, раз я самый умный, то лучше других всё знаю.

Не говоря уже о том, что ваши рассуждения разбиваются о примеры Нокиа и Яху. Там тоже сидели умные люди, которые зарабатывали небось миллиарды. И которые с усмешкой посматривали на конкурентов, точно зная, что нужно потребителям. Надеюсь пример вам будет понятен, так как общаться с людьми, которые любят переходить на подобную демагогию, когда стараются выставить собеседника идиотом сравнения умного и богатого Ларри с анонимным пользователем хабра — удовольствие ниже среднего



ValdikSS 22 сентября 2018 в 10:29

+1

> Зачем бизнес в это деньги вкладывает, если это ничего не дает?

За 4 года Webmoney сделали редизайн, по-моему, 6 раз, или даже 7. Зачем — непонятно.



third112 25 сентября 2018 в 14:28

0

> За 4 года Webmoney сделали редизайн, по-моему, 6 раз, или даже 7. Зачем — непонятно.

Не только Webmoney, но и много крупных банков и магазинов часто меняют дизайн. ИМХО это сильно раздражает постоянных клиентов, которые привыкли к прежнему. ИМХО многие начальники, принимающие решения, просто не знают основ психологии, а исполнителям нужно показывать свою необходимость делая ненужную работу.

khim 25 сентября 2018 в 16:53

0

> ИМХО многие начальники, принимающие решения, просто не знают основ психологии

Они как раз либо знают, либо догадываются.

Просто у них нет задачи понравится всем и каждому. И если старые пользователи ворчат, но не уходят, а новых сколько-то появляется — то цель достигнута, можно получить бонус.

popov654 25 сентября 2018 в 20:09

0

Но новые появились бы и без смены дизайна, разве нет?

Я в курсе, что есть новые тренды. Но тут получается массовый «вирус» — 25-30 процентов сайтов делают редизайн (не обязательно самые крупные), они становятся «модными и современными», и остальным уже как-то неловко оставаться со старым — в частности, они боятся, что новые клиенты начнут не так охотно прибывать :)

khim 25 сентября 2018 в 20:43

0

> Но новые появились бы и без смены дизайна, разве нет?

Статьи на Хабре и в других изданиях, шумиха в разных соцсетях и прочем. Ну а дальше — работает правило «не важно что о тебе говорят, лишь бы фамилию правильно называли».

Конечно тут важет факт редизайна, а не его тормоза, но… тут уж как получается…

> Но тут получается массовый «вирус» — 25-30 процентов сайтов делают редизайн (не обязательно самые крупные), они становятся «модными и современными», и остальным уже как-то неловко оставаться со старым — в частности, они боятся, что новые клиенты начнут не так охотно прибывать :)

И они, к сожалению, правы. Мода тоже часто заставляет носить одежду, которая откровенно неудобно, а иногда и опасна для здоровья. Но ведь носят. Тоже самое и тут.

> Но тут получается массовый «вирус» — 25-30 процентов сайтов делают редизайн (не обязательно самые крупные), они становятся «модными и современными», и остальным уже как-то неловко оставаться со старым — в частности, они боятся, что новые клиенты начнут не так охотно прибывать :)

И они, к сожалению, правы. Мода тоже часто заставляет носить одежду, которая откровенно неудобна, а иногда и опасна для здоровья. Но ведь носят. Тоже самое и тут.

Причём как раз некоторый парадокс в том, что айтишникам это очень тяжело понять, так как очень многие их них скорее готовы прослыть «немодными», тем мучиться с неудобной одеждой… но ведь людей, «следящих за модой» — куда больше.



third112 26 сентября 2018 в 07:55

0

> Мода тоже часто заставляет носить одежду, которая откровенно неудобна, а иногда и опасна для здоровья.

ИМХО с модой сложнее см. Википедию:

> Сегментами рынка модной индустрии являются категории, на которые подразделены различные марки и бренды, в зависимости от своих параметров — качества изделий, способа выпуска коллекций и ценовой политики производителя

Есть «Высокая мода», «Средний ценовой сегмент», «Массовые марки». В свою очередь в них есть под-сегменты.



third112 26 сентября 2018 в 07:40

0

> Просто у них нет задачи понравится всем и каждому.

Не всем и каждому, а удержать постоянных клиентов. Нпр., если клиенты массово забирают вклады из банка — он может рухнуть.

> И если старые пользователи ворчат, но не уходят,

Обычно уходят. В развитой экономике есть большой выбор. И внутри одной структуры часто бывает выбор. Нпр., операции по банковскому счету можно делать он-лайн, а можно в отделении. Если клиентам неудобно он-лайн, они будут использовать отделения банка загружая работников, а эффективность сайта упадет. Многие магазины торгуют не только по сети. Будет больше покупок вживую, больше заказов по телефону, а прибыль через сайт упадет.

aleksandros 25 сентября 2018 в 22:48

0

Недавно АкБарс Банк замутил редизайн интернет-банка. Был довольно простой и кондовый как топор. Стал весь такой плоско-модный, с ненужными анимациями, блеклыми надписями, странными мелкими размерами шрифта (и как ни странно, чем важнее информация, тем мельче) и как следствие огромными пространствами между строками. Тихий ужас. К счастью, сделали кнопочку «Старый дизайн», но наверно скоро уберут.



0xd34df00d 22 сентября 2018 в 16:56

+1

Потому что нет бизнеса как принимающей решения сущности, есть конкретные люди, и чем компания больше, тем больше этот разрыв.

Конкретный манагер просто хочет выслужиться и иметь красивые слова для написания в отчётах, манагер этого манагера — 50-летний чувак, которому вообще уже на всё плевать, дайте до пенсии доработать, ну и так далее.



third112 26 сентября 2018 в 08:02

0

В развитой экономике при достаточной конкуренции малая и средняя фирмы не выживут при плохих решениях. К сожалению, большие компании, которые почти монополисты, могут позволить себе значительные убытки от неоптимальных решений.



0xd34df00d 29 сентября 2018 в 02:11

0

С другой стороны, компании, которые почти монополисты, могут позволить себе инвестировать в фундаментальный ресёрч. А это уже совсем не к сожалению.

И если государство не начинает бегать и спасать монополии, о которых говорите вы, то в монополизации с таковым исходом я особо ничего плохого не вижу.



third112 29 сентября 2018 в 07:47

0

> Встречал в литературе мнения, что ОС Dr DOS была в свое время лучше, чем MS DOS, что C++ не стал бы столь популярным, если бы MS Pascal не сильно уступал турбо паскалю и VBA бы не было. IE был не лучший браузер. MacOS 7.x сильно превосходила W-95 и W-98, но тут в случившемся кризисе виновата прежде всего Apple Computer, Inc... Нпр., ценовой политикой: мышь для мака стоила целых $60US, в то время как мышь для PC только 8.



third112 29 сентября 2018 в 07:57

0

PS Слышал мнения, что большой успех Явы был обусловлен мощным PR, а не только достоинствами. Однако, все подобные утверждения непроверяемы: история не имеет сослагательного наклонения.



XXXXPro 26 сентября 2018 в 18:30

0

Попробую ответить на ваш вопрос. Причин тут несколько:
1) решение «добавить вот эту рюшечку» обычно принимается не программистом, а каким-нибудь продажником или маркетологом (по крайней мере, в случае малого и среднего бизнеса, а не контор уровня Facebook), большинство из которых — это гуманитарии из тех, что называют процессором системный блок. Соответственно, о том, что это повлияет на скорость, они просто не думают.
2) у программиста, реализующего это решение, обычно не хватает либо квалификации (просто не знает, как сделать ту же асинхронную загрузку или догадаться повесить подгрузку скрипта на подходящее событие), либо времени (когда задача ставится в духе «это надо было сделать вчера, а ты тут собираешься еще два дня с оптимизацией возиться»), либо мотивации («зачем тратить усилия и оптимизировать, если все равно никто этого не оценит толком»).
3) выросло поколение пользователей, которое толком не знает, что такое по-настоящему быстрые сайты, и сложившуюся ситуацию воспринимает как норму (поэтому и не уходит к конкурентам, даже если они есть).
4) отношение к пользователю как к слепому и умственно отсталому существу, которое неспособно самостоятельно найти кнопку «подписаться» или «задать вопрос» и его необходимо потыкать мордой в баннер на полэкрана.
5) всеобщее убеждение, что программа или сайт должны постоянно обновляться вместо «хорошо сделанное и выполняющее свои задачи ПО в обновлениях не нуждается» (кроме разве что исправления уязвимостей в безопасности, если таковые будут найдены).

popov654 21 сентября 2018 в 17:11

0

Имхо, никто не мешает публиковать хороший контент и быстрым сайтам без рюшечек (такие есть, и это не только старенькие форумы в классическом дизайне)... Другое же дело, что таких сайтов очень мало.

sumanai 21 сентября 2018 в 20:17

0

> Имхо, никто не мешает публиковать хороший контент и быстрым сайтам без рюшечек

Толку то. Поисковики жалуют оригинальный контент, и по возможности определяют его и показывают выше. А до второй страницы гугла всё равно никто не доходит.



popov654 22 сентября 2018 в 09:33

0

Так не поиском единым. В том-то и дело, что даже очень хороших и красивых сайтов с рюшечками — больше чем на одну страницу выдачи, и на первую страницу все попасть не смогут. Кто-то всё равно проиграет в этой борьбе.

А те, кто без рюшечек — должны искать посетителей иначе. Своё ядро преданных пользователей, сарафанное радио. Я вот вспоминаю, многие паблики в вк вроде «Подслушано» в начале своего пути так развивались. А вовсе не через поиск, рекламу или оплаченные посты в других сообществах.



third112 25 сентября 2018 в 14:35

+1

Солидные научные сайты (нпр arxiv.org) и Википедия без рюшечек. ИМХО наука должна делать погоду в IT. К сожалению, не все берут пример с научных сайтов.



XXXXPro 26 сентября 2018 в 18:34

0

Согласен с вами целиком и полностью! Жаль, плюсануть кармы не хватает. Впрочем, это частный случай более глобальной проблемы «жаль, что наука и научный подход лишь в малой степени определяют жизнь людей».



0xd34df00d 21 сентября 2018 в 20:10

+3

Выигрывали бы при прочих равных.

Фейсбук выиграл не потому, что у него пердящая форма авторизации. Гмейл выиграл не потому, что он свистит на компьютере пользователя. Ну и так далее.

А всякие сайтики поменьше, типа какого-нибудь производителя телефонов или, не знаю, рюкзаков, тоже выигрывают не за счёт спецэффектов.

А как же, господи, как же я ненавижу блоги с этими модными новыми адаптивными вёрстками, которые сделаны так, что на планшете половину экрана занимает неубирающийся, нескролляющийся, неинформативный блок с контактами, разделами и мордой автора.

khim 21 сентября 2018 в 21:04

0

> Гмейл выиграл не потому, что он свистит на компьютере пользователя.

Как раз за счёт этого. В 2004м, когда он появился большая чем почты в браузере была на обычных формах и потому работала гораздо медленнее, чем AJAX-подобный GMail…

Конечно в современных реалиях быстрого интернета и многомегабайтных страниц GMail'а разница невелика — но тогда GMail был меньше, а «обычная почта» — тормознее.



0xd34df00d 21 сентября 2018 в 21:12

+1

Я не пользовался им в 2004-м, но вот в 2006-м у меня уже был там инвайт, и я отчётливо помню, как именно и вокруг чего именно формировался хайп.

AJAX того уровня у меня на моих машинах того времени (а тогда я был школьником, едва начавшим получать свой доход в несколько тысяч рублей месяц, и комплюктеры у меня были не очень) почему-то не тормозил. Сегодняшнее гмыло у меня тормозит на сегодняшних окологотоповых машинах, при этом из каких-то ~~нетривиальных~~ новых фич фронтенда — instant actions (или как оно там), и всё.

И оно не свистело. Оно было удобно и по делу.

khim 22 сентября 2018 в 06:03

0

> И оно не свистело. Оно было удобно и по делу.

Так всегда бывает. Вначале добавляются фичи, которые реально удобны (AJAX позволил резко снизить задержки), а потом, через несколько лет — получаем монстра.

Во многом и Chrome был рождён из-за того, чтобы GMail-подобные сайты можно было писать не опасаясь добавить пару строк JavaScript'а… ну а потом, как и везде — оптимизаторы V8 упёрлись в возможности железа, а астронавты продолжили решать несуществующие проблемы…



popov654 22 сентября 2018 в 09:39

0

Так оптимизаторы упёрлись в попытках оптимизации. Вопрос, зачем было пихать столько фич и свистелок, чтобы это в итоге начало тормозить? Причём это вопрос не только к Хрому. А также к авторам всё новых и новых стандартов (которые часто сначала добавляли, потом выпиливали, потом снова добавляли спустя год уже в другой версии).

Только вспомнить, сколько всего редко используемого (и в целом опционального) было напихано: Flex, CSS Grid, WebSQL (на страницах MDN как-то осторожно пишут, что лучше этим не пользоваться, поскольку разработка стандарта была приостановлена на стадии драфта), SPDY/HTTP2, WebRTC, WebAudio с кучей фильтров, больше подходящих для работы со звуком в проф. аудиоредакторе типа Nuendo/Cubase, чем для применения на сайтах или в JS игрушках (который сначала добавили, потом убрали, потом снова добавили в сокращённом варианте, но только в Chrome, в Firefox не стали).

sumanai 22 сентября 2018 в 10:57

0

Пара из перечисленных вами вещей таки нужна. Flex позволяет наконец нормально верстать, HTTP2 ускоряет загрузку, несмотря на оверхед шифрования. CSS Grid был бы полезным, но он лишь немногим лучше Flex и с более худшей поддержкой, позднее внедрён, поэтому да, сейчас не нужен.

VolCh 22 сентября 2018 в 16:24

0

Флекс и грид для разных вещей, они отлично дополняют друг друга.

Sychuan 22 сентября 2018 в 17:49

+1

> Grid был бы полезным, но он лишь немногим лучше Flex и с более худшей поддержкой, позднее внедрён, поэтому да, сейчас не нужен

Грид совсем для других вещей. И он не с худшей поддержкой. Как раз грид, внедрился во все новые версии браузеров очень даже хорошо, в отличие от флексбоксов, которые внедряли долго и криво. Вообще грид и флекс — это наконец-то возможность нормально верстать сайты.

popov654 25 сентября 2018 в 20:14

−1

> HTTP2 ускоряет загрузку

Тогда почему я так мало где его вижу?

> CSS Grid был бы полезным

Он не то чтобы бесполезен. Он сложен в освоении. По крайней мере для тех, кто пос старинке верстал на блоках.

> Flex позволяет наконец нормально верстать

Вот здесь поподробнее, плиз. Чем до этого было не нормально?

И кстати с корректностью отображения у этой технологии всё плохо. Я молчу про Оперы на Presto, которые не умеют Flex, потому что его ещё не было, и Хромы версий до какой-то там. Но ирония в том, что даже в Firefox дизайн, сделанный на Flex-ах, безбожно корёжится аж до версии 51 включительно (она вышла, на минутку, не так уж и давно). Если кто не верит — можно открыть Twitch с помощью сервиса-эмулятора, позволяющего просматривать сайты в разных версиях браузеров под разными ОС.

Как по мне, при такой вёрстке как минимум меньше контроля над происходящим (хотя всё ещё можно корректировать огрехи расположения с помощью position: relative, но получается, что эти корректировки ещё и должны быть browser specific).

VolCh 26 сентября 2018 в 10:17

0

> По крайней мере для тех, кто пос старинке верстал на блоках.

Дожились, я думал по старинке это на таблицах…

sumanai 26 сентября 2018 в 19:55

0

> Тогда почему я так мало где его вижу?

Лично я вообще не вижу, пока не открою инструменты разработчика.

> Чем до этого было не нормально?

Тупо выровнять блок по вертикали. До флексов это можно было сделать только с хаками и ограничениями.

> Но ирония в том, что даже в Firefox дизайн, сделанный на Flex-ах, безбожно корёжится аж до версии 51 включительно

Верстал на нём версий 20 назад, проблем не встречал. Конкретный сайт не показатель, может, они скриптами вмешиваются, или чушь какую в стилях прописали.

> Как по мне, при такой вёрстке как минимум меньше контроля над происходящим

Больше его там.



gribunin 26 сентября 2018 в 14:07

0

WebRTC, например, единственная возможность получать realtime аудио с сервера сейчас, когда Flash по умолчанию выключен в браузерах. HLS даёт задержки 20-30 секунд что бывает неприемлемо.

Druu 22 сентября 2018 в 05:10

0

> Фейсбук выиграл не потому, что у него пердящая форма авторизации. Гмейл выиграл не потому, что он свистит на компьютере пользователя. Ну и так далее.

Возможно, но мы же наблюдаем четкую корреляцию. Если пердящие формы и свистящие гмейлы сами по себе не являются причиной победы, то этой причиной является, по крайней мере, нечто с ними связанное.

Если "простые и легкие" сайты заменяются на свистящие и пердящие — значит в этом есть какой-то смысл, ведь это все требует каких-то дополнительных телодвижений, согласитесь.

0xd34df00d 22 сентября 2018 в 05:16

0

Примеры старого реддита, википедии, арХива, SO и прочего уже приводили. Корреляция там не очень очевидна на ~~моей выборке~~ взгляд.

Druu 22 сентября 2018 в 05:33

0

Так эти примеры как раз и демонстрируют то, о чем я говорю.
Зачем бы кому-то делать новый дизайн реддита, если для этого нет причин? Значит, причины были.

0xd34df00d 22 сентября 2018 в 06:23

0

Не, для демонстрации того, о чём вы говорите, придётся продемонстрировать появившегося потенциального убийцу реддита с более лучшим дизайном.

masv 23 сентября 2018 в 10:10

0

Переключил GMail почту на упрощённое HTML отображение. Грузится мгновенно. В отличии от стандартного интерфейса, для которого даже прогресс-бар загрузки в виде логотипа сделали.

khim 23 сентября 2018 в 16:07

0

Это всегда так было. Даже в 2004м. Но раньше основной интерфейс, после загрузки, «летал». А теперь… увы…



popov654 21 сентября 2018 в 16:54

+2

Не скажите. Тот же ВК, например, если брать клиентскую часть — практически оригинальный продукт (до редизайна точно, да и после процентов на 70-80).

michael_vostrikov 20 сентября 2018 в 17:20

+5

> Веб — это информация, текст, таблицы, картинки.

Эм, нет. Веб — это *добавление* информации, текста, таблиц, картинок… А для этого нужен и js, и фреймворки, и на фронтенде, и на бэкенде. Предпросмотр нового коммента все-таки удобнее, чем перезагружать страницу отправкой формы и проверять как оно выглядит, не так ли?)

Nimo_tsi 20 сентября 2018 в 17:49

0

Согласитесь, глупо подтягивать всю библиотеку jQuery, чтобы добавить анимацию меню или прокрутки.

OlegMax 20 сентября 2018 в 18:08

+9

Нет.



popov654 21 сентября 2018 в 17:14

0

Аргументируйте. Если вам так нужна анимация или прокрутка, и вы не хотите реализовывать это каждый раз заново — сделайте свою библиотеку. Это будет как минимум компактнее и проще для работы лично для вас. А ещё полезно в целом как разминка для мозга :)

P.S. Никогда не люблю юзать сторонние решение, особенно если они явно избыточны.



PsyHaSTe 21 сентября 2018 в 17:58

+1

Потому что вы обрекаете пользователей на баги, которые давно починены в библиотеках.

Откройте любую статью, посвещенную опенсорсу. Как там хорошо, что много глаз код просматривают и все такое… (заметите под ковер heartbleed). А потом разверните аргументы на 180° и получите перечень проблем, которые возникают от «всего своего». Начиная от качества этого «своего» и заканчивая временем нового разработчика для понимания.

Нет ничего «крутого» в том, чтобы сделать медленну, кривую, работающую только в пачке браузеров реализацию, вместо того, чтоб взять нормальную.



popov654 22 сентября 2018 в 09:52

−1

При чём тут вообще крутость?

> Потому что вы обрекаете пользователей на баги, которые давно починены в библиотеках.

Надо просто писать без багов.

> Начиная от качества этого «своего»

Зависит от того, кто и как реализует (и насколько ему интересно/приятно этим заниматься).

> и заканчивая временем нового разработчика для понимания

Опять же, если код аккуратный, с комментариями (а ещё лучше, self-docemented, не нуждающийся в комментариях) — порог вхождения нового разработчика будет не очень высоким.

> чтобы сделать медленную

В том-то и дело, что как правило, это будет в 2-3 раза быстрее, чем на каком-то фреймворке/движке, как минимум из-за отсутствия лишней логики.

> работающую только в пачке браузеров

Так надо писать так, чтобы работало во всех, а не в пачке! Где я за «пачку» агитировал?

Единственные плюсы фреймворков — это обход особенностей и багов старых браузеров, о которых разработчик банально может не знать, и большая компактность получающегося кода.

Но с другой стороны, для старых браузеров можно использовать условно подключаемые полифилы, и результат будет надёжнее и предсказуемее, так как фреймворк неизвестно насколько большой совместимостью в сторону старых версий обладает, особенно если брать динозавров вроде IE 6-8 и новые версии JQuery. Причём даже если фреймворк обладает такой совместимостью, то вот плагины к нему (на примере того же JQuery) с вероятностью 99.9% — нет, и всё равно в старом браузере сайт сломается и будет нерабочим. Кроме того, в случае 90% проектов на совсем старые браузеры вроде IE до версии 9 можно вообще забить, поставив заглушку с просьбой обновиться.

Что до компактности, то из-за неё сложность исполнения кода браузером увеличивается если не на порядок, то в разы, также растёт и количество потребляемой памяти: например, известный факт что движки обычно работают с синтетическими объектами событий (заворачивая внутрь них обычные как одно из полей), оборачивают HTML элементы в специальные объекты, возвращая результаты из функций-селекторов, и так далее).

sumanai 22 сентября 2018 в 11:00

+8

> Надо просто писать без багов.

> Так надо писать так, чтобы работало во всех, а не в пачке!

Как у вас всё просто. А чтобы быть здоровым и богатым, нужно зарабатывать много и не болеть?

VolCh 22 сентября 2018 в 16:31

+1

Насчёт совместимости: браузеры постоянно обновляются, языки тоже. Написав свою библиотеку один раз вы всё время должны будете её поддерживать.

ScreamPassion 20 сентября 2018 в 18:28

+6

Иногда глупо не добавлять jQuery для подобных вещей, только потому, что jQuery порой более стабилен большинства известных браузеров.

SeraflmArts 21 сентября 2018 в 00:32

+12

Вот из-за подобных мыслей сайты и тормозят. Опытные разработчики возьмут и просто напишут нужный transition на css и добавят will-change для фонового пререндера и выноса этой анимации в отдельный слой, чтобы не перерисовывавать всё, когда что-то будет меняться. Это всё будет работать на GPU и не тратить процессорного времени и работать будет моментально, и не будет жрать лишнего, т.к. браузер знает что будет анимироваться, а что нет.

А другой человек подумает, возьму я JQuery (который, к слову, давно уже пора закопать) и буду анимировать всё на нём. И конечно же будем вычислять каждый кадр каждого свойства каждой анимации на CPU, и каждый раз будем сбрасывать кеш всей страницы. Это ведь на порядок, как вы говорите, стабильнее. А на пользователей нам плевать.

Верно я понял вашу мысль на тему «JQuery для анимаций»?



PsyHaSTe 21 сентября 2018 в 00:53

+6

Да я до сих пор помню свой проект, где я сделал на css необольшой аккордеон (я бекенд разработчик, которого на фронт сунули, не судите строго :)). Потратил около недели, чтобы все красиво было, принес на ревью, мне сказали «шо эта, сложный цсс, я не понимаю. Мы пишем не write-only код, а цсс который занимается вычислениями это юнсенс! (там были calc-и для заголовков элементов в аккордеоне). Жду от тебя через 2 дня все переписанным на понятный js…

Долго я там не проработал, но осадок остался…



SerafimArts 21 сентября 2018 в 06:07

+1

Ну css тоже надо уметь писать, иногда он действительно нечитаемым становится от избытка стилей. Для таких случаев и были придуманы препроцессоры с миксинами, функциями, инклудами и прочими штуками, сохраняющими DRY. С другой стороны может опыта у ревьювера было маловато, вот его и понесло чуть-чуть.

В любом случае визуализация должна быть на стороне css — это точно, а на JS уже бизнес-логика. Если ваш calc только для позиционирования элементов, то не вижу ничего страшного в этом, сам с недавнего времени начал активно его использовать и не понимаю теперь как жил без него.



verbaux 21 сентября 2018 в 11:44

−1

"… препроцессоры с миксинами, функциями, инклудами и прочими штуками" превращают понятный CSS в нечитаемай без дополнительных телодвижений код.

Barafu_Albino_Cheetah 22 сентября 2018 в 00:17

0

Ну да. Их назначение — брать код, нечитаемый даже с телодвижениями и сделать его кодом, читаемым с дополнительными телодвижениями.



ScreamPassion 21 сентября 2018 в 10:52

+4

Не забывайте jQuery — это в первую очередь более-менее удобные селекторы, ну а во вторых, если следовать вашей логике, то сперва надо написать свою ОС, потом свой браузер, а уже потом если все таки жизни хватит, писать нормальную анимацию)



SerafimArts 21 сентября 2018 в 15:05

0

А что по этому поводу говорит автор в статье? Тоже предлагает ОС свою написать?

Я привёл в пример всего лишь один из случаев, когда кто-то, не зная одного инструмента, но зная другой ненамеренно всё усложняет. По-этому весь софт и становится медленнее.

Конечно, есть и другие критерии. Например, дополнительные абстракции ради снижения сложности поддержки ПО. Как раз тот из вариантов, когда JQ можно использовать для селекторов. Хотя, лично я в этом вижу не слишком много смысла, благо 2018год на дворе и давно есть querySelector, но мнение имеет право на существование.



ScreamPassion 21 сентября 2018 в 15:12

0

Я бекенд программист, по этому лично для меня, написание нативных решений js, в любом случае подразумевает дополнительные затраты по времени. Автор предлагает улучшать все, где мы имеем счастье(и/или несчастье) работать, так почему бы (следуя этой логике) не начать с главного (ос), а потом уже про jquery поговорим отдельно…



SerafimArts 21 сентября 2018 в 15:33

0

Тут есть такая тонкая грань, где использование JQ обосновано, а где нет. Часто пишут велосипеды, вместо того, что бы использовать готовое, часто наоборот. Решать где нужно что-то писать с нуля, а где нет — опыт и зачастую чисто субъективное.



third112 25 сентября 2018 в 14:45

0

> Тут есть такая тонкая грань

Даже 2 грани. Можно взять готовый код, описание алгоритма и сильно переработать под свою задачу (это будет не с нуля), но иногда вставляют как есть.



popov654 21 сентября 2018 в 17:19

0

> по этому лично для меня, написание нативных решений js, в любом случае подразумевает дополнительные затраты по времени

Вы полагаете, изучить и научиться применять на практике такую большую библиотеку, как jQuery (пусть и не самую сложную из тех что есть), будет быстрее? Тут смотреть надо



0xd34df00d 21 сентября 2018 в 20:14

+1

Автора в статье поведение OOM-киллера в линуксах удивило, так что я уже боюсь представить, что он там про ОС думает.



popov654 21 сентября 2018 в 17:17

0

Но ведь удобные селекторы теперь есть из коробки с приходом querySelector и querySelectorAll в HTML5… Или jQuery умеет что-то ещё в этом плане?

MrDaedra 21 сентября 2018 в 20:24

0

Тот же GitHub заменил jQuery на querySelectorAll и fetch с полифиллом.

michael_vostrikov 21 сентября 2018 в 10:08

+1

Ее обычно добавляют не для анимации, а для удобства работы с DOM. Действия с информацией предполагают работу с элементами интерфейса. Кто-то jQuery использует, кто-то Angular, совсем без js сложно обойтись.



Wesha 21 сентября 2018 в 00:07

−2

> Предпросмотр нового коммента все-таки удобнее, чем перезагружать страницу отправкой формы и проверять как оно выглядит, не так ли?)

То есть в эпоху стамегабитного интернета Вам западло лишний раз "перезагрузить страницу отправкой формы"? Если уж Вам тяжело перещагружать страницу со 100500 комментариями — загрузите окно предварительного просмотра в iframe, наконец!

sumanai 21 сентября 2018 в 09:15

+1

> iframe

Который по сути отдельная страница, с кучей накладных расходов на её создание и поддержку.

Ranwise 21 сентября 2018 в 09:29

+1

на мобилках уже есть 100 Мбит?

sumanai 21 сентября 2018 в 09:37

0

По WiFi у себя дома.

saege5b 21 сентября 2018 в 11:02

+1

С 10-50 соседями на каждом канале.
Или, едва добивающемуся 5 ГГц в соседнюю комнату.

sumanai 21 сентября 2018 в 11:06

0

Я не говорил в квартире. Частный дом с участком в 6 соток позволяет вольготно использовать 2,4 ГГц.



Asaelko 21 сентября 2018 в 17:20

0

В Москве и Петербурге до 139Мб/с выжимал на 4G+ с Йотой. В среднем 50-70 где-то в центре города.

michael_vostrikov 21 сентября 2018 в 10:00

0

Ну у меня 400 КБ/с. Попробовал открыть страницу с этой статьей в приватном режиме, в первый раз 16 секунд, во второй 10. В этих условиях мне удобнее предпросмотр на js.

Если на каждой странице будет по iframe со своей разметкой на каждое действие, то мне кажется ресурсов это будет занимать еще больше. Вместо 20 вкладок в памяти браузера будет 60-100. А если и нет, все равно это менее удобно, почему и стали делать на js.

novice2001 21 сентября 2018 в 09:08

0

А может не надо пихать на каждую страницу комменты и т.п.? Не?
Тот же Хабр в данном случае исключение, здесь комменты могут быть даже интереснее статей. Но в подавляющем большинстве случаев комменты прямо на странице нафиг не нужны, а вот малый вес и быстрая работа еще никому не вредили. Хотите пообсуждать — вот вам ссылка на форум. Легко и просто, но не модно, это да.

michael_vostrikov 21 сентября 2018 в 10:22

+1

А причем здесь именно комменты? Создание информации не только из комментов состоит. Валидация форм, подгрузка элементов списка по фильтру при вводе, скрытие/показ элементов по условию. А еще есть бэкенд, который все это обрабатывает. И это все надо написать и потом поддерживать.

Критикуешь — предлагай. Если это легко и просто, покажите, что сделали лично вы и сколько времени на это потратили, без js и библиотек/фреймворков. Причина идет из создания кода, а не из моды. Я этим занимаюсь, и утверждаю, что не легко и не просто.

novice2001 21 сентября 2018 в 11:43

+1

Я говорю о конкретной и при этом самой распространенной категории сайтов. Таких сайтов, которые были практически единственными на заре веба. Сайтов информационных, на которых контент генерирует только автор сайта. Где пользователь только просматривает его. Поэтому там не нужно ни форм с валидацией, ни прочего. Но где все это навешивается, потому что у 100500 других сайтов такое есть.
В конце концов есть простое и абсолютно элементарное решение — кнопка или ссылка, по которой пользователь переходит на страницу, где вы можете хоть гигабайт жабаскрипта навешать. Хочет какой-то пользователь пострадать — пусть страдает. Не надо делать тормозные страницы, только из-за того, что кто-то захочет что-то прокомментировать.
И даже на более интерактивных сайтах можно что-то такое придумать, чтобы страницы, где 99,99% пользователей только читают, не загружали весь мусор, необходимый для интерактивного взаимодействия.



Sychuan 21 сентября 2018 в 11:54

0

> . Где пользователь только просматривает его. Поэтому там не нужно ни форм с валидацией, ни прочего. Но где все это навешивается, потому что у 100500 других сайтов такое есть.

Такие сайты никуда не делись. Вот пожалуйста, www.kurims.kyoto-u.ac.jp/~motizuki/top-english.html
Или возьмите arxiv там все так как и было. Но фейсбук и хабр не могут быть как в 90-ых.

novice2001 21 сентября 2018 в 12:03

+1

Перестаньте уже сравнивать все подряд с фейсбуком и хабром (а также ютубом, гмейлом, твиттером и т.п.).
Сайты на которых интерактивное взаимодействие — основная, критическая часть функционала — это одно. Где совершенно необязательная — совсем другое. Но вторые почему-то маниакально изображают из себя гибрид фейсбука с твиттером.

michael_vostrikov 21 сентября 2018 в 12:33

0

> Я говорю о конкретной и при этом самой распространенной категории сайтов. Сайтов информационных, на которых контент генерирует только автор сайта.

Таких сайтов сейчас практически нет. Интернет-магазины, соц-сети, веб-интерфейсы к сервисам типа почты — это самые распространенные категории, и везде есть функционал, который требует действий от пользователя. Даже на сайте газеты NY Times есть кнопки подписки, поиска, и подгрузка картинок при прокрутке. И как раз на популярных ресурсах перенос части обработки на клиент дает пользу.

> по которой пользователь переходит на страницу, где вы можете хоть гигабайт жабаскрипта навешать

Ну вот это они и есть. Не хотите, не переходите) Вы предлагаете тратить в 2 раза больше денег и поддерживать 2 версии. Мало кто на это пойдет.

Мне тоже не нравится тормознутость, но надо понимать, что у нее есть причины, и нельзя просто взять и убрать ее, ничего не потеряв.

novice2001 21 сентября 2018 в 13:05

+6

Таких сайтов сейчас есть сколько угодно.
Модная подгрузка страницы при прокрутке — на самом деле одно из вреднейших изверений. Если раньше при использовании pagination я мог сразу перейти на 10-ю, 100-ю, да хоть на миллионную страницу, то теперь чтобы попасть на 100-ю мне нужно перелистать все 99 предыдущих. Мало того раньше первая и последняя страницы работали одинаково быстро, то теперь быстро (и то относительно) работает только первая, поскольку при подгрузке постепенно растет объем пожираемой памяти и тормоза.

> Вы предлагаете тратить в 2 раза больше денег и поддерживать 2 версии

А зачем ее отдельно поддерживать? Если вы не будете лезть с ненужными никому кроме вас «улучшениями», то она может работать десятилетиями. В конце концов верстку вы можете иметь идентичной с интерактивной страницей.

> Мне тоже не нравится тормознутость, но надо понимать, что у нее есть причины

У нее гораздо меньше рациональных причин, чем вам хочется думать.

michael_vostrikov 21 сентября 2018 в 14:27

0

> Таких сайтов сейчас есть сколько угодно.

Примеры?

> Модная подгрузка страницы при прокрутке

Я не писал "подгрузка страницы", я написал "подгрузка картинок". Это совсем не то, о чем вы говорите.

> А зачем ее отдельно поддерживать?

У вас какое-то неправильное представление о том, как делаются веб-приложения. Если мы переименовали поле в базе данных с body на text, то и в обычном коде, и в коде "без скриптов" надо поменять условный output($news['body']) на output($news['text']). Если мы поменяли верстку шаблона интерактивной страницы для новости, то и верстку шаблона для страницы "без скриптов" надо менять, чтобы они были идентичные. Просто отключить скрипты не выйдет, вы можете в этом убедиться, отключив скрипты в браузере. Нужно предпринимать отдельные усилия, чтобы работало и так и так. Один и тот же шаблон использовать нельзя, потому что в одном шаблоне есть условная кнопка "Подписаться", а в другом нет. Либо каждый блок, требующий скриптов, оборачивать в проверку выводить или нет, и проверять, что при добавлении очередной кнопки в версии без кнопок ничего не поехало. И т.д. и т.п.

> У нее гораздо меньше рациональных причин, чем вам хочется думать.

Так я причины из практики знаю. То, о чем вы говорите, требует дополнительных ресурсов при разработке.

novice2001 21 сентября 2018 в 14:42

0

> Примеры?

Вики. Которая как раз прекрасно работает на отображение вовсе без js, а редактируется на отдельной странице.

> Просто отключить скрипты не выйдет, вы можете в этом убедиться, отключив скрипты в браузере

Это только при условии, что js служит не только для взаимодействия, но и для отрисовки. И да, вы меня тоже изначально неправильно поняли. Совсем необязательно иметь 2 визуально идентичных версии страницы, на одной из которых комментировать нельзя, а на другой можно. Есть смысл разделить функционал. Очевидно, что подавляющее большинство только читает, скажем, новости. Они получают легкую и быструю страницу без лишних скриптов с

кнопкой/ссылкой «обсудить». И вот на этой странице делайте что хотите. Я не вижу смысла делать ее визуально похожей на первую страницу.

> Так я причины из практики знаю. То, о чем вы говорите, требует дополнительных ресурсов при разработке.

Так это потому что вы и те, кто дает вам задание, мыслите именно в такой парадигме.
Вообще было бы гораздо проще обсуждать все на конкретных примерах.

michael_vostrikov 21 сентября 2018 в 16:27

–1

> Вики.
>
> Мы говорили о сайтах, которые "Поэтому там не нужно ни форм с валидацией, ни прочего. Но где все это навешивается, потому что у 100500 других сайтов такое есть.".
> В вики информация статичная, действия пользователей не требуются, потому там и скриптов нет. А там где есть, там навешивается не потому что "у других сайтов есть", а потому что надо для определенных целей. Даже анимацию добавляют не потому что на других сайтах есть, а потому что там из-за этого новых пользователей больше.
> Так какие есть распространенные сайты, где все это есть, но можно убрать без проблем?

> Они получают легкую и быструю страницу без лишних скриптов с кнопкой/ссылкой «обсудить». Я не вижу смысла делать ее визуально похожей на первую страницу.

Ну вот у нас и появилось 2 страницы вместо одной, да еще и с разной версткой. И почему вы все время про комменты говорите? Это был пример конкретно про Хабр, на других сайтах свои применения скриптов. Та же кнопка "Подписаться". Сделают так, как вы предлагаете, а потом будут жалобы от пользователей "почему у вас кнопку Подписаться не найти, вот сайте X она рядом со статьей, нажал и все".

> Так это потому что вы и те, кто дает вам задание, мыслите именно в такой парадигме.

Если на фреймворке прототип делается за час, а без него за неделю, тут не надо никаких парадигм.

novice2001 21 сентября 2018 в 16:53

+1

Я так понимаю, что для вас «распространенные сайты» — это сайты, которые вы разрабатываете.
Потому что если судить с позиции количества пользователей, то вики — это куда более «распространенный» сайт, чем все сайты, в разработке которых вы хоть как-то принимали участие за всю жизнь. Если судить с точки зрения количества самих сайтов, то как раз сайтов со статической информацией вася-пупкин.рф опять же на порядки больше ваших сайтов, в которых действительно нужны тонны скриптов.

> Даже анимацию добавляют не потому что на других сайтах есть, а потому что там из-за этого новых пользователей больше.

Это как? Откуда появляются при этом новые пользователи?

> Так какие есть распространенные сайты, где все это есть, но можно убрать без проблем?

Я свои примеры привел. Давайте наоборот — приведите хотя бы один пример сайта где тонны js прямо необходимы.

> Та же кнопка «Подписаться». Сделают так, как вы предлагаете, а потом будут жалобы от пользователей «почему у вас кнопку Подписаться не найти, вот сайте X она рядом со статьей, нажал и все».

По-видимому вы так и не поняли. Оставьте все необходимые кнопки, только сделайте их просто ссылками на соответствующие страницы — и все! Даже не на много, а на одну страницу.

> Ну вот у нас и появилось 2 страницы вместо одной, да еще и с разной версткой.

И не говорите, это просто ужас. Ведь на второй странице нужно обязательно делать какой-нибудь крутой дизайн, а это сложно, дорого и бла-бла-бла.
Правда я сделал бы там дизайн в духе ya.ru (ага!), который стоит 3 копейки и не нуждается в поддержке. Но да, это же сейчас не модно…

> Если на фреймворке прототип делается за час

Дак кто ж вам запрещает делать на фреймворке **прототип**? Сколько угодно. Но вот в продакшн — не надо…



PsyHaSTe 21 сентября 2018 в 18:00

0

> И не говорите, это просто ужас. Ведь на второй странице нужно обязательно делать какой-нибудь крутой дизайн, а это сложно, дорого и бла-бла-бла.
> Правда я сделал бы там дизайн в духе ya.ru (ага!), который стоит 3 копейки и не нуждается в поддержке. Но да, это же сейчас не модно…

Пользователям тоже будете сами объяснять, какой это крутой дизайн?

novice2001 22 сентября 2018 в 09:21

+1

Ах вон оно как, т.е. это пользователи требуют от вас какого-то особого дизайна на сугубо функциональных страницах, а не вы сами хотите выпендриться с мегатоннами украшений и жабаскрипта…
Это очень странно, поскольку google.com миллиардам пользователей по-видимому вполне нравится. А там как раз дизайн вроде того, что предлагаю я.



PsyHaSTe 23 сентября 2018 в 04:13

0

> Конечно пользователи. Я-то реализовываю то, что сказал продукт, а продукт анализирует как раз требования пользователей и «он те чуваки прикрутили X, и у них конверсия увеличилась, айда так же сделаем».

michael_vostrikov 21 сентября 2018 в 20:51

0

> Я так понимаю, что для вас «распространенные сайты» — это сайты, которые вы разрабатываете.

Из чего вы сделали такой вывод? Вы говорили про "распространенные сайты, где все это навешивается". Я попросил примеры. В Википедии это навешивается? Нет. Так какое отношение она имеет к разговору?

Я бы даже сказал, она подтверждает мои слова. Там не нужны скрипты, вот и не добавляют. А где добавляют, значит там они зачем-то нужны.

> как раз сайтов со статической информацией вася-пупкин.рф опять же на порядки больше ваших сайтов

А какая вам разница, что на личном статическом сайте Васи Пупкина используется JS? Там все равно ничего не тормозит, потому что функциональности мало, да и вряд ли вы столько личных сайтов посещаете.

> Это как? Откуда появляются при этом новые пользователи?

Ну вот так. Нужен человеку некий функционал, есть 2 сайта с таким функционалом, на одном есть анимации, приятные глазу, на другом нет. Поэтому он зарегистрировался на первом сайте.
На втором сайте заметили, что пользователей что-то мало, а в интернете все первый сайт обсуждают, запустили рекламу с а/б тестированием, и оказалось, что если интерфейс без анимаций, то меньше пользователей регистрируется.
А третий сайт не стал проводить тестирование, а сразу добавил анимации.

> Я свои примеры привел. Давайте наоборот — приведите хотя бы один пример сайта где тонны js прямо необходимы.

Нет, не привели. Только про одну Википедию написали, которая ни при чем.
И давайте без демагогий. Я не говорил, что необходимы именно тонны js, я говорил, что в целом js и всякие библиотеки/фреймворки нужны. Но вообще https://docs.google.com неплохой пример.

> Оставьте все необходимые кнопки, только сделайте их просто ссылками на соответствующие страницы — и все! Даже не на много, а на одну страницу.

"Я нажимаю кнопку Подписаться, а страница просто перезагружается. С остальными кнопками то же самое. Почините немедленно!". Ну и да, одна страница все равно требует меньше ресурсов (для разработки), чем две похожие. И зачастую разница гораздо больше, чем в 2 раза.

> Правда я сделал бы там дизайн в духе ya.ru (ага!), который стоит 3 копейки и не нуждается в поддержке.

Вы неправильно представляете, как делаются веб-приложения. Не требует поддержки только страница, которая сверстана полностью в блокноте и лежит на диске в виде отдельного файла. В любом блоге на Вордпрессе, в любом новостном сайте, информация сохраняется в базу данных и выводится кодом. Все это требует поддержки и проверки при изменениях.
Ну и да, большинству пользователей неохота смотреть на ужасный дешевый дизайн.

> Но вот в продакшн — не надо

Вы думаете, для продакшена математика другая и там по волшебству дополнительные недели появляются?)

popov654 22 сентября 2018 в 09:59

0

> Ну и да, одна страница все равно требует меньше ресурсов (для разработки), чем две похожие. И зачастую разница гораздо больше, чем в 2 раза.

Про сложность согласен, это важный фактор. Хотелось бы однако отметить, что есть техники, позволяющие обойтись всего одной

страницей, и при этом обеспечить её полиморфное поведение. То есть она будет одинаково хорошо работать и при отключенном, и при включенном JS. Вы наверное об этом знаете. Ну да, в разработке это сложнее, и даже не в 2 раза. Зато у нас одна страница, работающая везде и у всех (даже в каких-нибудь мобильных Операх Мини), все правки централизованы в одном месте, разработчики заняты делом и получают повышенную зарплату за возросшую сложность. Классно же.

dimm_ddr 24 сентября 2018 в 14:24

0

> Классно же.

Офигенно! Для всех кроме владельца сайта который за это платит.



XXXXPro 26 сентября 2018 в 18:58

+1

> Я бы даже сказал, она подтверждает мои слова. Там не нужны скрипты, вот и не добавляют. А где добавляют, значит там они зачем-то нужны.

Далеко не всегда. Скажем, надоедливые баннеры «добавляйтесь в друзья» или «закажите обратный звонок», закрывающие контент в самый неподходящий момент, вряд ли можно назвать нужными. Но их вешают. Про навязывание подписки на push-уведомления и вовсе молчу. Или, скажем, онлайн-чаты, которые сейчас на каждом втором магазине висят. В принципе, неплохая вещь, если их делать нормально, то есть сначала видна только кнопка «Задать вопрос» (еще для привлечения внимания по ней можно какой-нибудь блик пустить), а при щелчке по ней подгружаются необходимые скрипты и начинается чат. Но нет же, делают загрузку скриптов сразу и сразу же разворачивают окно чата.

crystallize 24 сентября 2018 в 14:22

0

> Если на фреймворке прототип делается за час, а без него за неделю, тут не надо никаких парадигм.

Так речь же о том что на фреймворке у конечного пользователя тормоза начнутся.

michael_vostrikov 24 сентября 2018 в 14:42

+1

Речь была о том, что убрать фреймворки легко и просто, потому что они низачем не нужны, кроме моды. Я объяснил, что это не так. Есть выбор — уменьшить стоимость разработки (время и деньги), добавив тормозов конечному пользователю, или наоборот. Обычно выбирают первое.

sumanai 21 сентября 2018 в 15:20

0

> Либо каждый блок, требующий скриптов, оборачивать в проверку выводить или нет, и проверять, что при добавлении очередной кнопки в версии без кнопок ничего не поехало. И т.д. и т.п.

```
.come_block {
    display:none;
}
body.has_js .come_block {
    display:block;
}
```

michael_vostrikov 21 сентября 2018 в 16:34

0

Ну вот этот come_block надо добавлять в каждый блок, и желательно в обертку, чтобы собственным классам не мешало. Суть же не в том, что сделать нельзя, а в том, что поддерживать надо. А еще оно все равно на клиент поедет.



popov654 21 сентября 2018 в 17:29

0

В обёртку — это как? come_block — Это и есть собственный класс в примере выше)

michael_vostrikov 21 сентября 2018 в 21:00

0

```
<button class="display_inline_block come_block"></button>
```

Ну если вы вот так сделаете, то верстка может поехать, там же display другой задан.



popov654 22 сентября 2018 в 10:01

0

А, не понял вас. Да, поехать теоретически может. Добавление/удаление каждой кнопки по-хорошему надо будет тестировать на обеих версиях. Я просто имел в виду, что класс самой кнопки останется без изменений, и правила для неё тоже.

sumanai 22 сентября 2018 в 11:04

+2

Мой код там только в качестве примера, притом не самого лучшего. Можно сделать наоборот

```
body.no_js .come_block {
    display:none;
}
```

И удалять класс no_js яваскриптом.

VolCh 22 сентября 2018 в 14:12

0

На сайтах может и не нужны, но сейчас веб это не только, а может и не столько сайты, сколько распределённые приложения, а браузеры — среда исполнения толстых клиентов.

Kodath 20 сентября 2018 в 15:49

+26

Но автор же предлагает не переписать абсолютно все с нуля или начинать писать не используя чужие библиотеки и фреймворки, а хотя бы не городить по верх и так уже раздутого кода еще и свою халабуду из такой же «эпоксидной смолы» и палок. А на счет расширения штата разработчиков — тут вообще не понятная ситуация, когда команды/компании набирают сначала овер дохера менеджеров, чтоб продать продукт на который код еще и писать то не начали, а потом еще и выясняется, что на разработчиков ничего уже не осталось из заложенного бюджета.

springimport 20 сентября 2018 в 16:20

+1

Недавно как раз одна компания отчиталась о превышении некоего лимита впервые в истории. Думаете не хватает на разработчиков?



Massacre 20 сентября 2018 в 16:31

–1

А причём тут ядро, система может быть одна, просто вместо интерпретируемых языков для прикладных приложений будут снова использовать компилируемые. И делать сайты без тонны JS и с серверным рендером тоже можно. Всего-то должна пройти мода на одностраничники.

lubezniy 21 сентября 2018 в 09:48

+1

У серверного рендера есть свои недостатки на перспективу. Если проект имеет в будущем шансы на интеграцию со сторонними приложениями, нужно будет проектировать и писать API. На старте проекта можно сделать единый механизм для веб-фронтенда и приложений, а рендерить на клиенте, и таким образом не делать двойную работу.

solver 21 сентября 2018 в 11:43

+1

Только вот большинство сайтиков никогда ни с кем не интегрируются…
Но страдать пользователей от тормозов теоретической будущей интеграции заставляют уже сейчас.

lubezniy 21 сентября 2018 в 15:31

0

Ну так совсем не обязательно реализовывать на клиентской стороне функции, не нужные

браузерному приложению.

DrPass 20 сентября 2018 в 17:17

+19

Так дело же не в том, что для получения хорошего софта надо платить в сто раз больше. Надо платить, условно говоря, на 20% больше, чтобы решить 80% проблем. Этого достаточно.
Мы пользовались софтом, который работал на компьютерах в сто раз медленнее нынешних смартфонов, при этом он был значительно более функциональный, отнюдь не стоил космических денег для пользователя, а разработавшие его программисты были ничуть не менее богатыми, чем современные. Так что проблема отнюдь не в том, что для разработки более качественного ПО нужно намного больше денег.

yefrem 20 сентября 2018 в 17:42

−4

> Мы пользовались софтом, который [...] был значительно более функциональный

это каким же?

DrPass 20 сентября 2018 в 18:06

+19

Да что угодно возьмите. Сравните функционал почтового клиента конца 1990-х годов и того, что в вашем телефоне. Сравните ваш просмотрщик электронных таблиц и Excel, который весил меньше, чем калькулятор у вас на телефоне.

lorc 20 сентября 2018 в 18:19

+7

Кстати да, Слак никогда не будет уметь того, что умеет kvirc, например.

Мобильный софт — это вообще издевательство. Там количество фич только уменьшается. Старые гугл.карты (еще на Андроиде 2) умели трекать остановки транспорта. В новых появилась прекрасная фича «потряси телефон, если нашел ошибку», зато они перестали показывать таймлайн с остановками маршрутки. Хотя, вроде в этом году вернули эту фичу обратно…

Атом (который редактор) в принципе не умеет того, что умеет мой emacs.

Про новый скайп уже говорили.

Даже в новом MSPaint потеряли какую-то фичу, которой многие пользовались в оригинальном приложении. Извините, не помню деталей.

При желании, этот список можно продолжать еще долго.

yefrem 20 сентября 2018 в 19:01

+2

Сравнение с мобильными некорректно, так как у них совсем другая ниша, потому и софт такой, каков он есть.

> При желании, этот список можно продолжать еще долго.

В том-то и дело, что при соответствующем желании можно вывести любой тренд, какой душе угодно.

В приведенных примерах, имхо, сравниваются не совсем аналоги. Но даже в них, скорее всего (т.к. не в курсе про Атом), можно найти обратные примеры, навроде «kvirc не умеет в звонки и скрин-шеринг». Только вы ответите «это не нужно» и будете правы. И аналогично можно найти кучу фич современного софта, о которых раньше вообще не шло и речи.

VolCh 22 сентября 2018 в 18:04

0

Так речь в том числе и о мобильных приложениях, некоторым из которых с десяток лет, но развиваются они странно с точки зрения пользователя и программиста — фичи выпиливаются, а добавляются рюшечки.

AlexPancho 22 сентября 2018 в 18:26

0

Вот пример одинакового: функциональноЕксель97 и Ескель2016 если и отличаются, то как раз не существенно для 80% пользователей. Но скорость и качество работы… Про размер дистрибутива я вообще скромно умолчу.

lokkiuni 24 сентября 2018 в 17:35

0

Есть нюанс — скорость работы на современном железе как раз в 16 выше. Да, интерфейс тупит, но когда надо 100500 формул считать (а эксель это в первую очередь мегакалькулятор, а не блокнот с ячейками) — 2016 будет сильно впереди. Ситуаця ровно как с фотошопом — в операциях, важных для профессионалов, новые версии сильно быстрее. Просто не всем это нужно, и в отличие от экселя, быстрые маленькие замены для фотошопа есть.

Xaliuss 25 сентября 2018 в 19:12

0

Функционально он мало отличается при использовании в качестве калькулятора. А так в каждой новой версии появляется несколько полезных конкретно мне фич. Тоже автосохранение, интеграция с облаком и совместная работа появившиеся в версиях 2016+ резко повысили комфорт работы, особенно с нескольких компьютеров и более чем одним пользователем.

Скорость работы именно программы выросла — Эксель научился нормально использовать многопоточность для расчетов (сравнивал 2016 на десятке с 2013 на семерке). Либреофис отстаёт очень сильно, и может служить нормальной заменой только для относительно простых задач.

khim 25 сентября 2018 в 20:49

0

> Скорость работы именно программы выросла — Эксель научился нормально использовать многопоточность для расчетов (сравнивал 2016 на десятке с 2013 на семерке).

Вот это, пожалуй, первое **реально** полезное добавление после MS Office 2000. Хоть кто-то что-то может назвать.

> совместная работа появившиеся в версиях 2016+

Тоже интересно.

Но, заметьте, обе этих вещи отсутствовали в MS Office 2000 в силу полнейшей бессмысленности в тогдашних реалиях, а не потому что разработчики не могли их реализовать… и совершенно неясно почему эти вещи нельзя было добавить сохранив быстрый и отзывчивый интерфейс…

> Либреофис отстаёт очень сильно, и может служить нормальной заменой только для относительно простых задач.

Возможно. Я не нагружаю его так сильно, чтобы меня нервировало отсутствие многопоточности — а вот то, что MS Office изображает на экране «паток» даже на 24-ядерной машинке… меня раздражает…

VolCh 26 сентября 2018 в 10:20

0

> в силу полнейшей бессмысленности в тогдашних реалиях

Ну в начале 2000-х как минимум одноранговые локальные сети были нормой, а многие даже небольшие компании и домен на w2k поднимали. И терминальные серверы тогда же или чуть позже (в w2k3) в массы пошли.

khim 26 сентября 2018 в 22:10

0

> Ну в начале 2000-х как минимум одноранговые локальные сети были нормой

В одноранговой сети совместное редактирование документов не нужно. Если уж так хочется вот прям одновременно вдвоём редактировать — проще взять стульчик и сесть вдвоём у одного компьютера. Правда-правда, попробуйте.

А вот что **нужно** — так это передача документов от одного отдела другому с разными рецензиями и прочим. И это всё в MS Office 2000 (да и в MS Office 95/97) достаточно хорошо развито.

Simplevolk 21 сентября 2018 в 09:32

+7

В новом скайпе я физически ощущал его сопротивление моей попытке напечатать сообщение…

ashalaenko 21 сентября 2018 в 17:13

0

Я в этом плане грешу на интернет и распространение софта по сети. Раньше пользователь покупал физическую копию и если там обнаруживался баг, то быстро доставить фикс было проблематично, что стимулировало больше времени тратить на тестирование, тестирование на различных архитектурах, стабилизацию и тд. Ведь бажный софт мог повлиять на репутацию компании. Это же касается игр. Сейчас, когда патч можно доставить за считанные минуты практически всем клиентам, наступила эра сырого софта (и игр в том числе). Принцип «лучшие qa — это наши пользователи» вышел на новый уровень. Репутационные проблемы решают пиарщики (вот вам фтболка и скидка на следующую версию). И вправду, зачем тратить время на глубокое тестирование и отшлифовку, если через 10 минут после релиза пользователи все сами напишут, и если, вдруг, у кого-то будут проблемы — мы сможем сразу им выкатить фикс. Это же позволяет выкатывать недоделанные фичи, доделаем следующим минорным релизом, а пока и mvp сойдет.

Druu 22 сентября 2018 в 05:44

+1

> Ведь бажный софт мог повлиять на репутацию компании. Это же касается игр.

Будто раньше в предел забагованных игор не выходило.

Druu 22 сентября 2018 в 05:42

−1

> что умеет мой emacs.

Только вы почему-то не упомянули что емакс *тормозит* ;)



DarthVictor 20 сентября 2018 в 18:46

−1

Современные смартфоны существенно мощнее суперкомпьютера НАСА конца 60-х годов. Но тогда производительности компьютеров хватало, чтобы запускать человека на луну, а сейчас только чтобы запускать птиц в свиней.

yefrem 20 сентября 2018 в 19:07

+3

вы платите оверхедом не за возможность запускать птиц, а, грубо говоря, за возможность играть в новую игру хоть каждый день, при этом устанавливая ее и удаляя старую в течение минуты, причем бесплатно (относительно)

yarric 20 сентября 2018 в 22:57

+2

> за возможность играть в новую игру хоть каждый день

А можно поподробнее, что в этом инновационного по сравнению с, к примеру, приставками из 90-х?



shpaker 21 сентября 2018 в 09:57

+4

Графен, не более. Механики мобильных игр крайне тупы. Плюс с тех пор как там продоминировал фри2плей так и вообще все маркеты сгнили.

VolCh 22 сентября 2018 в 18:07

0

идти и покупать/арендовать/менять картриджи надо было, переставлять их, фактически меняя аппаратную конфигурацию компьютера.

yarric 22 сентября 2018 в 18:14

0

На PlayStation уже шли игры на CD.

khim 22 сентября 2018 в 19:43

0

И тем не менее, это можно было желать «хоть каждый день». Картриджи арендовались за небольшую сумму и киосков с ними было… достаточно.

VolCh 23 сентября 2018 в 16:43

0

По сути каждый день происходил апгрейд аппаратной части.

DrPass 20 сентября 2018 в 23:53

+2

Возможность играть в новую игру каждый день мы получили благодаря современным средствам доставки софта от производителя до клиента, когда любой может сделать игру и продать её вам без участия издателя. Средства быстрой разработки не слишком продвинулись с того времени, когда в конце 1990-х появились независимые игровые движки. А в сфере прикладного софта и того хуже, переход приложений в веб отбросил инструменты разработки (и продуктивность программистов) лет на десять-пятнадцать назад, потеряв почти все преимущества, которые приобрели десктопные IDE.



SHadDim 21 сентября 2018 в 15:18

0

А AppStore и GoogleStore это не «издательство», это «средство доставки софта, которые имею приличную долю в финальной стоимости софта» <_<

DrPass 21 сентября 2018 в 15:31

+1

В том-то и дело, что это всего лишь удобные магазины. Они не имеют вообще никакой доли в финальной стоимости софта и не претендуют на неё. Отличия от издательств тут колоссальные.
1. Вам не нужно убеждать магазин в том, чтобы он начал распространять ваш продукт. Зарегистрировались, выложили. А дальше уже покупатель голосует рублём/долларом/юанем.
2. Магазин не имеет никаких прав на ваш продукт, ни даже эксклюзивного права на распространение. Как вам больше нравится — так и распространяйте. Издательство же вас сожрёт с потрохами.
3. Магазин у вас не требует никакой финансовой доли в бизнесе. Он продал ваш софт — он взял свою комиссию, не продал, он ничего не потребовал. Насчет того, что комиссия большая, вопрос спорный. Сколько вам не жалко отдать денег посреднику за продажу вашего товара, которой без него вообще бы не было? Особенно если учесть, что тиражирование этого товара не стоит вообще ничего.



0xd34df00d 21 сентября 2018 в 20:24

0

> Они не имеют вообще никакой доли в финальной стоимости софта и не претендуют на неё.

И они даже не берут некоторый процент с продаж?

> Магазин не имеет никаких прав на ваш продукт, ни даже эксклюзивного права на распространение. Как вам больше нравится — так и распространяйте.

А зачем аппстору юридические эксклюзивные права, если 99.9% пользователей iOS не имеют других источников?

DrPass 22 сентября 2018 в 03:44

−1

> И они даже не берут некоторый процент с продаж?

Ну, на ваши расходы на создание продукта это уж точно никак не влияет. Согласитесь, факт, что магазин берёт свою наценку при продаже вашего товара, абсолютно естественен и логичен.

> А зачем аппстору юридические эксклюзивные права, если 99.9% пользователей iOS не имеют других источников?

Зато пользователи Андроида тоже с удовольствием поиграют в те же Растения против Зомби, что и на аппсторе. А вот издатели иногда трясут ананайкой перед студиями, дескать, «хочу, чтобы в ЛастОфАс играли только на нашей любимой плейстейшен, и нигде более».



0xd34df00d 22 сентября 2018 в 05:17

+1

> Ну, на ваши расходы на создание продукта это уж точно никак не влияет.

Я склонен считать, что при отсутствии этого процента продукты в среднем стоили бы ниже.

VolCh 22 сентября 2018 в 18:11

0

Скорее приносили бы больше прибыли. Процент не такой большой, чтобы снизив цену на него получить аналогичный процент роста продаж. Стоит сейчас приложение 0.99 или 9.99 — снижать цену на 10% смысла очень мало.



0xd34df00d 22 сентября 2018 в 19:57

0

Так там вроде не 10% берут, а 30. И для уникальных нетривиальных приложений смысла действительно нет, а для какой-нибудь ерунды по 2.99 продавать её по 1.99 вместо этого — почему бы и нет?

DrPass 24 сентября 2018 в 01:38

0

При отсутствии этого процента разработчику надо было бы самому тратиться на дистрибьюцию игр. Поэтому вряд ли. Чудес и аттракционов невиданной щедрости не бывает. До появления маркетов игры обычно продавались непосредственно правообладателями и стоили намного дороже.



Sychuan 20 сентября 2018 в 23:04

+4

Суперкомпьютер НАСА в 60-ые был именно, что только в НАСА. У обычных людей ничего такого не было и в помине. И вы бы не захотели платить ту цену за компьютеры и программы, которые были в 60-ые годы. И я почему-то уверен, что в ужасном «тяжелом» вебе сегодняшнего дня вы смотрите кино, качаете игры со стима, ориентируетесь по гуглокартам и так далее. Ну а если вам нужны супервозможности 60-ых годов, то никто от вас ничего не закрывает. Браузер Lynx, какая-нибудь Windows Xp и можно спокойно считать траектории полета на Луну



Am0ralist 20 сентября 2018 в 22:47

+1

Windows 98 в эпоху, когда сжатие инфы больно дорогим накладным расходом занимал 245 метров после выкидывания всяких справок. Офис 97, хм... 100 мб установщик? И по сколько там каждое приложение в отдельности занимало из всего пакета?
Давайте, расскажите мне, что какая-нибудь прибluда на телефон, типа читалки сложнее этих программ при соизмеримом размере?
Да даже программа эмуляции командной строки под винфон, которая умеет только в пару комманд — 16 мб. Эмулятор нескольких команд доса, блин, который занимал сколько килобайт на дискетке?

НЛО прилетело и опубликовало эту надпись здесь

khim 21 сентября 2018 в 00:11

+6

> Не напомните, какой размер указателя в байтах был на windows 98, и каков он в «современных приблудах на телефоне», раз уж взялись мерить?

Припомню, да. 4 байта размер указателя в Windows 98 (сегмент 16 бит, смещение 16 бит), 4 байта сегодня (да, большинство приложения 32-битные до сих пор). И?

> При этом с помощью этого браузера я могу решить в тысячи раз больше задач, чем с помощью вашего 97го офиса, 98 винды, и каждого по отдельности приложения для нее.

Замётано. Не расскажете — как именно с помощью вашего браузера создать табличку с формулами и посчитать что-нибудь в ней. Сделать каталог с книгами и найти там чего-нибудь? Ну или, на худой конец, подготовить и распечатать книжку с картинками? Это всё Офис делал в 97й версии без каких-либо дополнительных аддонов или чего-либо ещё…

НЛО прилетело и опубликовало эту надпись здесь

khim 21 сентября 2018 в 01:35

+3

Если это «наглая ложь», то вы, я думаю, сможете привести десяток популярных приложений без поддержки armeabi-v7a.

Я про существование таких знаю, могу даж пример привести. Целое **одно** приложение, вы не поверите.

Все остальные — либо ABI-agnostic (то есть там может быть любой поинтер), либо с поддержкой armeabi-v7a.

P.S. Да, я знаю, что в мире iOS всё не так — ну так это, опять-таки, чисто и исключительно хотелки чьи-то. То ли Тима Кука, то ли духа Стива Джобса, я не знаю. Никакой объективной необходимости в таком «форсированном» переходе на 64-бита нет и не было.



0xd34df00d 21 сентября 2018 в 20:27

0

Я тут одну игру вспомнил, которая 16-20 гигабайт памяти жрёт только так. Интересно, как ей бы было с 32-битными указателями.

Хотя это десктопное всё, конечно.



Skerrigan 26 сентября 2018 в 11:33

0

Я такую точно знаю (из-за неё пришлось приобретать «до 64Гб» мать и обвес к ней) — это Cities Skyline

▸ Из коробки, junior-level

▸ После напильника, Skyline 'Skyline здорового человека', senior-level

Akon32 26 сентября 2018 в 14:24

0

> Интересно, как ей бы было с 32-битными указателями.

Иногда удивляюсь, почему "compressed pointers" так мало распространены. Выравнивание по 8 байт — и можно адресовать 32ГБ, используя 32-битные указатели. Немного памяти сбережёт.

khim 26 сентября 2018 в 22:13

0

Потому что позволяет решить очень узкую задачу (расшилить с 4GB до 64GB) путём усложнения кода буквально на всех уровнях. В языках без указателей (Java, к примеру), этот приём используют иногда, кстати.



Massacre 21 сентября 2018 в 02:48

+1

Конкретно для винды — вполне правда, это под линукс всё собирается, как правило, под битность дистрибутива, а так 64-bit only это, в основном, новые игры.

khim 21 сентября 2018 в 02:54

0

Речь шла про «современные приблуды на телефоне». Процент винды на телефонах — где-то под микроскопом нужно рассматривать.



YemSalat 21 сентября 2018 в 08:16

+3

> Не расскажите — как именно с помощью вашего браузера создать табличку с формулами и посчитать что-нибудь в ней. Сделать каталог с книгами и найти там чего-нибудь? ...

Гуглодокс и прочие веб приложухи.

И не говорите мне что 97ой офис не тормозил на компах того времени. Еще как тормозил. Ворд частенько падал при сохранении/загрузки нетривиальных документов, все они не показывали контент при скролле и т.п.

Попробуйте собрать себе старый комп с 98ой виндой и поработать под ним. Очень быстро захочется обратно в «тормозное будущее»

Neznauy 21 сентября 2018 в 10:32

+3

А причем здесь «гуглодокс и прочие веб приложухи»? Это совершенно отдельные вещи, живущие на далеких гугловских серверах.
А браузер, грубо говоря, парсит и генерирует http, и может открыть некоторые файлы для чтения.

НЛО прилетело и опубликовало эту надпись здесь



popov654 21 сентября 2018 в 17:45

+2

Полная глупость. Ну да, API много, систем много, но всё то, что умел офис старых версий, вы не просто не сделаете с той же лёгкостью, вы вообще этого не сделаете. Через веб-приложение — да, но при чём тут браузер?

Как верно заметили выше, браузер только парсит HTML, применяет CSS и исполняет скрипты; в довесок он умеет определять геопозицию юзера, показывать картинки и сохранять их на диск, проигрывать аудио и видео (уже лет 7-8 как), выводить и подсвечивать в консоли JS код, а также исполнять пользовательский код прямо там же. Но это обстоятельство не позволит вам сделать электронную таблицу, сверстать книгу или статью, или сделать слайд-шоу. Для всего этого нужно облачное приложение — браузер же может его только исполнить. Некорректно говорить, что функционал, предоставленный этим приложением, предоставлен непосредственно браузером. С таким же успехом можно сказать, что мы обязаны всему JVM или компилятору (браузер по сути такой же JIT-компилятор/интерпретатор).

InterceptorTSK 21 сентября 2018 в 18:24

0

Еще большая глупость заключается в том, что то что может быть исполнено на юзерской тачке в 100 раз быстрее — зачем-то сваливают на сервак. Это вообще верх идиотии.
А может и в 1000-100000 раз быстрее. Смотря что…



Neikist 21 сентября 2018 в 18:26

0

Вот ни разу. Откуда вы мощность машинки пользователя знаете? Может там i7, а может целерон. Или вообще какой нибудь андроид тысяч за 7-8.

YemSalat 21 сентября 2018 в 18:28

0

> Через веб-приложение — да, но при чём тут браузер?

Эээ… А браузер без веба зачем нужен то вообще?

khim 21 сентября 2018 в 21:09

+1

Дык в этом и дело! MS Office — является весьма функциональным приложением, которым можно пользоваться самим по себе, без ничего, браузер же, по сути — «современный тонкий клиент»… и при этом мне безапелляционно заявляют, что он умеет больше чем офис…

P.S. Это если забыть про то, что начиная с MS Office 2000 в офис и «настоящий» браузер входит. Конкретно в MS Office 2000 — MS IE 5.0

НЛО прилетело и опубликовало эту надпись здесь

swelf 22 сентября 2018 в 00:52

+2

excel и эта табличка? ну это что-то вроде
«Зачем тебе телевизор мультики смотреть, давай я тебе картинку нарисую»
Ты попытался сымитировать эксель и считаешь что браузер из-за этого не уступает по функционалу? Надо было на бумаге все посчитать и изображение вставить.

НЛО прилетело и опубликовало эту надпись здесь

swelf 22 сентября 2018 в 01:29

+2

> Я следовал поставленному ТЗ

но ты же его провалил, ни одного пункта не реализовано, кроме простой части первого пункта, я ж говорю на листе мог бы посчитать и скинуть фото.
1)Таблицы с формулами нет, только предзаданные вычисления
2)Каталога с книгами нет
3)Книги с картинками нет.

> При этом с помощью этого браузера я могу решить в тысячи раз больше задач, чем с помощью вашего 97го офиса, 98 винды, и каждого по отдельности приложения для нее.

Если под этими словами ты имел ввиду
«Я буду кодить новую таблицу под каждую задачу», то да можно в тыщу раз больше чем может эксель. Хотя очевидно у нас разные понятия под функционалом, когда 1 программа имеет функционал, а в другой необходимо руками реализовывать каждый конкретный случай, то спорно все же

> При этом с помощью этого браузера я могу решить в тысячи раз больше задач, чем с помощью вашего 97го офиса

Даю новый челендж, писать программы при помощи рисунков, рисунок может в тыщу раз чем браузер и эксель. Рисунком можно при должно терпении создать и браузер и эксель и любую программу. Ну знаешь такой метод, тыкаешь определенными цветами в пиксели, сохранешь БМПшку, переименовываешь в exe, вуаля у нас программа. Все, языки программирования/компиляторы/интерпретаоры сакс, рисунком можно че угодно закодить.

НЛО прилетело и опубликовало эту надпись здесь



popov654 22 сентября 2018 в 10:15

0

Видимо, речь шла о лёгкости реализации каждой такой задачи. Человек уверен, что в Экселе это делается проще и быстрее (наверное, и правда быстрее, но здесь многое уже индивидуально).

У меня вопрос по существу вашего примера имеется. Код простой, но у меня он, однако же, не работает. Ошибка очень забавна: "document.querySelectorAll(...).forEach is not a function". Перед этим я заменил стрелочную функцию на обычную, так как моя версия Хрома стрелочные ещё не поддерживает. Но тем не менее подсчёт итоговой суммы не работает. Я так понимаю, querySelectorAll возвращает в моей версии не массив, а NodeList или что-то вроде того, и у этой сущности отстутствуют методы массивов. То есть надо делать Array.prototype.forEach.call(...). Вопрос, почему у Вас работает без этого?

P.S. Не знал про темплейты, спасибо, теперь где-нибудь применю при случае :)

mayorovp 22 сентября 2018 в 14:13

0

querySelectorAll всегда возвращает NodeList, но в новых версиях у NodeList тоже есть метод forEach

НЛО прилетело и опубликовало эту надпись здесь



Kwisatz 22 сентября 2018 в 14:14

0

Коллега, поверьте, вы практически носитель сакрального знания. Функции для манипулирования DOM сейчас практически забыты. А чтобы создать подобную страничку современному разработчику нужно подключить jquery, react, less, настроить сброку итд итп

Зы foreach работает не везде

НЛО прилетело и опубликовало эту надпись здесь



0xd34df00d 21 сентября 2018 в 20:40

+3

Да и программа такая для этого не нужна, открываете командную строку, и copy con чётотам, и фигачите прям в машинных кодах.

DrPass 21 сентября 2018 в 11:55

+2

> И не говорите мне что 97ой офис не тормозил на компах того времени. Еще как тормозил.

На новом компе 1997-го (а это был бы Pentum 133+ с 16 Мб памяти и выше) года он работал адекватно, не тормозил. На компе 1999-го уже бы летал. Сложно сравнивать производительность тех лет с современной, т.к. тогда за год-полтора производительность вырастала в разы, а сейчас лет за пять процентов на сорок.



YemSalat 21 сентября 2018 в 18:30

0

О, ну на «новом компе 1997-го» любой дурак может…
У меня на новом компе 2018-го последний офис тоже не тормозит.

khim 21 сентября 2018 в 21:20

+4

> У меня на новом компе 2018-го последний офис тоже не тормозит.

А вы в этом **уверены**? Или вам так **кажется**?

Поставьте себе MS Office 2000 (он на Archive.org доступен), поиграйтесь пару дней — и, внезапно, MS Office 2016 начнёт тормозить.

Мы просто **забыли** что бывают программы, которые не тормозят.

И я даже могу сказать когда это произошло: 1984й год, вот тот самый ролик с молотком, явление GUI народу.

Потому что до этого времени всем было известно сколько времени занимает реакция компьютера на действия пользователя: 16.6ms, вот как клавишу нажали, так в следующем кадре — видим результат. Интерфейс весь под этот подгонялся: не можем перерисовать весь экран — думаем что можем за 16.6ms изменить, так и работаем.

А вот появление GUI сделало «патоку» нормальной… но это, в общем, ощущалось как нечто временное: да, оно тормозит пока, но потом машинки станут мощнее, и будет как в рекламе — красивые окошки и реакция за 16.6ms…

Однако прошло 30 с лишним лет, а «патока» — по прежнему на экране. Только разработчики игр знают, что может быть по-другому…



PsyHaSTe 21 сентября 2018 в 21:58

0

Ох уж эти свидетели ДОСа...

DrPass 22 сентября 2018 в 03:25

+1

> У меня на новом компе 2018-го последний офис тоже не тормозит.

Вы знаете, я на последний офис в этом плане тоже не жалуюсь, видимо, там ещё достаточно кодовой базы с тех лет осталось. Зато у меня на новом компе 2018-го года заметно глазу подтормаживает последний вайбер и скайп, просто при наборе сообщений. Вдумайтесь, клиент чатика, который со всем шифрованием и эмодзи по функционалу представляет несопоставимые крохи в сравнении с офисом, приторможивает на восьмиядерном процессоре и NVMe диске.



YemSalat 24 сентября 2018 в 17:02

0

Я как-то читал статью про разработку первых версий офиса (сейчас к сожалению не могу найти) — так там такие «ужастики», что скорее всего его переписали.
Ну и скайп никогда вроде как не был эталоном производительности и вообще хорошего софта.

DrPass 24 сентября 2018 в 18:03

0

Ну, крови Word 6 у меня в своё время попил, это факт. А вот на скайп вы зря наговариваете :) Я помню, как 15 лет назад говорил с заказчиком из США по скайпу, причем у меня интернет тогда был через диалапный модем, 49 килобит в хорошую погоду (сейчас про такие скорости никто уже и не помнит). Это было офигеть как круто.

DMGarikk 24 сентября 2018 в 18:31

+1

вот качество голосовой связи у Скайпа это наверное единственное что удерживает от полного отказа от него



Am0ralist 21 сентября 2018 в 09:55

+1

> я могу решить в тысячи раз больше задач

Сами назвали цифру, сами доказывайте.
Например, вы сможете запустить компьютер и без ОС… ах да.
Ок, тогда выйти через него в интернет… Черт, я забыл.
Обратится к прибору через ком-порт… да что ж такое-то.
Вывести страничку на печать на принтер… блин, опять ОС требуется.

То есть вы надстройкой над ОС сможете сделать больше задач, чем может ОС. Ну-ну.

НЛО прилетело и опубликовало эту надпись здесь



Am0ralist 21 сентября 2018 в 10:41

+1

Во-первых, вы так и не доказали, что те тысячи задач, которые вы можете выполнить в браузере — *больше* задач, которые может сделать винда.
А я вам лишь указал, что все свои задачи браузеры выполняют исключительно через ОС, а значит по умолчанию браузер в этом — второй.

Более того, а что может сделать ваш браузер БЕЗ сайтов, ну раз вы ОС отказываете в дровах, которые без нее работать и не могут? Ну или хотя бы без интернета? Что, оказывается, что бы гуглить код и смотреть ютубчики нужны не только браузер, а наличие сайта и интернета? То есть по факту, специально для этого написанные вещи?
А сам по себе он что умеет? Ах, считай — ничего.

После чего ваше высказывание про демагогию ниже — начинает играть новыми гранями. Вы б вначале думали, прежде чем в позу мушкетера вставать.

НЛО прилетело и опубликовало эту надпись здесь

novice2001 21 сентября 2018 в 12:11

+1

Т.е. Microsoft много лет занимается мошенничеством и получает миллиарды долларов за приложение, функции которого выполняет любой бесплатный браузер???

НЛО прилетело и опубликовало эту надпись здесь



Am0ralist 21 сентября 2018 в 12:34

0

> Как минимум то, что может офис — он может и без интернета. И даже больше.

Ну давайте. Начнем. Создайте мне в голом браузере электронную таблицу с кучей формул, аналогом макросов и нормальным интерфейсом взаимодействия с пользователем. Без интернета, сайтов и расширений, ибо вы указали ос без драйверов типа ничего не может, а значит в таком сравнении браузер это все может из коробки без перечисленного. Вот прям открываешь браузер и в нем.

НЛО прилетело и опубликовало эту надпись здесь



Am0ralist 21 сентября 2018 в 13:30

0

«Повышение ставок», недопустимый прием в споре. Что вы там говорили про демагогию и про тысячу вещей, которые можете делать в явно пустом браузере и не смогли бы в ОС без драйверов?

НЛО прилетело и опубликовало эту надпись здесь



Am0ralist 21 сентября 2018 в 22:43

0

> Тут нет никаких повышенных ставок.

А вы как-нибудь логику ведения споров осильте, компетентный вы наш. Может узнаете, что такое недопустимые приемы и демагогия на самом деле.

> А вы крайне некомпетентны.

да здесь все не согласные с вами крайне некомпетентны. один вы, товарищ мушкетер.

novice2001 21 сентября 2018 в 13:42

+1

У нас и доказательства есть, но они секретные, мы их не покажем

InterceptorTSK 21 сентября 2018 в 14:53

0

А причем тут плата?))
Договоритесь со знакомым Васей, он вам сделает Ворд, а вы ему Эксель)
Можете обсудить стоимость. Только она не важна) Вы можете сделать друг-другу куски офиса скажем за мильярд долларов. Хватит? Но взаимозачет же. Он вам. Вы ему.
Только фишка же получается не в деньгах вовсе) Ни он ни вы абсолютно ничего не сделаете.
Выводы.

Причем тут фишка еще и в том, что тут совсем деньги ни при чем. Потому что вы и так знаете, что Вася вам ничего в итоге не сделает) А значит и вы не захотите. Смысла же нет.

Если вы ваш скил обратите в экономические бумашки — эти бумашки вообще ничего не будут стоить. Потому что вы ничего не можете вовсе) Потому что есть другие, которые могут. Которые сделают круче, быстрее и дешевле, а скорее всего вообще задаром.

И да, браузер тут вообще не нужен.

fukkit 21 сентября 2018 в 16:44

0

> Договоритесь со знакомым Васей, он вам сделает Ворд, а вы ему Эксель) Можете обсудить стоимость. Только она не важна) Вы можете сделать друг-другу куски офиса скажем за мильярд долларов.

Идея богатая и в некоторых кругах весьма уважаемая. У каждого фигуранта в этом случае остается компания с миллиардным оборотом. Остается отрастить бороду, подвернуть штаны — и на ICO, а того краше — на IPO.

popov654 21 сентября 2018 в 17:54

0

> Отображать pdf

Только если стоит плагин от Adobe, либо, опять же, через JS расширение, написанное сторонним разработчиком (да, его иногда включают в поставку, но у меня даже нет уверенности, что его пишут те же люди, что сам браузер).

> Предоставляет инструменты для их верстки.

Да, но это верно только для HTML, в какой-то другой формат экспорт сделать будет вряд ли возможно)

> С помощью браузера и написанного на коленке скрипта, я могу p2p позвонить знакомому по webrtc, без всяких скайпов.

Окей, что-то такое анонсировали, как минимум в Mozilla. И как это сделать? Что для этого нужно знать, куда нажать и что куда ввести?

> И делается это движением мизинца правой ноги.

Вы сильно утрируете. Хотя в целом такое и правда возможно, для создания веб-приложения не нужен сервер, можно работать и локально, и даже делать импорт/экспорт в файлы, и использовать прикольные штуки вроде Indexed DB, WebSQL и Local Storage.

НЛО прилетело и опубликовало эту надпись здесь

Am0ralist 21 сентября 2018 в 22:49

0

> Любое более-мене крупное ПО состоит из компонентов написанное разными людьми. Вы снова скатываетесь в демагогию. Я пользуясь вашими аргументами расщеплю вам любую программу на составные части и скажу, что вы не можете сделать ничего с помощью этой программы, потому что компоненты написаны не разработчиками конкретного софта.

Итак, читаем вас же:

> Во-вторых перечисленное вами не может сделать ни одна ОС, без наличия соответсвующих драйверов на оборудование.

Вы только что на глазах у изумленной публики расщепили программу на составные части и заявили, что она ничего не может. После чего обвиняете других в демагогии и некомпетентности?

НЛО прилетело и опубликовало эту надпись здесь

Am0ralist 22 сентября 2018 в 09:58

0

> То, как вы стойко отстаиваете и несете чушь, каждый раз все больше повышая ставки в споре, что очень хорошо характеризует вас. Вы написали про тысячи задач, но перечислили десяток, слившись даже на попытке сделать лично рыками эксель (ибо нифига у вас и близко не получилось).
> А приводить приложения, которые кто-то написал под браузер — это ещё больше нести чушь, потому что задачи будет решать не браузер, а он будет прослойкой. Еще более тонкой, чем ОС.

НЛО прилетело и опубликовало эту надпись здесь

НЛО прилетело и опубликовало эту надпись здесь

bykvaadm 21 сентября 2018 в 13:10

0

и чем эта урезанная ОС, в которой есть только браузер отличается от концепции ос + браузер? сам браузер без ОС этого ничего не сможет

Alexeyslav 21 сентября 2018 в 14:09

0

Надо сказать что специализированное приложение без ОС тоже ничего не сможет сделать, даже запустится.

VolCh 22 сентября 2018 в 18:20

0

Не помню у какого вендора, но уже встречал браузер без ОС, только БИОС.

IRainman 23 сентября 2018 в 13:57

0

Хромбуки от гуглы наверное?

VolCh 23 сентября 2018 в 16:57

0

Нет, имеется в виду обычные материнки какого-то вендора, с которыми можно серфить из биоса, без ОС на диске и вообще диска.

Alexeyslav 24 сентября 2018 в 10:11

0

UEFI-биосы так называемые… но это вранье! Там только кажется что нет никакой ОС, а на деле начнёшь копать а там… обычный микроскопический линукс, с полноценным(насколько это возможно) браузером. Всё это сделано на уровне абы было поэтому не оптимизировано под многозадерность, большое количество памяти и еле шевелится. И всё это работает с… диска виртуального, мини-SSD встроенный в материнку на 8…16Гб.

Gordon01 21 сентября 2018 в 10:25

0

Все что было для windows mobile было гораздо функциональнее. SoftMaker Office — повторял функционал НАСТОЛЬНОЙ версии ms office на КПК. Почтовые клиенты тоже были такого уровня.
Программы от Resco — тоже образцы невероятного функционала.
При этом все это очень быстро работало на мобильных процессорах того времени (200-500 МГц), 64-128 МБ памяти.
Но глючновато, винмобайл часто любил зависать наглухо.

Alexeyslav 21 сентября 2018 в 13:39

0

Потребности людей с тех пор сильно уменьшились. Зачем на мобильном клиенте выбор кодировки письма? настройка mime-типов? способы кодирования вложений? Оно просто должно работать… причем без лишних заморочек — на мобильных девайсах возможность ввода и взаимодействия с приложением ограничена, а значит и многие функции становятся невостребованными.
Кстати, а почтовые клиенты прошлого умели передавать файлы больше 100мб? а 10Гб? Сейчас это на уровне средних потребностей.

DrPass 21 сентября 2018 в 13:57

+1

> Кстати, а почтовые клиенты прошлого умели передавать файлы больше 100мб? а 10Гб? Сейчас это на уровне средних потребностей.

Ну не перегибайте, ни раньше, ни сейчас почтовые клиенты для передачи подобных файлов не используются. И проблема тут не столько в клиенте (дайте ему достаточно памяти и времени, он вполне себе сжуёт такой файл), сколько в ограничениях на размер вложений у почтовых сервисов. Что касается самой возможности обмена крупными файлами, то она не слишком-то и изменилась, единственное, сейчас у вас будет файлообменник с веб-мордой, а двадцать лет назад у вас был бы файлообменник с FTP-клиентом. И дисковая квота, конечно, с поправкой на размеры файлов тех лет :)

Alexeyslav 21 сентября 2018 в 14:18

0

Почему же? вот совсем недавно отправлял фото для печати в онлайн-сервис, почта… 5Гб. Приложил вложение вжуух и письмо ушло. Всё абсолютно прозрачно, никаких плясок с какими-то там файлообменниками, настройками FTP-клиента и т.п. всё делается автоматически сервисом. И, кстати, надо заметить никаких оверсайзов на 140% из-за BASE-64 кодирования вложения.
Технологии вроде бы как не изменились, но пользовательский опыт — очень сильно.
Почему, кстати, досихлор не видно отдельных нативных почтовых программ способных автоматически передавать большие файлы вложений через файлообменники?

DrPass 21 сентября 2018 в 14:30

0

> Почему же? вот совсем недавно отправлял фото для печати в онлайн-сервис, почта… 5Гб.

Ну а я так ни на одном из своих ящиков не смогу. И никто из тех пользователей, у кого нет специального корпоративного ящика с отсутствием лимита на вложения, так не сможет. Gmail сейчас имеет лимит на вложения 25 мегабайт, Mail.ru — 20 мегабайт. И подобные лимиты практически везде. У вас это редкое исключение, связанное с профессиональной деятельностью.

> Почему, кстати, досихлор не видно отдельных нативных почтовых программ способных автоматически передавать большие файлы вложений через файлообменники?

Кстати, идея очень даже неплохая :)

Alexeyslav 21 сентября 2018 в 15:33

0

Нет, обычный GMAIL, и даже маилрушечка так умеет. Про яндекс не знаю, но наверняка и его уже научили. Конечно, они не гоняют письма по почтовым серверам — они для этого используют промежуточное хранилище и делают это совершенно прозрачно обходя ограничения почтовых серверов. Будь там даже 1Мб ограничение.

popov654 21 сентября 2018 в 17:59

+4

Так это делается совсем не так. Просто файл заливается на их собственный сервер, а к письму прикрепляется ссылка. Технически, будь такие компании с такими объёмами для хранения в 90-ые — это технически было бы возможно и в 90-ые. Тут нет какого-то особенного «ноу-хау» :)

Alexeyslav 21 сентября 2018 в 23:58

0

Так ведь фишка в том что это делается прозрачно для пользователя. У — удобство.

khim 22 сентября 2018 в 03:05

0

Вот только прозрачность эта — мнимая. Как вам уженавязчиво указали.

Да, если все вокруг пользуются GMail'ом и GDisk'ом — то это работает. Но шаг вправо, шаг влево… и нет больше прозрачности…

SUA 21 сентября 2018 в 14:56

0

вот совсем недавно тоже отправлял сканы в обыкновенную корпоративную почту (банки — не «совсем безнадежные» должны быть по используемым технологиям)…
один робот мне ответил что их предел 25Мб,
второй — что 20,
один человек — «шлите все только в pdf, картинки не открою»
и двое — «приложенные ссылки гуглдиска не открываются, шлите только вложением»
и это при том что мобильный клиент гугла умеет напрямую с интернет-диска гугла же только ссылки шарить (хорошо что наоборот еще отключить не догадались — с приложения гуглдиска можно отправить письмо с вложением)
так что «вжжух и ушло» — это несмотря на 2018 ситуация до сих пор из серии «любой каприз за ваши деньги»

staticlab 23 сентября 2018 в 15:41

+2

> банки — не «совсем безнадежные» должны быть по используемым технологиям

ИБ — информационная безопасность

> «шлите все только в pdf, картинки не открою»

Скорее всего регламент.

> «приложенные ссылки гуглдиска не открываются, шлите только вложением»

Google Drive запрещён как файлообменник.

popov654 10 ноября 2018 в 23:38

0

> Google Drive запрещён как файлообменник.

А для чего он тогда создавался, для личного архивирования? Там вроде в пользовательском соглашении ничего нет про запрет передачи ссылок.

staticlab 11 ноября 2018 в 12:11

0

Речь идёт про банки. В целях безопасности у многих сотрудников там вообще интернета нет. А даже если он и есть, то многие ресурсы, такие как веб-интерфейсы почты, мессенджеров, файлообменники и т.п. просто запрещены. Потому-то ссылки на гуглдиск они открыть не смогут, и нужно присылать файлы вложением.

popov654 11 ноября 2018 в 12:24

0

Какой смысл закрывать доступ к файлообменникам, если почта (и файлы в ней) разрешены? Странная логика. Точно так же могут что-то получить или отослать, из того что нельзя согласно правилам.

> В целях безопасности у многих сотрудников там вообще интернета нет.

Чтобы не слили данные по операциям (соблюдение банковской тайны)? Разумно. Но у кого совсем нет интернета, там и почты не будет. А у кого интернет есть, те обычно к особо секретным базам и не допущены (отдел по работе с клиентами какой-нибудь).

staticlab 11 ноября 2018 в 12:36

+2

> Точно так же могут что-то получить или отослать, из того что нельзя согласно правилам.

Входящие вложения всё-таки проверятся корпоративными политиками и антивирусом. А вот исходящие за пределы домена отослать тоже не все могут — на это нужно отдельное разрешение.

> Но у кого совсем нет интернета, там и почты не будет.

Вот и выросло поколение… Почта с внутрикорпоративного сервера работает. Через почтовый клиент типа Outlook или Lotus.

> А у кого интернет есть, те обычно к особо секретным базам и не допущены (отдел по работе с клиентами какой-нибудь).

Зачем отделу по работе с клиентами интернет?

popov654 11 ноября 2018 в 13:24

0

> Почта с внутрикорпоративного сервера работает. Через почтовый клиент типа Outlook или Lotus.

Нет, ну это для компании логично. Я имел в виду, у кого нет физически доступа в сеть. То есть значит сеть есть, но лишь локальная, нет именно интернета? Я просто уже привык, что это синонимы, сорри.

> Зачем отделу по работе с клиентами интернет?

Например, чтобы работать с CRM. Хотя скорее всего, она тоже локально развёрнута, в интрасети.
А ещё чтобы уточнить детали тарифа/услуги, хотя наверное для этого у сотрудников стоит специальный справочный софт, не нужно даже корпоративный сайт открывать.

staticlab 11 ноября 2018 в 14:54

+2

Конечно обычная локальная сеть есть. В ней свои интранет-сервисы, конечно, крутятся, типа корпоративного портала, CRM, баз знаний и СЭД, но к интернету доступа у большинства не будет в принципе. Впрочем, он на самом деле сотрудникам для их обязанностей и не нужен.

Alexeyslav 12 ноября 2018 в 11:33

0

Почта проверяется централизованно антивирусом, если есть подозрение — файл/вложение не пропускается. Проверять же все обращения к файлообменникам это решение уже гораздо дороже.

encyclopedist 10 ноября 2018 в 23:22

0

> Почему, кстати, досихпор не видно отдельных нативных почтовых программ способных автоматически передавать большие файлы вложений через файлообменники?

Thunderbird умеет. Outlook тоже умеет.

Alexeyslav 12 ноября 2018 в 11:35

0

Что-то им это не помогло, верней не стало ключевой особенностью при выборе.

DrPass 12 ноября 2018 в 12:20

0

Ключевой особенностью при выборе стало то, что в веб-почте достаточно зарегистрироваться и приступить к работе. Не надо ничего устанавливать и настраивать. Но мы-то говорим не в контексте того, что больше нужно пользователям, а в контексте того, почему старые приложения с бОльшим функционалом работали качественнее, чем современные.

AmOralist 21 сентября 2018 в 14:06

0

> Кстати, а почтовые клиенты прошлого умели передавать файлы больше 100мб? а 10Гб? Сейчас это на уровне средних потребностей.

Послать то они могли, только и сейчас на метрах 15-30 тебя входящий сервачок завернет очень часто. Используйте всякие сервисы для расшаривания файлов, говорят они.



Alexeyslav 21 сентября 2018 в 14:11

0

Вот. Вы говорите именно о почтовом приложении, которое кроме POP3 и IMAP протоколов не знает ничего, но работает быстро. Собственно это ограничения не интернета а протокола и сети почтовых серверов.

khim 21 сентября 2018 в 14:07

0

Но если потребности людей уменьшились, то почему потребности приложений увеличились?



Alexeyslav 21 сентября 2018 в 14:27

0

Потребности в стороннем, вспомогательном функционале. Нынешние пользователи пугаются даже слов сервер, порт и необходимость куда-то вводить эти данные в приложении вводят их в ступор. Как впрочем и большие перечни настраиваемых параметров приложения.
Вспомните, какой прорыв в голосовой связи через интернет произвёл скайп. Который не надо настраивать, не надо вручную прокидывать какие-то там порты в NAT(ещё одно страшное слово для пользователей) и куда-то вводить адрес прокси-сервера. Он просто работает. Запустил и работает. В этом плане потребности пользователей уменьшились, они хотят и получили заветную кнопку «сделать всё хорошо».

sumanai 21 сентября 2018 в 15:23

0

Как будто список настроек оправдывает рост объёма приложений на сотни мегабайт.

popov654 21 сентября 2018 в 18:02

–1

Но разве так жить интересно? Это же деградация в чистом виде какая-то…

То есть пользователь, конечно, не обязан становиться разработчиком и даже админом. Но всё-таки… Хоть для себя-то какой-то кайф получить от того, что сделал, настроил, и оно заработало в итоге



SirEdvin 21 сентября 2018 в 18:18

+1

Столько много всего можно настроить, что чем меньше, тем лучше, на наш век все равно хватит.



Alexeyslav 22 сентября 2018 в 00:04

0

Это не деградация, это специфика мобильного девайса. У него ограниченые возможности по вводу, особо не разгонишься. И ведь это прекрасно когда программа просто работает и не нуждается в низкоуровневой настройке — все варианты опробованы, найдена оптимальная конфигурация и зафиксирована.

VolCh 22 сентября 2018 в 18:24

+1

Не все от этого тащатся вообще, а некоторые раньше тащились, а теперь перестали. Я, например.

DrPass 24 сентября 2018 в 01:47

0

> Хоть для себя-то какой-то кайф получить от того, что сделал, настроил, и оно заработало в итоге

Кайф можно получить один раз — когда вы решаете новую для себя задачу. Когда вы настраиваете софт на своём десятом компе, эта задача вызывает у вас уже не кайф, а раздражение и скуку.



third112 25 сентября 2018 в 14:57

0

> Так что проблема отнюдь не в том, что для разработки более качественного ПО нужно намного больше денег.

ИМХО нужно больше времени. Раньше обновления были не очень часто, а сейчас лидеры индустрии ПО похоже считают, что чем чаще — тем лучше для их фирмы-производителя этого ПО.

darthmaul 20 сентября 2018 в 19:34

–7

Не надо многотысячных КБ для разработки софта, достаточно выкинуть «свистелки и перделки» из имеющегося. Замечательный пример — современный вэб. Что нужно для сайта? Ответ прост — HTML, CSS, PHP/Python, SQL — совместимая БД. Можно еще написать на JS несколько простейших функций для подтягивания контента без перезагрузки (тех же комментов). А все остальные мегабайты «нод джс», «джхверри», другие фреймворки и прочие кучи дерьма работают лишь ради «красивеньких» анимаций и прочих перделок.

khim 20 сентября 2018 в 20:14

+22

> А все остальные мегабайты «нод джс», «джхверри», другие фреймворки и прочие кучи дерьма работают лишь ради «красивеньких» анимаций и прочих перделок.

Но ведь это же **нормально**!

Люди, которые думают иначе либо не читали пресловутого секрета айберга, либо не до конца осознали его.

Прицитирую самую важную часть:

> Вы знаете, что 90% айсберга находится под водой? Ну и с большинством программ то же самое – есть красивенький интерфейс, который занимает 10% работы и потом 90% программистской работы «за кулисами». И если вы примете во внимание, что около половины времени уходит на исправление ошибок, то на пользовательский интерфейс уходит только 5% работы. И если вы ограничиваете себя только визуальной частью интерфейса, картинками, которые показываются в PowerPoint, то мы говорим сейчас менее, чем об 1%.
>
> Это не секрет. Секрет в том, что *люди, которые не являются программистами, не понимают этого.*

Сейчас мы наблюдаем простое и **естественное** следствие из этого закона: если люди озабочены только и исключительно пискелями на экране — то только на оптимизацию этого параметра и будет обращено внимание.

И, кстати, зря он программистов отделил. Они точно также считают. Я это понял, когда увидел фразу Есть еще Gerrit, который там как-то с этим борется, но интерфейс из 90х убивает все желание пользоваться этим. Речь шла об управлении VCS для программных проектов. Уж гиковой задачи быть не может. И… всё равно: пиксели — важнее.

Кстати разработчики Gerrit'а отреагировали адекватно: последняя версия выкинула нафиг «устраивавший» интерфейс, который позволял мне с клавиатуры сделать то, что мне нужно за секунды. Теперь это «чудо природы» грузится многие секунды (иногда десятки секунд), глючит, но… зато у него теперь современный интерфейс! Победа!

Автор статьи вообще не понимает, где причина, а где следствие: машину с потреблением топлива 1000 листров на 100 километров — никто не купит. А сайт, который теряет данные, будет пользоваться большей популярностью, чем сайт «с интерфейсом из 90х». Вот и всё. Вот и вся «тайна».

Люди — вы получаете ровно то, за что вы платите (деньгами или временем — неважно), почему вы недовольны? Пока куча дерьма в красивой обёртке пользуется большей популярностью, чем маленькая, быстрая, но «негламурная» программа — люди будут порождать звереств дерьма.

Почему автор вообще к разработчикам обращается? **Не в них ведь дело!**

engine9 20 сентября 2018 в 20:44

–1

Тут, батенька, вот еще какое дело. Капитализм выращивает (*нытик-фильм на 3 часа*) невежд, так как фрустрированному тревожному дураку проще все это барахло и скармливать.

Тому кто не может планировать — кредиты с конскими процентами. Тому, кто озабочен статусом — оверпрайснутые цацки. Тем, кому промыли мозги и насадили чувство неполноценности — уколы силикона в губы и амулеты на успех.

Дурака растят под рынок и рынок под дурака, конвергенция-с. Но это только часть вопроса, ибо цифровая техника и все эти айтишные абстракции крайне тяжелая и не очевидная для человеческого мозга вещь! Так как не имеет привычных аналогий в реальном мире. Если, например, файл на диске можно сравнить с вещью, то жесткую ссылку уже намного сложнее.

Т.е. требовать от большинства людей хотя бы базового понимания как это все устроено нерационально. И сопоставимо с требованием от всех понимания физиологии человека или электродинамики. Это в мире где большинство верит в ящериков и думает, что Солнце вращается вокруг ~~плоской~~ Земли.

Neuromantix 20 сентября 2018 в 21:07

+16

Во всем всегда виноват капитализм. Вот в других странах нет ничего, кроме ядерных ракет и медведей, зато у всех мозги — так мозги, кредитов нет, цацок дорогих нет, дешевых — тоже нет, ничего нет, кроме ЧСВ раздутого «зато не капитализм».
Причем тут капитализм? Просто производство хламья очень просто, дешево и удобно — думать не надо в первую очередь разработчикам и производителям.
Сейчас зачастую даже имея неограниченные деньги, некоторых нормальных вещей не купить — разве что в виде завода, и потом дрючить его до тех пор, пока руки переместятся с места ниже поясницы на свое нормальное.

engine9 20 сентября 2018 в 21:27

0

Окей, я могу плавать в терминах или же вообще ошибаться. На мой взгляд корни у всей этой байды в том, что экономику «разгоняют» через стимуляцию спроса. А именно путем агрессивного маркетинга, инфляции и запланированного устаревания. И т.ч. софтового.

Еще вмешательством в образование, которое (мало того, что само превращается в модную услугу, впариваемую родичам через тревогу за детей) не позволяет сформировать у человека целостную картину мира. Производя «радостных олигофренов» чья уверенность следствие эффекта Даннинга-Крюгера.

Neuromantix 20 сентября 2018 в 21:36

+2

Не экономику разгоняют, а разгоняют цифирь показателей (рост экономики в процентах, число обновлений софта за год, количество моделей за сезон и тд). Плюс маркетинг — это такая религия, которая верит в силу самой себя. Маркетологи, а за ними программисты тверднят: «Нужно больше обновлений, потому что людям нужно новое бла-бла-бла». Только вот я с сотню людей спрашивал — все они **ненавидят** обновления лютой ненавистью, и зачастую не ставят их. Т.е. получается, что религиозники-маркетологи придумали миф, и подгоняют все под этот миф, а то, что реальность с ним не соотносится — тем хуже для реальности.
Ничто не мешает писать не кривой софт, и сократить число обновлений до одного в год-два вместо бешеного клепания бесконечных обнов, которые делают только хуже. И это будет даже дешевле, чем сейчас, но никто это не делает, потому что производители живут в своем вымышленном мире и пытаются следовать его законам, которые в реальном мире не работают или работают не так.

khim 20 сентября 2018 в 21:53

+3

> Только вот я с сотню людей спрашивал — все они **ненавидят** обновления лютой ненавистью, и зачастую не ставят их.

Вот только выход обновлений — приводит к увеличению продаж/скачиваний/etc.

Вы забываете что задача маркетолога заключается не в том, чтобы сделать людей счастливее, а в том, чтобы увеличить доход и, в идеале, прибыль.

И тут получается парадокс: те пользователи, которые у вас уже есть — они обновления ненавидят… но они у вас **уже есть**, и, в среднем, никуда не деваются… а вот **новых** выпуск обновлений с новыми фишками привлечёт с большей вероятностью, чем исправление багов.

Tantacula 20 сентября 2018 в 22:40

0

> Ничто не мешает писать не кривой софт, и сократить число обновлений до одного в год-два вместо бешеного клепания бесконечных обнов, которые делают только хуже.

Как минимум мешает невозможность учесть все nullday эксплоиты, повышенные затраты времени и денег на усиленное тестирование (которое опять же не гарантирует 100% отсутствие багов, а лишь увеличивает вероятность их обнаружения) и тот факт, что значительная часть софта — это не изолированная вещь в себе, а программа, общающаяся со сторонними компонентами, которые часто меняются. Вы выпустили клиент для инстаграмма, а завтра фб объявил об изменении api или вовсе закрыл доступ к нему. Вы выпустили графический редактор с поддержкой всех существующих камер, а через неделю олимпус объявил о выходе 5 новых фотоаппаратов с новым форматом рав, либо конкурент встроил функцию новую, без которой ваш продукт стал неинтересен аудитории. Вы уже отчаялись и написали простой блокнот для айфона, а тут эппл решил отказаться от поддержки 32-битных приложений в новых версиях ios.

muon 21 сентября 2018 в 08:58

+2

> Как минимум мешает невозможность учесть все nullday эксплоиты, повышенные затраты времени и денег на усиленное тестирование

Это же противоречивое высказывание. Качественное тестирование (вместе с качественной разработкой) спасёт вас от null-day эксплойтов.
И не говорите, что это невозможно. Сколько дырок нашли в AS/400? В системах крутятся бухгалтерии космических масштабов, наверняка есть желающие их расковырять. И не расковыряли почему-то.
А разгадка проста: AS/400 делается для людей, которые умеют зарабатывать и считать большие деньги. А условный айфон — для закомплексованных потребителей, которые считают, что за новый телефон их сильнее будут уважать коллеги по опенспейсу. И новые сайты с 10-мегабайтными страницами — для них же, и будут посещать, потому что конформное поведение.

Tantacula 21 сентября 2018 в 18:24

0

> И не говорите, что это невозможно. Сколько дырок нашли в AS/400? В системах крутятся бухгалтерии космических масштабов, наверняка есть желающие их расковырять. И не расковыряли почему-то.

Разгадка проще — вы либо не слышали об этих nullday уязвимостях (вы слышали о stuxnet, heartbleed или meltdown до их появления? насколько публично IBM вообще объявляет о своих уязвимостях? Или они всегда пишут идеальный код?), либо система представляет собой неуловимого Джо (экономически целесообразнее ломать конкурентов, либо другие компоненты системы, особенно если атака точечная).

Ну не может единоразовое тестирование системы никак закрыть все баги и уязвимости. Не с нынешними технологиями. В таких компаниях как google, facebook, microsoft тестирование включает в себя регулярно работающую bug bounty не потому, что они наняли идиотов разработчиков. Сколько лет прошло с момента запуска интелом уязвимой архитектуры и до момента обнаружения spectre/meltdown? Тоже скажете, что в интел дураки сидят?

muon 24 сентября 2018 в 08:43

0

> вы слышали о stuxnet, heartbleed или meltdown до их появления?

Слышал после.
А вы слышали о дырах в System i, до или после?

> скажете, что в интел дураки сидят?

Интел и остальные упомянутые делают ширпотреб. Они не делают идеальных продуктов, ну так за идеальность им никто и не платит. Выгоднее приделать новые фичи, чем вылизывать старые — такая ниша.

khim 24 сентября 2018 в 09:13

+1

> А вы слышали о дырах в System i, до или после?

А вы вообще много с System i общаетесь? Дыры там тоже есть: одна, две, три, четыре… и процессоры тамошние тоже уязвимы…

Так что дыр в ваших любимых AS/400 — примерно столько же, сколько в процессорах Intel или AMD. Спасает их одно: на них кошечек из YouTube не смотрят… потому проще что-то сделать через одну из тысяч дыр в системе, через которую с System i общаются, чем взламывать собственно System i через одну дырку и нескольких десятков, что там есть…

muon 24 сентября 2018 в 10:45

0

> Так что дыр в ваших любимых AS/400 — примерно столько же, сколько в процессорах Intel или AMD.

Ну да, что там больше нуля, что тут больше нуля. Столько же, чо.

khim 24 сентября 2018 в 17:10

0

> Ну да, что там больше нуля, что тут больше нуля. Столько же, чо.

И там и там их [пока] накопали больше нуля, но меньше десятка. Что сравнимо (когда у вас есть одна-две уязвимости оценить дисперсию очень сложно).

А вот в «обычных» операционках — их тысячи!

crystallize 24 сентября 2018 в 15:42

0

> Сколько лет прошло с момента запуска интелом уязвимой архитектуры и до момента обнаружения spectre/meltdown? Тоже скажете, что в интел дураки сидят?

Нет, чекисты :)

DrPass 24 сентября 2018 в 16:42

+1

> А разгадка проста: AS/400 делается для людей, которые умеют зарабатывать и считать большие деньги.

Разгадка, как мне сдаётся, в том, что к айфонам и сайтам доступ имеют миллиарды людей, из которых десятки тысяч ищут уязвимости. А к iSeries доступ имеют десятки тысяч людей, из которых уязвимости ищут единицы. Мне довелось лет десять назад работать с iSeries, я не искал там уязвимости, но по субъективным ощущениям количество индусов, пишущих под неё софт, ничуть не меньше, чем в других сферах.

Alexeyslav 21 сентября 2018 в 10:59

0

Сократить количество обновлений просто нельзя — банальные ошибки в софте и уязвимости будут оставаться незакрытыми годами… Посмотрите на нынешние обновления. Что-то кардинально меняется в интерфейсе? Ну, может быть… раз в год. А в остальном обновления касаются исключительно внутренней скрытой части софта, что не видно глазами но сильно влияет на его функционирование в современном мире. Перестанешь обновлять софт — через год он окажется дырявым как сито и никому не нужным т.к. конкуренты залатали дырки и распространили с очередным недельным обновлением.

vin2809 21 сентября 2018 в 15:33

+1

Так может нужно сразу выпускать РАБОТАЮЩИЕ приложения, прошедшие тестирование. А не обновлять (читайте исправлять) «внутренние».

Alexeyslav 21 сентября 2018 в 23:36

0

Предлагаю попробовать прикинуть сколько надо ресурсов чтобы проверить простейшую программу которая складывает два 32-битных числа, а именно полный её тест. А потом приступить к надёжному тестированию более сложных программ. Без полного теста всех вариантов состояний программы и переходов между ними никакой гарантии отсутствия ошибок в программе быть не может. А пока такие программы выходят в свет лишь в надежде что АЛУ процессора нас не подведёт, и как правило так и бывает. Но есть нюанс…

0xd34df00d 22 сентября 2018 в 00:03

+1

> Без полного теста всех вариантов состояний программы и переходов между ними никакой гарантии отсутствия ошибок в программе быть не может.

Звонили ненужные для всего этого JS-буйства красок академики, просили передать про системы автоматического доказательства.

Хотя...

> А пока такие программы выходят в свет лишь в надежде что АЛУ процессора нас не подведёт, и как правило так и бывает.

А, так вы на каждой целевой машине тестировать будете? Ну ладно.

Alexeyslav 22 сентября 2018 в 00:13

0

А как ещё можно доказать что программа действительно правильно складывает ЛЮБЫЕ числа? Целевые машины непричём — мы проверяем программу. Программа — черный ящик, она складывает числа. На входе два 32-битных числа, на выходе — результат. Нам надо убедится что программа работоспособна и правильно работает для любых чисел на входе. Сейчас мы просто предполагаем что такие простейшие операции не нуждаются в проверке и заранее считаем их валидными, но если такой уверенности нет? Механизм сложения, допустим, нетрадиционный и в нём проверяется множество условий. Каждая такая проверка увеличивает количество вариантов состояний которые надо проверить, и с ростом сложности программы оно растёт очень быстро. Даже для современного варианта «хелло-ворлда» количество состояний стремится к непристойной величине, и даже в такой программе может возникнуть сбой по вине операционки или скажем драйвера видеокарты.

0xd34df00d 22 сентября 2018 в 00:20

0

Ну если у вас нетрадиционный вариант сложения, скажем, myplus, то вы берёте и пишете другую функцию, например, myPlusEqualsPlus с типом (a, b : Int32) -> (myplus a b = a + b) (как говорит Карри-Говард, этот тип эквивалентен утверждению, что для любых a и b значение myplus a b равно значению a + b), пишете её реализацию (как говорит Карри-Говард пополам с теорией типов, реализация является доказательством соответствующего утверждения) и радуетесь жизни.

Правда, при написании доказательства от чёрноящиковости придётся доказаться.

Или не отказаться и использовать внутреннюю верификацию, тогда на внутренность myplus смотреть не нужно, но её сигнатура изменяется с myplus : (a, b : Int32) -> (c ** a + b = c) . Зависимый тип-сумма, это всё.

Как, по-вашему, математики теоремы доказывают? Проверяют все возможные значения переменных?

Alexeyslav 22 сентября 2018 в 00:39

+1

Масло масляное, в этом примере альтернативного сложения мы используем тот же АЛУ что и оригинальный вариант. Но вот допустим программа использует свою реализацию двоичного сложения чисел. Та что используется в АЛУ проверена вдоль и поперёк и даёт гарантию что работает корректно(хотя вообще это утверждение нуждается в проверке, вдруг для какой-то пары чисел возникает сайд-эффект приводящий к неверному результату?) но мы должны проверить свой алгоритм. Эта задача в некоторой степени очевидна, но уже приводит к определённым трудностям и сложностям доказательства. Что будет с более сложными программами?
Ошибки в арифметике процессора в истории были, так что даже алгоритмы которые мы считаем надёжными на самом деле таковыми не являются и по большому счету даже такие элементарные вещи нуждаются в проверке и доказательствах.
Что касается доказательства корректности алгоритма, даже математически точное доказательство не даёт гарантии отсутствия ошибок в самом алгоритме. Ну да, алгоритм делает то что должен делать. Но то ли он делает что НАМ нужно?

0xd34df00d 22 сентября 2018 в 01:08

0

> Масло масляное, в этом примере альтернативного сложения мы используем тот же АЛУ что и оригинальный вариант.

Почему? Вы вольны использовать что угодно внутри myplus, главное — доказать, что оно ведёт себя как обычное сложение.

> Что касается доказательства корректности алгоритма, даже математически точное доказательство не даёт гарантии отсутствия ошибок в самом алгоритме. Ну да, алгоритм делает то что должен делать. Но то ли он делает что НАМ нужно?

А это вопрос совсем другого уровня. Даёт ли вам теорема то, что нужно в данной конкретной прикладной задаче?

PsyHaSTe 22 сентября 2018 в 02:59

0

На википедии есть неплохая статья, советую ознакомиться: en.wikipedia.org/wiki/Formal_verification

И это не абстрактная хрень, так баги в реальном софте находят:

> Btw, this bug was found while trying to prove soundnes of (an idealized version of) Mutex. Yay for formal methods :D

Более подробно можно почитать в блоге парня, который это нашел.

0xd34df00d 22 сентября 2018 в 05:20

+1

К слову о, я буквально сегодня набрёл на такое. Это, наверное, первый случай, когда я вижу самое что ни на есть прикладное применение гомотопической теории типов.

И, что особенно метаиронично, учитывая контекст поста, программы на языках с завтипами компилируются просто здово долго, C++ с трехэтажными темплейтами покажется вам резким как понос.

gearbox 22 сентября 2018 в 12:52

0

О, гомотопию приспособили таки к жизни :)

pzctfw 25 сентября 2018 в 16:21

0

Ещё существует довольно давно Coq Proof Assistant, идут в похожем направлении. А построением практически применимого языка на HoTT уже несколько лет занимается R&D-подразделение JetBrains:

research.jetbrains.org/ru/groups/group-for-dependent-types-and-hott
github.com/JetBrains/Arend

@gearbox, вам тоже интересно посмотреть, наверное)

0xd34df00d 25 сентября 2018 в 18:57

0

Ну, Cosette по ссылке выше как раз поверх кока.

А про JB — это круто, не слышал про эту инициативу, спасибо! *Взял на заметку.*

pzctfw 25 сентября 2018 в 20:21

0

Coq, выходит, и был первопроходцем, потому что JB-группа тоже под вдохновением от него начала ковырять гомотопическую

теорию. У них в Research есть ещё одна отдельная группа по верификации и анализу программ, там тоже интересные подходы к автоматическому поиску и устранению багов:
research.jetbrains.org/ru/groups/pvalab

Druu 22 сентября 2018 в 06:05

0

А как доказать, что ваша система, проверяющая корректность доказательства, сама корректна?

0xd34df00d 22 сентября 2018 в 06:25

0

А как доказать, что ваше доказательство теоремы ручкой на бумажке корректно?

А если чуть серьёзнее, вроде были всякие критерии, позволяющие в известном смысле извне верифицировать систему верификации. Но я эту тему пока особо не ковырял.

Ну и в правильных языках с тактиками кривая тактика не приведёт к приёму кривого порождённого ей доказательства, по идее.

Druu 22 сентября 2018 в 06:37

0

> А как доказать, что ваше доказательство теоремы ручкой на бумажке корректно?

Как? :)

> Ну и в правильных языках с тактиками кривая тактика не приведёт к приёму кривого порождённого ей доказательства, по идее.

С чего вы взяли что в компиляторе вашего языка с тактиками багов нет? :)

0xd34df00d 22 сентября 2018 в 06:42

0

> Как? :)

А я вот хз, как. Папиры публиковать, разве что, чтоб их другие читали.

> С чего вы взяли что в компиляторе вашего языка с тактиками багов нет? :)

Ни с чего, но я сужаю поверхность кода, который нужно внимательно читать и/или обмазывать глупыми тестами.

gearbox 22 сентября 2018 в 12:58

+2

Druu — оставьте этот детский сад. Абсолютного доказательства не бывает, Гедель постарался. Но формальная верификация — она не про абсолютное доказательство. Начну издалека. Возьмем например криптографию — это наука о чем? Это наука о том как большой секрет сделать маленьким. Но избавиться от секрета полностью не получается. Вернемся теперь к верификации. Это наука о том как большую веру сделать маленькой. Если вы формально верифицировали код, то на выходе вам не надо ВЕРИТЬ в его корректность, достаточно ВЕРИТЬ в то что корректны аксиомы, на базе которых строилось доказательство И верны методы доказательства. Но доказать все и вся — не будет :)

Druu 22 сентября 2018 в 15:22

+2

> Если вы формально верифицировали код, то на выходе вам не надо ВЕРИТЬ в его корректность, достаточно ВЕРИТЬ в то что корректны аксиомы, на базе которых строилось доказательство И верны методы доказательства. Но доказать все и вся — не будет :)

Ну если уж говорить о вере, то мы вообще сразу попадаем в область субъективного. Кто-то 100% покрытие тестами выдрочил и верит теперь, что все работает идеально.

gonzazoid 23 сентября 2018 в 00:53

0

Что есть аксиома если не вера?

Druu 23 сентября 2018 в 03:45

0

Аксиома — это просто гипотеза. Математика занимается поиском ответов на вопросы вида: "что, если X?", при этом истинность X (аксиомы) где-то за пределами исследования.
Так что верить в аксиомы не требуется, можно даже быть уверенным в том, что они все ложны. С математической точки зрения это ничего не меняет.

0xd34df00d 22 сентября 2018 в 17:02

0

Где-то тут потенциал для треда о философии математики и доказательства в математике чую я.

bano-notit 20 сентября 2018 в 22:43

0

Почему все к разрабам цепляются? Я вот как разраб могу сказать, что это моя работа делать сайт соизмеримый моему гонарару. Если мне интересно что-то делать или мне это тупо выгодно, то я сделаю хорошо. Если мне заплатили 5 рублей за функционал «как у фейсбука» и «ну к завтрашнему ведь готово будет», то и сделано это будет на таком говне, которое я сочту достаточно воночим.
Если же это мой проект и мне реально дороги пользователи, то я сделаю нормально за минимальные затраты для пользователя. Но это вложение души. Почему я должен вкладывать душу в какое-то левое предприятие или дело. Почему я должен тратить свою душу на каждую вебвизитку?

Cipper 21 сентября 2018 в 09:55

0

Во первых, зачем тратить свое, чужое время и душу на то, к чему душа не лежит? Дайте возможность работать тем, кто хочет заниматься визитками хотя бы для образования. И вы не будете лишнее время тратить и клиент не будет недоволен. А вы это время можете потратить на то, что вам нравится, и результат, вполне возможно, принесет вам большую прибыль. Во вторых, правильно пишется соизмеримый и гонорар, реально сложно читать просто

VolCh 22 сентября 2018 в 18:35

0

Не путайте прибыль с зарплатой/гонораром. Прибыль не гарантирована, это риски. А кушать хочется.

bano-notit 22 сентября 2018 в 19:21

0

Вот тут есть некоторая проблема. Я бы рад перейти на то, что мне больше нравится, и большую часть времени я этим и занимаюсь. Но критике последнее время подвергаются сайты-производители конктента, а это, к счастью или сожалению, не глобальные игроки, вроде вк или яндекс.новостей. Контент создают микроблоггеры и микрофорумы. И именно они пытаются из рекламы выжать максимум, вложив при этом минимум денег. И тут появляюсь я и ещё 1000 фрилансеров, которые готовы взяться за работу под нормальные деньги и нормальное тз. Конечно сразу появляются правки в тз после его заявки и иногда даже сдвиг сроков в меньшую сторону, при этом денег за это не добавляют. И что мне и ещё 1000 фрилансеров делать? Конечно говнокодить настолько, на сколько нам позволяет совесть, чтобы уложиться в сроки и бюджет.
А потом заказчик хочет прикрутить ещё пару фич, через месяц, нанимает ещё людей на сдельную, они могут заметить говнокод и комментарий «не связывайтесь» и наговнокодить ещё больше, или вообще новые фичи навернуть поверх всей старой архитектуры (что очень часто и происходит).

Новички же делают ещё больше гавнокода, даже если хотят сделать хорошо. Уровень их «нормальной работы» и моего говнокода различаются. Если бы всё, что делается сейчас «нами» делалось бы новичками…

Раз уж начали придираться к символам, то числительные пишутся через дефис.

Cipper 21 сентября 2018 в 09:29

-3

Вы меня извините, но не это ли следствие капитализма, где основной принцип — потрать меньше, продай дороже, в том числе и усилия и время и рабочие руки, ведь руки ниже поясницы ценятся дешевле, чем руки из плеч? А что касается «других стран» так и там ситуация ровно такая же. А касаемо всё же не капитализма — там ведь принцип был иной, делалось многое для удобства «пользователя», а не «разработчика и производителя» а почему? А потому что пользовались все одним и тем же продуктом, и косо глядели на того кто халтурил, даже зарабатывая много, а сейчас — деньги есть (продал родину ли, другой товар не важно какого качества, или родители обеспечили) и смотреть то косо боятся, и уважения масса, — «Надо же, с деньгами», и это вплоть до судебной системы, которая вообще исходя из простейшей логики независима в суждениях должна быть… Ну результат налицо, как говорится…

Neuromantix 21 сентября 2018 в 09:56

+8

Похоже, основы капитализма многие изучали по журналу «Крокодил». При этом распрекрасные не-капиталистические страны вначале строились на крови и костях, а потом вообще передохли. Видимо рептилоиды помогли.

senglory 22 сентября 2018 в 22:51

−1

Все страны на костях строились. У капиталистических этих костей на порядки больше за сотни лет истории было, считая хотя бы со времен огораживания в Англии.

Sychuan 23 сентября 2018 в 00:04

0

> У капиталистических этих костей на порядки больше за сотни лет истории было

Какие ваши доказательства?
Ну и одно дело 15 век и 20-ый. Во времена огораживания вроде как процветал феодализм с лордами, королями и рыцарями

senglory 23 сентября 2018 в 02:25

0

1. Вот из того, что находится на раз-два:

На вскидку, раз
На вскидку, два

В обоих случаях виновником была страна с капиталистическим укладом экономики.

2. Первая мировая. Напомню, что развязали ее страны с капиталистическим укладом экономики.



PsyHaSTe 23 сентября 2018 в 03:41

+1

> 2. Первая мировая. Напомню, что развязали ее страны с капиталистическим укладом экономики.

Исследование показало, что все дышащие воздухом люди умирали.

khim 23 сентября 2018 в 16:16

0

Ваш сарказм был бы уместен, если бы никаких других стран, кроме капиталистических, в природе не было. Но они были. Были даже «чистые» эксперименты (Вьетнам, Корея). По ним видно, что кореляции зверств со строем — не так много, но… массовое уничтожение людей **в других странах** в сочетании с благополучием в метрополии — это как раз характерная черта капитализма.

И не надо про то, что «это были далёкие колоннии, о том, что там происходит никто не знал». Ирландия рядом с Англией всегда была, никуда не отплывала…

Druu 23 сентября 2018 в 16:24

+2

> Ваш сарказм был бы уместен, если бы никаких других стран, кроме капиталистических, в природе не было.

Погодите, сколько всего существовало стран за последние 200 лет и сколько из них не были капиталистическими?

> массовое уничтожение людей в других странах

То есть если в своей — то это ок?

khim 24 сентября 2018 в 17:20

−2

> Погодите, сколько всего существовало стран за последние 200 лет и сколько из них не были капиталистическими?

Несколько сотен. Капиталистических было примерно половина.

> > массовое уничтожение людей в других странах
>
> То есть если в своей — то это ок?

С точки зрения жителей других стран? Да, разумеется.

Своих жителей гробят все страны: и капиталистические и социалистические и феодальные. А вот чужих… только давно забытый рабовладельческий строй и капитализм… забавно, кстати, что капитализм, как раз, возродил рабство (чтобы потом с пафосом с ним бороться, да)…

VolCh 24 сентября 2018 в 18:04

0

Хотите сказать, что, например СССР, не грабил ту же Германию?

vs_starosta 24 сентября 2018 в 22:05

+4

Репарации — не грабёж. Объем репараций в лучшем случае составил 4-ю часть от общих потерь СССР от войны с Германией. Кроме того, часть из этих репараций СССР выделил Польше. Ну и в 54-м отказался от репараций со стороны Восточной Германии. Ну и про совместные предприятия не стоит забывать. Сейчас это называется иностранными инвестициями — новые рабочие места и всё такое.
ИМХО, весьма гуманно обошлись еще с немцами.

VolCh 25 сентября 2018 в 09:46

−2

Были официальные репарации, а был откровенный грабёж на самых разных уровнях.

khim 25 сентября 2018 в 17:08

−2

Да, был. Но даже в немецких книгах претендующих на какую-то объективность, приводят ключевую фразу: *«Страна фашистов должна стать пустыней, как наша страна, которую они сделали пустыней. Фашисты должны быть уничтожены так же, как они убивали наших солдат.»*

Если бы до этого Германи не напала на СССР — то не было бы ни грабежей, ни убийств, ни мародёрства. А если вы считаете, что уничтожение индейев и индусов, ирландцев и африканцев — это вещи того же, плана, то вам, вне всякого сомнения, не затруднит составить списочек:
1. Когда ирландские войска высадились в Великобритании.
2. Когда африканские войска высадились на территории Бельгии.
3. Когда вооружённые индусы прибыли в Европу.
4. Когда китайцы совершили высадку в Японии.
Для начала — хватит, я думаю.

Если же вы такого списка предоставить не можете — то и заткнитесь на тему «зверств русских в Германии».

khim 25 сентября 2018 в 17:09

0

Факты грабежа, случившегося до 22 июня 1941 года — в студию, пожалуйста.

Druu 26 сентября 2018 в 13:44

−2

> Своих жителей гробят все страны: и капиталистические и социалистические и феодальные. А вот чужих… только давно забытый рабовладельческий строй и капитализм…

Вы, кажется, перепутали. Все страны грабят чужих жителей, потому что интересы своих жителей для нормального государства всегда должны ставиться выше, и если можно ограбить соседа- это следует сделать, иначе такое государство ненужно.
И только рабовладельческий строй и прочий социализм замечен в гноблении собственного народа на потсоянной основе.

Am0ralist 26 сентября 2018 в 15:01

+2

> И только рабовладельческий строй и прочий социализм замечен в гноблении собственного народа на потсоянной основе.

Посмотрел на историю… посмотрел на ваше высказывание… посмотрел на историю…
А, понял, вы с другой планеты?
Если же нет, то как в вашей истории пропали такие куски, в которых говорилось про феодальное общество и про монархии, а так же как минимум про капиталистические общества начала

прошлого века?
Удивительна избирательность...

Druu 26 сентября 2018 в 15:34

–2

> Если же нет, то как в вашей истории пропали такие куски, в которых говорилось про феодальное общество и про монархии

Феодальные общества в итоге представляли либо то же рабство, либо и не было там никакого гнобления. Монархии у нас и сейчас есть, что-то не наблюдаю гнобления британских граждан.

> а так же как минимум про капиталистические общества начала прошлого века?

Ну и кто кого гнобил-то, вы пальцем укажите?



Am0ralist 26 сентября 2018 в 16:44

0

> Феодальные общества в итоге представляли либо то же рабство, либо и не было там никакого гнобления.

Альтернативная история в треде!

> Монархии у нас и сейчас есть, что-то не наблюдаю гнобления британских граждан.

Не, конечно не было ничего в истории монархии Англии с постоянным гноблением своего населения.
Хотя где вы там хоть какую-то полноценную монархию увидели сейчас — ума не приложу.

Druu 26 сентября 2018 в 16:54

–1

> Альтернативная история в треде!

Никакой альтернативной истории.

> Хотя где вы там хоть какую-то полноценную монархию увидели сейчас — ума не приложу.

Ну вы же понимаете, что у вас теперь ненастоящий шотландец?

> Не, конечно не было ничего в истории монархии Англии с постоянным гноблением своего населения.

Ну подробнее можно?



Am0ralist 26 сентября 2018 в 17:52

+2

Подробнее? История Европы с падения Рима и до начала 20 века.
Примерно там.
А с шотландцами я ваша попытку пошутить не понял совершенно.

Druu 2 октября 2018 в 10:33

0

> А с шотландцами я ваша попытку пошутить не понял совершенно.

Какая шутка? Я указал вам на то, что вы вместо аргументации используете вполне конкретный полемический прием.

> История Европы с падения Рима и до начала 20 века. Примерно там.

Чего не история Земли с вымирания динозавров до наших дней? Конкретные примеры можно?



Am0ralist 2 октября 2018 в 10:55

0

> Чего не история Земли с вымирания динозавров до наших дней? Конкретные примеры можно?

Демагогия с попыткой подменить тезис и увести в сторону.
Феодальные и монархические государтсва в европе были как раз в указанный период. Конкретные примеры — «история Европы с падения Рима и до начала 20 века»

> Какая шутка? Я указал вам на то, что вы вместо аргументации используете вполне конкретный полемический прием.

Сейчас я перейду на полемические приемы: о великий последователь тайных знаний рептилоидов, открой же завесы тайны, что имел ты ввиду про шотландцев и где в Великобритании сейчас — полноценная монархия?

Druu 2 октября 2018 в 11:10

0

> Феодальные и монархические государтсва в европе были как раз в указанный период.

В указанный мной период они тоже были.
Я вас не спрашивал, в каком периоде были примеры. Я вас попросил указать эти примеры. Не период. Сами примеры. Вы понимаете разницу?

> что имел ты ввиду про шотландцев

https://ru.wikipedia.org/wiki/%D0%9D%D0%B8_%D0%BE%D0%B4%D0%

Применяя к нашему случаю — какую бы монархию я вам в пример не привел, вы просто объявите, что это "ненастоящая" монархия (ненастоящий шотландец), а потому мой пример нерелевантен.
Так что в том виде, в котором вы это формулируете — это не аргумент, это полемический прием. Это вы должны пояснить, почему по-вашему Великобритания монархией не является, и что надо, чтобы государство являлось таковой. Ведь, согласно общепринятым критериям, Великобритания — монархия.



Am0ralist 2 октября 2018 в 12:03

0

А, все понятно. Слив в полную демагогию.

> В указанный мной период они тоже были.

ну вот и доказывайте, что при динозаврах она была. Ну или в во дни с первого по седьмой создания Земли Богом.

> Применяя к нашему случаю — какую бы монархию я вам в пример не привел, вы просто объявите, что это «ненастоящая» монархия (ненастоящий шотландец)

Ага, аргумент к личности. За меня решили что я буду считать, а что нет, после чего героически побеждаете ваше мнение обо мне.
Хотя, то как вы смело требуете указать вам периоды и конкретные вещи — вы явно страдаете переносом своего поведения и мышления на окружающих, после чего смело их обвиняете в своем же поведении.

Вы победили и смело можете называть себя королем троллей.
Пока-пока.

Druu 2 октября 2018 в 12:51

+1

> ну вот и доказывайте, что при динозаврах она была.

Если что-то было в период X то оно было в любой период, содержащий X. Указанный мной период содержит ваш, ч. т. д.

Но это все не важно, еще раз. Я не задавал вам никаких вопросов про период. Я просил вас привести конкретные примеры. Не период, в котором были примеры, а примеры. Непонятно, что тут сложного для понимания.

> Ага, аргумент к личности. За меня решили что я буду считать, а что нет

Я ничего не решал о том, что вы БУДЕТЕ делать, я констатировал факт того, что вы УЖЕ сделали — применили полемический прием "ненастоящий шотландец" вместо аргументации.
Еще раз — УЖЕ применили, по факту.
Что вы там дальше будете делать — мне искренне наплевать.

BigBeaver 5 октября 2018 в 08:37

0

> Ага, аргумент к личности.

Это не аргумент к личности. Аргумент к личности это «ты не прав потому, что лысый».

0xd34df00d 29 сентября 2018 в 02:15

0

> Монархии у нас и сейчас есть, что-то не наблюдаю гнобления британских граждан.

Ножи там запретили уже или таки здравый смысл подебил?

Sychuan 23 сентября 2018 в 16:44

+1

> Вот из того, что находится на раз-два:
>
> Меня интересовало утверждение
>
> > У капиталистических этих костей на порядки больше за сотни лет истории было
>
> А не ссылка на голод в Бенгалии. И потом сравнить в относительных цифрах, чтобы было ясно видно. Коммунистические режимы убили 5% человек от общего числа проживавших на их территориях и территориях их сателитов, капиталистические страны убили 35% от от общего числа проживавших на их территориях и территориях их сателитов. Тогда все будет замечательно. Ну и опять таки хотелось бы рассматривать 20-ый век, когда уже можно точно определить, какие страны капиталистические, а какие нет. И посторонние факторы вроде феодальных завоеваний отсутствуют.

Fracta1L 21 сентября 2018 в 10:55

+8

> А касаемо всё же не капитализма — там ведь принцип был иной, делалось многое для удобства «пользователя»

Што, простите? Это в СССР многое делалось для удобства пользователя? А примеры можно увидеть? А то я сколько ни встречал советских изделий, начиная от мелких штучек типа посуды, и заканчивая заводами и жилыми домами — всё сделано через ЖОПЕНЬ, за очень редким исключением.

Cipper 21 сентября 2018 в 11:04

–5

Вы как то не учитываете что вы судите мерой человека видевшего гораздо больше чем обыденный рабочий в СССР, жившего в свое время и в изолированной стране, но даже сейчас многие наработки используются в других странах и очень успешно, я уж промолчу что последние лет 30-35 страна не развивается вообще, если не больше. Смотрите шире и всё увидите

Fracta1L 21 сентября 2018 в 11:35

+8

Т.е. создаваемое в СССР кажется удобством только тому, кто ничего лучше не видел из-за железного занавеса? Интересное определение удобства, очень такое своеобразное XD

> последние лет 30-35 страна не развивается вообще

Если вы под развитием подразумеваете шагание строем в ногу в сторону очередной «великой цели», то да, не развивается, и нафиг бы оно упало, такое «развитие».

Mabuslus 21 сентября 2018 в 16:51

0

Посмотрите советский сериал «Фитиль». Там был сюжет как раз по этой теме. Инженер предложил начать выпускать новую модель пылесоса(показывают типичный современный «капиталистический» пелесос), но чиновники начинают доказывать инженеру, что они умнее него и вносят кучу правок в проект. И в конце концов получается у них типичный советский пылесос больше похожий на трактор.



Sychuan 21 сентября 2018 в 11:49

+1

А как же самое вкусное мороженное! (сарказм)

vin2809 21 сентября 2018 в 15:57

–1

Кстати, действительно самое вкусное. И колбаса была из мяса… (Это не сарказм).



PsyHaSTe 21 сентября 2018 в 16:01

+1

Но не для всех.
А еще есть подозрение, что вкуса придавал факт стояния в многочасовой очереди, известно ведь, что чем труднее достается что-то, тем больше ценится. В «Контрольной Закупке» помню чуть ли не на МРТ проверили, как человек пьющий одно и то же вино но из дешевой бутылки и из дорогой по разному его описывал. И не потому, что он понтовался, а на МРТ было видно, что при испитии напитка про которой была априорная информация, что он дороже, просто тупо больше отделов мозга активировалось.

Simplevolk 21 сентября 2018 в 09:45

0

> Просто производство хламья очень просто, дешево и удобно

, вот при этом и капитализм: если это сделать просто, дешево и быстро, то значит и прибыль получишь быстрее.

Ясное дело, что в утопиях Стругацких или Ефремова, производители бы думали об улучшении качества вещей, о том чтобы производить вещи развивающие человека и тд.

Neuromantix 21 сентября 2018 в 10:01

+7

Скорость получения прибыли важна не для капиталистов, а для местного бизнеса, Пелевиным описанного в Генерэйшн Пи. Отсюда же и все разговоры вокруг «убыточности» предприятий Маска, Безоса и тп. Это только в воспаленном мозгу российского псевдо-бизнеса может быть идея «хапать, пока не замели, не развалилось и водка не кончилась».

vs_starosta 22 сентября 2018 в 12:57

–2

А прекрасному эльфу капиталисту нужна прибыль не здесь и сейчас? Или биржевые роботы создаются с целью эстетического удовольствия? Или после каждой презентации родстеров в космосе не меняется цена акций?

Neuromantix 22 сентября 2018 в 13:34

+2

Разница в том, что в нормальных условиях капиталист может себе позволить вложить больше денег в производство, создание инфраструктуры и тп, и в долгосроке получить прибыли намного больше, чем он получит, «нахапав» здесь и сейчас, пока очередной политик не спятил, пока валюта не рухнула, пока еще что-то в «стабильности» не произошло.

senglory 22 сентября 2018 в 22:57

0

> Разница в том, что в нормальных условиях капиталист может себе позволить вложить больше денег в производство, создание инфраструктуры и тп, и в долгосроке получить прибыли намного больше

Вот что пишут люди из самого сердца мира наживы и чистогана

> На бизнес-встрече представителей венчурного капитала с «господами учеными» инвесторы, то есть те, кто реально финансирует прикладные исследования, на пальцах объясняли некоторые прописные истины. К примеру, одним из первых тезисов был таков: We do not need cure, we need treatment.

Это как, «нормальные условия»? Или капитализм какой-то неправильный, цветов радужных?



Sychuan 23 сентября 2018 в 00:17

+1

Какой-то чувак в Ж.Ж. написал про каких-то неизвестных чуваков где-то неизвестно где. О да. После такого могучего аргумента пора всем обратится в коммунизм. Ну а в реальном мире, лекарство от гепатита С было наконец изобретено пару лет назад. Хотя казалось бы, ситуация до этого фармкомпании отлично устраивала. Я уж не говорю о генетических лекарствах, которые сейчас горячая тема. Вот уж где можно было бы собирать деньги за тритмент десятки лет

senglory 23 сентября 2018 в 01:27

+2

> Какой-то чувак в Ж.Ж.

«Какой-то чувак в Ж.Ж.» уже 22 года живет и работает в Университете Нью-Йорка биохимиком и профессором. Зовут его Виталий Эпштейн.

Про коммунизм никто и не говорил, кроме Вас. Приписывать мне то, чего нет в моих словах точно не стоит. А вот что стоило бы (мечты, мечты) — так это иметь инструмент давления на всех этих «инвесторов» чтобы, например, разработке новых антибиотиков они бы уделили финансирование в таком же объеме, что они тратят на цитостатики. Пока же большинство крупных фармацевтических компаний забросили свои программы разработки антибиотиков к 1995 годую

> лекарство от гепатита С было наконец изобретено пару лет назад.

Не спорю, что находятся и среди руководителей фармы вменяемые люди, понимающие, что HCV безразличен твой размер счета. Но большинство

напоминает азартных игроков, которым главное отчетность перед акционерами показать хорошую, да бонус свой годовой получить за продажи.

> Я уж не говорю о генетических лекарствах, которые сейчас горячая тема.

… и с эффективностью работы которых пока что как с УТС — вроде движемся, но сильно медленнее, чем нужно. А вот лидара и понтов — выше Эвереста.

0xd34df00d 23 сентября 2018 в 22:01

+5

> А вот что стоило бы (мечты, мечты) — так это иметь инструмент давления на всех этих «инвесторов» чтобы, например, разработке новых антибиотиков они бы уделили финансирование в таком же объеме, что они тратят на цитостатики.

Больше регулирования богу регулирования. Вот бы ещё отрегулировать, чтобы лекарства сами на деревьях росли, тогда заживём.

Поразительное желание залезать в чужой карман. Давайте я в ваш залезу, но глазиком, не ручечкой: сколько денег вы лично задонейтили или хотя бы инвестировали в исследования разработки новых антибиотиков?

senglory 24 сентября 2018 в 23:55

0

<blockquoteПоразительное желание залезать в чужой карман.>

Поразительное желание по холуйски заботиться о тех, кто в Вашем кармане поселился и с него кормится, а потом еще и прилетает в Конгресс с воплями про too big to fail и добавкой из налогов.

0xd34df00d 25 сентября 2018 в 01:40

+2

Я забочусь только о формальной логике.

Вы так говорите, будто отрицание необходимости (и допустимости) регулирования деятельности инвесторов мешает мне негативно относиться к кормёжке с налогов и too big to fail (хотя я так и не понял, причём это к инвесторам).

0xd34df00d 23 сентября 2018 в 21:59

0

> Это как, «нормальные условия»? Или капитализм какой-то неправильный, цветов радужных?

Если знать английский немножко за пределами ландан из зе кэпитал оф грейт британ, то становятся ясны некоторые коннотации. Cure — это полное исцеление, treatment — в том числе, достаточное поддержание качества жизни. Ничего о стоимости тут нет. Более того, стоимость этого cure может быть 100500 денег за раз, а treatment — 100 за 1000 необходимых доз. И не очень понятно, почему второе безусловно хуже первого. А, не безусловно? Ну, значит, додумывание, как всегда.

Придумывают все эти лекарства в основном почему-то кляты капиталисты.

senglory 23 сентября 2018 в 23:49

+1

> И не очень понятно, почему второе безусловно хуже первого

Что тут непонятного? Вылечивание — это точка в истории болезни, а tretament — это деньги, вытягиваемые зачастую пожизненно. При таком раскладе тот, кто поставляет препараты, становится просто на сторону заболевания, а не пациента. Что неясно-то?

0xd34df00d 24 сентября 2018 в 00:12

0

> Вылечивание — это точка в истории болезни, а tretament — это деньги, вытягиваемые зачастую пожизненно.

Я привёл пример, когда исцеление — это, конечно, точка, но нереализуемая.

Если у меня будет выбор, платить за лечение пипецомы 100 денег в месяц или 100500 единоразово, я скорее выберу первое, потому что 100500 единоразово у меня нет, а кредит людям на смертельные пипецомы выдавать — довольно рисковано.

И, кстати, жить я буду вряд ли больше 83 лет, так что это даже финансово выгодно.

> При таком раскладе тот, кто поставляет препараты, становится просто на сторону заболевания, а не пациента.

Если он неоправданно затягивает лечение, то появляется другой фармацевт с более эффективным лечением и выталкивает первого с рынка.

Alexeyslav 24 сентября 2018 в 10:40

−1

А потом остальные фармацевты просто топят выскочку и всё возвращается на свои круги. Всё это хорошо когда реально работают рыночные отношения, но реальность она несколько иная — монополии, сговоры и т.д.

0xd34df00d 24 сентября 2018 в 16:48

0

Чем топят?

Alexeyslav 24 сентября 2018 в 17:07

0

В основном по коррупционным схемам.

0xd34df00d 24 сентября 2018 в 17:09

+1

То есть, через государство? ;)

Alexeyslav 24 сентября 2018 в 17:18

−1

Не факт, могут и через мафиозные структуры.

third112 25 сентября 2018 в 15:13

0

> Причем тут капитализм?

При том ИМХО, что у капитализма есть свои плюсы и минусы. Минус, о котором речь — это потребительская культура — покупать как можно больше и чаще. Я про это писал (вслед за другими):

> Казалось бы, ко всем этим выводам можно придти, и не прибегая к примерам типа ранних версий Windows, а используя только здравый смысл. Однако так называемая культура общества потребления вносит заметный деструктивный вклад в сознание людей.

vadimzz 25 сентября 2018 в 17:29

0

Капитализм тут очень даже причем. Постоянная гонка по времени за конкурентами, бешеные бюджеты на эту самую конкуренцию, и самое главное — высокая фрагментированность всего, включая платформы или сами конкурирующие продукты (кому нужны эти 16.000 андроид устройств и вообще 10 разных мобильных ОС?). Ни у кого нет времени остановиться. Создание программы качественно не происходит, потому что любой продукт изначально создается в условиях риска провала. Есть прекрасные исключения — когда нечто происходит вопреки этому, но это не является системой, это именно исключения. Не все становятся Торвальдсами или Джобсами, способными в чем-то изменить мир.

В большинстве мы не стремимся к «прекрасному завтра», мы стремимся не провалиться прямо сейчас.

Neuromantix 25 сентября 2018 в 17:49

0

Ну Джобс прекрасно конкурировал одним смартом и вылизаным софтом. Но это было давно, Джобс помер, и все свернуло в ту же самую колею, по которой несется вся куча. Может, дело не в капитализме, а в том, что большинство производителей не способны выкарабкаться из колеи, в которую их так и тянет.

vadimzz 25 сентября 2018 в 18:03

0

Проблема куда как более фундаментальна. Если бы одновременно 20 Стивов начали делать бы одно и тоже, то либо большинство бы провалилось с финансами, либо провалились бы все. Потому что нельзя просто так взять, и сделать продукт в 10 раз дороже и в 2 раза качественнее, чтобы никто этого не заметил :)



PsyHaSTe 25 сентября 2018 в 18:12

0

Почему же в Союзе параллельно куча КБ занималась одним и тем же?

> Постоянная гонка по времени за конкурентами, бешенные бюджеты на эту самую конкуренцию, и самое главное — высокая фрагментированность всего, включая платформы или сами конкурирующие продукты

Конкуренкция — это хорошо

> любой продукт изначально создается в условиях риска провала

То ли дело работать, когда знаешь, какое Г не произведи, его обязаны у тебя закупить.



vadimzz 25 сентября 2018 в 18:17

–1

Да, но с КБ это был управляемый процесс. Эта куча КБ не тратила деньги на борьбу друг с другом, и уж точно не стремились слепить Г поскорее конкурентов и сплавить.

> То ли дело работать, когда знаешь, какое Г не произведи, его обязаны у тебя закупить.

Че-то при капитализме оно именно так и происходит. «Невидимая рука рынка» творит чудеса. Телефоны на полгода и еда из пенопласта в тренде.



PsyHaSTe 25 сентября 2018 в 19:52

0

> Да, но с КБ это был управляемый процесс. Эта куча КБ не тратила деньги на борьбу друг с другом, и уж точно не стремились слепить Г поскорее конкурентов и сплавить.

А чем же они занимались?

> Че-то при капитализме оно именно так и происходит. «Невидимая рука рынка» творит чудеса. Телефоны на полгода и еда из пенопласта в тренде.

Нормальные телефоны и нормальная еда, не знаю, что там у вас происходит.

К слову, у нас есть исторический период двух разных мироустройств с одинаковыми исходными параметрами. И я прекрасно представляю, с какой завистью относились к людям с не то, что заграничными вещами, а хотя бы экспортными вариантами отечественных.



vadimzz 25 сентября 2018 в 19:59

–1

> А чем же они занимались?

Не поверите — работали над продуктом.

> исторический период двух разных мироустройств

Очень не хотелось бы переводить разговор в политическую плоскость, но если справедливо делить тот мир на 2 части, то в «ненашу» часть попадало куда больше стран, чем только страны западного мира. И средний уровень жизни тут и там будет выглядеть несколько иначе. И насколько жители какой-нибудь Либерии и прочих Гондурасов могли себе позволить эти заграничные вещи, вызывавшие зависть советского потребителя — большой вопрос.



PsyHaSTe 25 сентября 2018 в 21:46

0

> Не поверите — работали над продуктом.

Так конкуренция заключает в том, чтобы сделать продукт лучше! В чем же тогда разница?

> Очень не хотелось бы переводить разговор в политическую плоскость, но если справедливо делить тот мир на 2 части, то в «ненашу» часть попадало куда больше стран, чем только страны западного мира. И средний уровень жизни тут и там будет выглядеть несколько иначе. И насколько жители какой-нибудь Либерии и прочих Гондурасов могли себе позволить эти заграничные вещи, вызывавшие зависть советского потребителя — большой вопрос.

Ну, в данном случае я про два «полюса» рассуждаю (раз уж про них зашла речь), ведь даже я считаю, что второй «полюс» не настолько плох был, чтобы сразу уж с гондурасом сравнивать.

VolCh 26 сентября 2018 в 10:26

+1

> Эта куча КБ не тратила деньги на борьбу друг с другом, и уж точно не стремились слепить Г поскорее конкурентов и сплавить.

А вы изучите вопрос получше, мемуары почитайте что ли. Да даже официальные отчёты о том как те же автоматы принимали на вооружение: реально конкурсы между КБ проводились, из нескольких разработок выбирали одну. Официально лучшую по совокупности характеристик, по мемуарам и прочим свидетельствам что только в дело не шло, чтобы убедить членов госкомиссий. Особенно когда дело касалось выбора между изделиями от КБ, которые «крышевали» разные властные кланы.



Tantacula 20 сентября 2018 в 22:19

+4

Глупо списывать на капитализм все проблемы общества. То, что вы перечислили — лишь отражение самого процесса эволюции. Желание человека стать более успешным (более конкурентным в сравнении с другими представителями своего вида) было задолго до капитализма (еще когда он был обезьяной), только раньше вам пришлось бы постоянно драться с альфой и бетами за ресурсы и возможность оставить потомство, а теперь можно ограничиться хорошо оплачиваемой работой.



Am0ralist 20 сентября 2018 в 22:53

+2

> . Капитализм вырашивает (нытик-фильм на 3 часа!) невежд.

Извиняюсь, но фильм начался с фразы про «30 лет реформирования» и меня уже с нее бомбит, если честно. Ну потому что бред…
Как человек, который по семейным, дружеским и жизненным причинам был долгое время связан с оным.

Xandrmoro 20 сентября 2018 в 21:47

0

Вот только рисовать красивые пиксели можно без того, чтобы это всё безбожно тормозило и заставляло мой восемнадцатого года ноутбук изображать из себя вертолёт.

khim 20 сентября 2018 в 21:54

0

Можно — но тогда вы этих красивых пикселей нарисуете меньше и, соответственно, заработаете меньше денег.

yarric 20 сентября 2018 в 23:05

+2

Игры с рисованием красивых пикселей нормально справлялись ещё в 90х — на компах, которые сейчас даже свежую винду не потянут (интерфейс которой выглядит примитивнее, чем у старой винды). Фейл по всем фронтам: даже красивые пиксели рисовать ниасилили, на четырёхядерных процах на 2 ГГц.

khim 21 сентября 2018 в 00:20

+4

Те игры, о которых вы говорите разрабатывались месяцами, иногда — годами. А современные тяп-ляп сайты создаются на дни, иногда за часы…

И если люди отказываются пользоваться интерфейсом 90х, а требуют продуктов в стиле Windows 10 (или в «материальном дизайне»), то, стало быть, цель достигнута.

Что там сказал бы Роден на тему этих интерфейсов — неважно…

SerafimArts 21 сентября 2018 в 00:39

+2

> И если люди отказываются пользоваться интерфейсом 90х, а требуют продуктов в стиле Windows 10 (или в «материальном дизайне»), то, стало быть, цель достигнута.

Начнём с того, что использовать Windows 10 пользователей просто *вынуждают*. Лично я на свой компьютер 7ку или более ранние версии просто физически не могу поставить. Начиная с того, что там не работает Docker, заканчивая тем, что то ли процессор (i7 7600k), то ли оперативка (DDR4) несовместима и из-за этого вылетает синий экран при установке. Интерфейс у 10ки неприятный и отталкивающий, а в более ранних версиях вообще отвратный. Глаза кровоточили, когда видел иконки не по гайдлаинам (в свой время занимался вёрсткой и получал люлей от дизайнера, если где-то какой-то пиксель вылезал).



Neikist 21 сентября 2018 в 07:26

+2

Хм, а мне десятка нравится. Управление окнами удобное, иконки (да и вообще дизайн), по моим ощущениям поживее шевелится на удивление.

popov654 21 сентября 2018 в 18:22

+3

> Управление окнами удобное

Чем оно хуже в семёрке?)

> по моим ощущениям поживее шевелится на удивление

А по моим дико тормозит. По крайней мере в виртуальной машине. Версия 1609 ещё кое-как ворочалась, после апдейта до 1803 — это просто слайдшоу с загрузкой по 7-8 минут.

Am0ralist 21 сентября 2018 в 22:55

0

> Чем оно хуже в семёрке?)

больше вариантов, что с окном приложения можно сделать с помощью шорткатов (посмотрите, как они меняются от версии к версии).

PsyHaSTe 22 сентября 2018 в 03:04

0

Один из крупнейших плюсов — если приложение вдруг сдохло (такое бывает), игра например, заташив с собой весь UI, то в семерке была старая проблема, что даже если открыть диспетчер устройств, то игра рисуется поверх и её не прибить. Раньше приходилось через ctrl+alt+delete попадать на страницу пользователя и делать logout. Хоть не полная перезагрузка, так сказать. В десятке сдохшее приложение можно перенести на другой виртуальный рабочий стол, спокойно открыть диспетчер и прибивать надоедливую фигню. Очень удобно и не раз пригождалось.

vis_inet 22 сентября 2018 в 21:34

+1

Кажется, что это достаточно нечастая потребность

PsyHaSTe 22 сентября 2018 в 23:08

0

Ну раз в месяц где-то стабильно появляется.

vis_inet 23 сентября 2018 в 15:32

0

Нечасто...
А в целом насколько удобнее чем Win7?

PsyHaSTe 23 сентября 2018 в 16:43

–1

Не лично мне по мелочам кажется удобнее. Пуск тот же сильно удобнее, перед глазами спокойно может поместиться 10x10 иконок, то есть 100 программ в доступности одного клика. 100 же элементов выпадающего меню в семерке выглядят не очень.

Больше софта ставится из коробки. Сюда же можно отнести, что после установки винда сразу готова. Не надо ставить 350 обновлений, которые вышли за все эти годы. Да, есть репаки, но я им не доверяю.

Ну а в целом это та же семерка, только с небольшими бонусами вроде перечисленных + напирмер красивый слайд-шоу скринсейвера. То есть все ровно то же самое, глобально внешне ничего не поменяли.

sumanai 23 сентября 2018 в 17:12

+1

> Больше софта ставится из коробки.

Что может быть и минусом. Куча неиспользуемого мусора, который не удаляется стандартными средствами.

PsyHaSTe 23 сентября 2018 в 21:35

0

Под софтом я имею ввиду драйверы. Всякие принтеры/вайфаи/мышки/... определяются намного бодрее. WSL опять же.

sumanai 23 сентября 2018 в 23:15

0

Нестандартная трактовка. Так то да, я не против драйверов. Хотя как по мне, лучше иметь вариант их загрузки из сети.
Но всё таки именно ненужных программ в десятке вагон.

Am0ralist 24 сентября 2018 в 12:02

0

Ну вот есть нюк, в котором все изменения — два типа SSD дисков в M2 порту, кроме обычного сата разъема, разъем под вафлю и исключительно перефирия по юсб и hdmi. Но при этом в ос хранится полный пак всех дров под всякие видеокарты и прочие вещи, которые туда физически не воткнуть, например...

PsyHaSTe 24 сентября 2018 в 12:20

0

Это естественное следствие стандартизации.

Например, у кулера в вашем офисе есть только режим «налей холодную» и «налей горячую». Чтобы налить воду комнатной температуры можно было бы просто сделать кран, чтобы подавать воду напрямую из бака, но вместо этого предлагается смешивать холодную и горячую, хотя это и не оптимально.

Am0ralist 24 сентября 2018 в 13:03

0

Я про то, что на устройство, в которое физически не возможно поставить всякие платы расширения — как бы логично не кидать все дрова на всякий случай.

PsyHaSTe 24 сентября 2018 в 13:20

0

Ну пользователи как раз этого хотят.
Когда в убунте мне нужно компилить из исходников драйвер на вай-фай, потому что разработчики сочли это «очередной платой расширения», это очень неудобно и раздражает.

Am0ralist 24 сентября 2018 в 13:59

0

Хм, ещё раз. Есть немаленький сегмент оборудования, в котором просто физически не возможно подключить некоторые типы оборудования от слова ВООБЩЕ.
Нюки, планшеты, ноутбуки. Но вот мне как-то сомнительно, что ОС автоматически определяют оные и выкидывают всю ту охапку дров, которая у них хранится на случай, если юзер решит воткнуть вот эту ISA плату в этот комп.
Да, конечно, остаются случаи, когда жесткий диск переключают в другой комп. Можете сами оценить, насколько часто такое бывает (и да, все равно в этом случае половину дров надо качать заново).

Alexeyslav 24 сентября 2018 в 14:04

0

С программной точки зрения, ISA шина у вас всё ещё есть, и пофиг что она где-то там в чипе смотрит своими контактами «вникуда». В следущей ревизии внезапно решат что она нужна и выведут наружу.

PsyHaSTe 24 сентября 2018 в 15:02

0

Виндс — ОС общего назначения, которую могут запускать на произвольном оборудовании.
Внезавно если у вас есть узкий кейс, вы можете самостоятельно пропатчить установочный образ, вырезав лишнее. Поэтому принципу небезызвестные ZVER EDITION были построены.

Или претензия к МС, что они умолчания используют для общего сегмента ПК, а не для сегмента оборудования с фиксированным набором компонентов? И не выпускают по одишну на каждую комбинацию этих самых устройств?

Alexeyslav 24 сентября 2018 в 14:01

0

И потом ты подключаешь к устройству физический расширитель шины и внезапно оказывается возможным подключить к девайсу все эти устройства.
Иначе что придётся сделать? Надо будет пересмотреть всё разнообразие девайсов, сопоставить к ним драйвера которые есть в системе и удалить их сделав уникальную сборку. А потом… при каждом апдейте проверять что в этой сборке ничего не поломалось, что хватает всех файлов, все имеющиеся драйвера не конфликтуют друг с другом. Кто это всё делать будет и за чей счет? Дешевле взять уже существующее, запускается? запускается! так чего ещё надобно? Ну подумаешь с флешки 2-3гб меньше стало, кого это волнует — если вам из 16гб не хватает 2-3гб то это лишь означает что железка была выбрана сильно в притык и дешевле удвоить память чем ковырять свою сборку и поддерживать её в дальнейшем.

PsyHaSTe 24 сентября 2018 в 17:32

0

> И потом ты подключаешь к устройству физический расширитель шины и внезапно оказывается возможным подключить к девайсу все эти устройства.
> Иначе что придётся сделать? Надо будет пересмотреть всё разнообразие девайсов, сопоставить к ним драйвера которые есть в системе и удалить их сделав уникальную сборку. А потом… при каждом апдейте проверять что в этой сборке ничего не поломалось, что хватает всех файлов, все имеющиеся драйвера не конфликтуют друг с другом. Кто это всё делать будет и за чей счет?

Это аргумент, почему МС не должна этим заниматься.

Alexeyslav 25 сентября 2018 в 09:13

0

Так она и не занимается — просто поставляет все доступные драйвера скопом.
Вот совсем недавно наткнулся на такой мост — USB3->PCI-E не было шины, и тут вдруг она появилась. Неждачик для операшоники.

Am0ralist 25 сентября 2018 в 10:51

0

а можете кинуть ссылкой и в двух словах сказать, для чего она реально подходит?)

Alexeyslav 25 сентября 2018 в 12:47

0

Для подключения видеокарты, например в устройствах где нет для этого даже разъёма.
Видел где-то использовали для организации фермы для майнинга. Ограничения есть конечно, по скорости и отзывчивости но всё что можно подключить к PCI-E и не требовательное к производительности интерфейса можно использовать.
voron.ua/catalog/034118

Am0ralist 25 сентября 2018 в 13:10

0

Ээээ… По фотке обычный рейзер с кабелем USB3.0 — ибо по ним сигнал хорошо гонять. Там даже на 1 фотке обычный PCI Express 1X разъем с другой стороны (а есть еще под mini-PCI-e ноутным). То есть шина уже есть в устройстве, а в нее просто позволяют воткнуть PCI Express 16X карту, вместо 1X…

Am0ralist 24 сентября 2018 в 17:58

0

> И потом ты подключаешь к устройству физический расширитель шины и внезапно оказывается возможным подключить к девайсу все эти устройства.

Да, именно. Ты подключил — ты и дрова притащил. По сути в винде огромный набор дров для совместимости с оборудованием, которое ты в нее никогда и не воткнешь. И так же по факту, хорошо если половина таких втыкаемых железок (я не учитываю флешки, естественно) всё равно не требовала свои дрова ставить. Как и то, что обновление дров МСом на ноутах постоянно ломало что-либо (ибо у производителя дрова были патченыные, а винда накатила новее и от производителя чипов или последовательность накатывания их нарушила, так что дрова нвидии переписали какой-то файл от дров видюхи интела и у ноута перестала управляться подсветка экраном).
Вот и нафига так делать?

Kwisatz 23 сентября 2018 в 19:41

0

А вы видели как на ходу винда вытягивает обновления для драйверов видеокарты и в процессе меняет два монитора местами? Один раз я выцарапывал уи с телевизора, вторая раз эта гадина вообще не смогла отрисовать уи изза частоты в 120гц (о как это весело было).

А как вам невозможность уменьшить громкость не дефолтного звукового устройства?

Или в процессе очень важной работы вам понадобилось перезагрузить машину, а винда задумалась на час.

PsyHaSTe 23 сентября 2018 в 21:38

0

> А вы видели как на ходу винда вытягивает обновления для драйверов видеокарты и в процессе меняет два монитора местами? Один раз я выцарапывал уи с телевизора, вторая раз эта гадина вообще не смогла отрисовать уи изза частоты в 120гц (о как это весело было).

Не видел. На двух разных компах пользуюсь 2 мониторами (в одном случае — вообще ноут, где родной монитор выключен). Всё, что замечал — после установки через пару минут винда может мигнуть экраном, доустанавливая драйвер. Хотя верю, что такое может произойти.

> А как вам невозможность уменьшить громкость не дефолтного звукового устройства?

Хм, не знаю, я обычно уменьшаю громкость текущего. То есть включил наушники — регулирую наушники, зачем мне что-то другое? Через 2 устройства одновременно слушать как-то странно, мне кажется.

> Или в процессе очень важной работы вам понадобилось перезагрузить машину, а винда задумалась на час.

Видел много сешных роликов в интернете на эту тему, но у меня почему-то ни разу не было. Ставлю себе активные часы с 10 до 20 и всё, в это время точно ничего не перезагрузится.

Kwisatz 23 сентября 2018 в 21:49

0

> Через 2 устройства одновременно слушать как-то странно, мне кажется.

У меня телек вторым.

> в это время точно ничего не перезагрузится.

А я и не говорю что само, но будто вам никогда не требуется)

PsyHaSTe 23 сентября 2018 в 21:54

0

Говорю, на видео с конференций где у докладчика винда решила обновы поставить я видел, действительно бред. Но меня бог миловал. Мне трудно сказать, надуманная ли проблема или просто мне повезло.

Ogra 24 сентября 2018 в 05:09

0

> Через 2 устройства одновременно слушать как-то странно, мне кажется.

Телевизор для одного члена семьи, наушники с монитором для другого. У нас дома каждый вечер так.

PsyHaSTe 24 сентября 2018 в 10:40

0

Ну, видимо у меня небольшой уголок рая, ибо у каждого по своему девайсу с отдельным монитором. Телевизора же давно нет, т.к. смотреть нечего, а консоли я после сеги не покупал.

Понял ваш сценарий. Видимо, действительно неудобно.

Ogra 24 сентября 2018 в 11:11

0

Сериалы не смотрите, на Ютуб не заходите? Детей с мультиками дома нет? Байонетта, Лара Крофт, Мортал Комбат — все есть на ПК, консоли покупать необязательно.



PsyHaSTe 24 сентября 2018 в 11:51

0

Детей нет. Для сериалов у меня есть 27 4к монитор. Мортал комбат тоже на компе играю, джойстики он поддерживает.

DMGarikk 24 сентября 2018 в 14:17

0

Тут не полноценная вина винды
у меня была более веселая ситуация:
Включаю комп, биос, логотип винды, окно логина… бац монитор в standby
в безопасном режиме все работает… причем ничего не помогает, не снос дров (точнее удаление устройств из диспетчера) ничего
кое как обходными путями ставлю radmin… гружусь… после загрузки винды висит окно «Обнаружено новое устройство: Монитор SVGA, продолжить?» жму «да!»… монитор включается… пипец.
а косяк был в драйверах на видеокарту ATI которые обновились и они решили что если монитора нет в системе то и видеосигнал выдавать на VGA не стоит

Am0ralist 24 сентября 2018 в 14:36

0

У меня было веселее — с чего-то перестал работать монитор по DVI. Переключил на HDMI (причем тоже там хитро — если при включении компа ничего не было вставлено в этот разъем, то он отключался и до следующего ресета).
Так пару лет проработало. Пришлось переустановить винду и… монитор через DVI заработал, но вот через HDMI — перестал.
Старая 8800gs. Копать более старые варианты дров и перепроверять все не стал, ибо времени требовалось больно много



Mingun 23 сентября 2018 в 09:31

0

В таких случаях (если игра закрывает весь экран и диспетчер задач не виден) помогает его открытие не по комбинации Ctrl+Shift+Esc, а Ctrl+Alt+Delete -> «Запустить диспетчер задач». Это если у него в параметрах не стоит «Поверх всех окон». Если же стоит, проблем обычно не бывает.

PsyHaSTe 23 сентября 2018 в 09:42

0

Все равно игра рисуется поверх. После ctrl+alt+delete действительно появляется окно логина с выбором вариантов, но после тыка на «диспетчер задач» он открывается под игрой. Это если галка не стоит. Если стоит — то проблем нет, но она в 99 случаях из 100 просто бесит.

sumanai 23 сентября 2018 в 11:16

+1

Теоретически должно помочь Alt → ↓ Enter.

Mingun 23 сентября 2018 в 11:44

0

А Alt-Табом после этого переключиться можно? Просто иногда он может открываться ниже игры, но он *открывается* и на него можно переключиться, в отличие от варианта с простым Ctrl+Shift+Esc, когда комбинация (судя по всему) даже до системы не доходит.

sumanai 23 сентября 2018 в 12:57

0

ctrl+alt+delete ничем нельзя перехватить, в отличии от других комбинаций.



Am0ralist 21 сентября 2018 в 10:33

+4

> Начнём с того, что использовать Windows 10 пользователей просто вынуждают.
> <...>
> Начиная с того, что там не работает Docker,

И в этом тоже MS виноваты, которые винду вам десятую навязывают?



san-smith 21 сентября 2018 в 13:04

0

Справедливости ради — некоторое давление всё же присутствует. Например, отказ от поддержки старых версий windows, по сути, вынуждает переходить на ту же десятку. Понятно дело, нельзя во всём винить мелкомягких — у них свои резоны, но мне, как пользователю, такая политика не очень нравится.

Am0ralist 21 сентября 2018 в 13:35

0

> Например, отказ от поддержки старых версий windows

Внезапный отказ? Или запланированный?
И вообще, давайте раскритикуем, что они 98 не поддерживают до сих пор? Ну всё же, границы поддержки должны же быть, причем вменяемые? Или разок деньги получили — теперь сто *лет* поддерживайте?

san-smith 21 сентября 2018 в 13:42

0

> И вообще, давайте раскритикуем, что они 98 не поддерживают до сих пор?

Я специально уточнил, что нельзя во всём винить MS\Google/%company_name%. Естественно, нельзя требовать «столетней поддержки». Тем более, что, как я уже сказал, у них свои резоны — никто и не обещал этой поддержки, да и рынок требует.

И всё ж таки, как пользователь, я имею право возмущаться, что в новой версии windows/android/skype/etc всё не так хорошо, как в моей любимой.

DrPass 21 сентября 2018 в 14:03

+3

> Внезапный отказ? Или запланированный?

Сам по себе «отказ от поддержки» ничего плохого не несёт. Ну не поддерживают, и фиг с ним. Никто же не возмущается, что у чайника гарантия три года, а не сто лет?
Плохо здесь то, что во многих случаях новое программное обеспечение перестаёт быть совместимым со старыми версиями ОС, без технической необходимости в этом. Просто ради маркетинга. Т.е. не потому, что разработчики там какой-то новый API закодили, а просто потому, что вставили строчку if (OSVersion < LowestSupported) { SayByeBye(); }



Alexeyslav 21 сентября 2018 в 15:24

0

В основном это происходит не по этой причине, а по причине развития технологий и накопления отрицательного опыта. Сейчас смотришь на 98-ю винду… какой бы она хорошей тогда не была и слава богу что её похоронили. Слишком много практических недостатков, она ещё тогда достигла своего потолка и запуск нескольких приложений уже мог приводить к нехватке внутренних ресурсов(вспомните ресурсы USR и GDI), ограниченность которых жестко прошита в архитектуре операционки. Тут уже хочешь развиваться — надо ломать совместимость.

DrPass 21 сентября 2018 в 15:44

+1

Win 98 — это слишком маржинальный пример. Там архитектурная пропасть между линейкой NT и 9x. Но какие, например, есть технические, а не маркетинговые причины не выпускать сейчас виндовый софт, совместимый вниз до WinXP, а не до Win7 (по крайней мере, для 99% такого софта?).



0xd34df00d 21 сентября 2018 в 20:56

0

Чувак, делавший виндовый платформенный слой получения событий о питании в одном моём опенсорс-проекте, говорит, что нужные API появились только в 7, что ли.

Ну, как конкретный пример.



Alexeyslav 21 сентября 2018 в 23:55

0

А до этого, нужное событие в принципе можно было бы получить но написав уже собственную обертку которая активировалась бы при запуске приложения под соответствующей операционкой. Это лишь дополнительная работа, дополнительный довесок кода в приложение вместо использования готового API.



0xd34df00d 22 сентября 2018 в 00:05

0

Ну оно там всё равно отдельным плагином, так что плагин просто не грузится, и всё.

Но это усложняет систему сборки (потому что теперь надо собирать всё с одними флагами, а этот конкретный модуль — с другими, и проверить, что это ничего не ломает), усложняет распространение, усложняет логику загрузчика модулей, и так далее. И ради чего?

Коммерческие приложения при этом на этот вопрос ответить могут в терминах соотношения между долей пользователей WinXP среди потенциальной аудитории и зарплаты программистов и тестировщиков, а если я делаю некоммерческий опенсорс, то получается «это технически не очень интересная задача, так что ч-т лень».



Alexeyslav 21 сентября 2018 в 23:53

0

Так именно в этом проблема. Отбросили какашку наллевая на совместимость.
Полагаю, что обширный перечень накопившихся костылей препятствует желанию поддерживать софт в старых операционках вроде XP. Во первых, это поддержка старых интерфейсов API, которые могут нести с собой нехилые уязвимости для системы — вспомните SMB v1, v2 и нынешнее v3... некоторые концепции были переосмыслены и некоторый функционал закрыт для прямого доступа в новых операционках, только через новый API другим способом и т.д. ради обеспечения безопасности операционной системы и т.п. вернуть прежний доступ будет означать открыть давно залатаную уязвимость системы. Конечно, часть подобного API эмулируется через различные ухищрения но не в полной мере и с дырами в абстракциях — всякие там режимы совместимости, прозрачное перенаправление доступа в системную папку на пользовательский профиль и т.д. всё это вроде бы работает, но не так хорошо как хотелось бы.

VolCh 22 сентября 2018 в 18:40

0

Банально как проверить эту совместимость, если легально приобрести XP мягко говоря не просто.

khim 22 сентября 2018 в 19:48

0

Для проверки вам не нужна полноценная XP. Достаточно запустить образ XP в Hyper-V

sumanai 21 сентября 2018 в 15:26

0

> а просто потому, что вставили строчку if (OSVersion < LowestSupported) { SayByeBye(); }

Вряд ли кто так делает. А вот версия подсистемы в PE заголовке часто мешает просто взять и запустить софт на старой ОС, хотя по API полная совместимость. Или линковка со системной msvcrt вместо версии конкретной студии, лишь бы не ставить пакет Visual C++ для Visual Studio 20xx.



hermes-jr 21 сентября 2018 в 17:13

0

> if (OSVersion < LowestSupported) { SayByeBye(); }
> Вряд ли кто так делает.

Ооо, нет. Именно так и делает Apple. Помню правил iso образ OS X чтобы поставить, кажется, Lion на старую модель ноута (беленький пластиковый), т.к инсталлятор утверждал, что железо устарело. Ось встала и заработала на ура. Единственное, что изменил — значение переменной а-ля LowestSupportedHwVersion в каком-то файлике.

sumanai 21 сентября 2018 в 20:07

0

> Именно так и делает Apple.

А, ну извиняюсь, не имел дела с продукцией этой компании, кроме разве что одного доисторического айфона.

SuperNovaWS 22 сентября 2018 в 22:48

0

Точно так же делает Microsoft

support.microsoft.com/en-us/help/4012982/the-processor-is-not-supported-together-with-the-windows-version-that

8-ка на 7+ поколении процессоров Интел не получает поддержку из-за наличии буквально такой функции. Если пропатчить пару функций в памяти — все начинает опять работать.

SuperNovaWS 22 сентября 2018 в 22:46

0

Внезапный, не волнуйтесь. Вы знаете, например, что с определенного момента на 8-ке нельзя получить обновлений, если она стоит на машине с Intel 7-го поколения? Просто нельзя — и всё.

support.microsoft.com/en-us/help/4012982/the-processor-is-not-supported-together-with-the-windows-version-that

Ставьте 10ку — и не волнует.

Правда... Есть один маленький патчик. Он подменяет пару вызовов библиотечных функций. После чего пропадает голубое окошко и алдейты становятся без дальнейших танцев с бубнами!

И это называется «вменяемые границы поддержки»? По моему это просто — наипалово типичное.



PsyHaSTe 23 сентября 2018 в 03:39

0

Они таким образом просто не поддерживают старые ОС. Эти патчики в 99 случаев из 100 будут работать, а в одном будут ломаться. Если бы они официально поддерживали восьмерку, их саппорт бы завалили жалобами. А так, сам пропатчил — сам и разбирайся, юникс вей так сказать. Не скажу, хорошо это или плохо, но вполне логично.

sumanai 23 сентября 2018 в 11:17

0

Интересно, с чего бы им ломаться? Ограничение исключительно искусственное.



PsyHaSTe 23 сентября 2018 в 13:51

0

Если вы майкрософт, и у вас есть баг один на миллион, то у вас сотни недовольных клиентов, у которых всё сломалось. Ломаться им с того, что их никто не тестировал на этих ОСях, и никто не знает, достаточно ли разницы между 7 8 и 10 чтобы разломать этот патч.

sumanai 23 сентября 2018 в 14:59

+1

Ну да, пусть будут недовольны 999900 клиентов из-за отсутствия обновлений и требований поставить новую ОС.

Druu 23 сентября 2018 в 15:03

0

Им же, как выше указали, никто не мешает обновление поставить, пусть и с костылем. Просто никто не гарантирует что все будет хорошо.

sumanai 23 сентября 2018 в 19:03

0

Этот костыль нарушает лицензионное соглашение, если что. И да, всё будет нормально работать, несовместимость с новыми процессорами — это блажь.



PsyHaSTe 23 сентября 2018 в 21:40

0

Каким образом он что-то нарушает?

sumanai 23 сентября 2018 в 23:18

–1

Патчингом файлов? Я в упор не знаю, что именно там делается, но уверен, что это нарушает лицензионное соглашение.



PsyHaSTe 24 сентября 2018 в 00:01

0

Майкрософт не может нарушать свое лицензионное соглашение, потому что там наверняка написанно «можем делать что угодно, и вообще вы не купили, а арендовали на бессрочной основе».

sumanai 24 сентября 2018 в 00:12

0

> Майкрософт не может нарушать свое лицензионное соглашение

Вы меня не поняли. Нарушают те, кто ставит патч, обходящий запрет на установку обновлений.



PsyHaSTe 24 сентября 2018 в 00:16

0

Это да, поэтому если у кого-то что-то ломается, майкрософт вместо «ой, чего у вас случилось, сию секунду исправим» говорит «а чей-та вы соглашение нарушаете?». И им не надо это чинить.

Druu 24 сентября 2018 в 03:43

0

Ну все правильно, что не так-то? Костыль же не поддерживается.



SuperNovaWS 23 сентября 2018 в 18:59

0

Так пусть бы и не ставили именно те апдейты, которые ломают систему. Но полностью отключать часть пользователей от апдейтов из-за того, что они купили новые компы — это жлобство высшей пробы! При том — это в системе, которая официально поддерживается и для которой не наступил EoL!

Типа, в микрософте такие — «А нам лень писать патчи для новых ядер — идите нахрен! И пофиг.»

Я даже не знаю, что делать, когда восьмёрку перестанут поддерживать. Видимо — прийдётся жевать кактус линукса, ибо 10ка — это за гранью приемлемого для меня.



PsyHaSTe 23 сентября 2018 в 21:40

0

Ну ваш выбор, голосуйте рублем. Так и надо.

Fracta1L 21 сентября 2018 в 13:02

0

Оперативка несовместима с ОС? Вот уж не думал, что на хабре можно встретить людей аля герои бородатых анекдотов «ваш коврик для мыши не подходит к системному блоку»

Впрочем, дизайн у десятки действительно не очень, и не с точки зрения оформления, а с точки зрения эргономики. Проводник это вообще какая-то свалка.



Alexeyslav 21 сентября 2018 в 14:01

0

До этого недалеко. Уже есть процессор не совместимый со старыми ОС…

SuperNovaWS 22 сентября 2018 в 22:49

0

Несколько ПОКОЛЕНИЙ от РАЗНЫХ вендоров, вообще-то

support.microsoft.com/en-us/help/4012982/the-processor-is-not-supported-together-with-the-windows-version-that



popov654 21 сентября 2018 в 18:20

+2

> а требуют продуктов в стиле Windows 10 (или в «материальном дизайне»

Не требуют, а навязали. И даже многих обновляли принудительно. И XP заранее объявили end of life, хотя могли поддерживать ещё года 2-3 как минимум, не было там особых проблем. Новые протоколы шифрования бы завезли, aac и h264 кодеки, DirectWrite, и IE9 с ним вместе — и была бы сказка. Вы вот спросите пользователей честно, многие ли довольны новым интерфейсом десятки. Мне кажется, результаты вас удивят.

> А современные тяп-ляп сайты создаются на дни, иногда за часы…

А куда спешить-то? Чем круче проект, тем больше времени можно потратить на разработку. Но даже сайт-визитку чтобы сделать прилично, нужно потратить 2-3 дня в идеале. И это не так уж и много. В веб-студии, кстати, если несколько человек будут работать параллельно, получится быстрее.

DMGarikk 21 сентября 2018 в 18:35

–1

Навязали… я отлично помню разговоры в 01-04 годах что MS навязывает тормознутую и обвешенную рюшечками XP, в то время как под 98 реально все работало быстрее, и w2k без лишних рюшечек стабильнее… и только в SP2 и с подешевевшей памятью стало реально в ней жить
==
все идет все изменяется, а придирки старые

khim 21 сентября 2018 в 21:27

+3

> все идет все изменяется, а придирки старые

Потому что и проблемы, как бы, старые, да.



SuperNovaWS 22 сентября 2018 в 22:50

+1

Навязывание установки Windows 10 — уже забыли?

khim 21 сентября 2018 в 21:26

+1

> Не требуют, а навязали.

В том-то и дело, что требуют! Я про это уже писал: человек на полном серьёзе отказывается смотреть как и что работает в gerrit'е, потому что у того, ай-ай-ай — интерфейс из 90х.

Сейчас, впрочем, запилили модночий интерфейс с «материальным стилем», так что всё стало жутко тормозить… но может теперь этим начнут пользоваться?



third112 25 сентября 2018 в 15:23

0

> И если люди отказываются пользоваться интерфейсом 90х, а требуют продуктов в стиле Windows 10 (или в «материальном дизайне»), то, стало быть, цель достигнута.

ИМХО десятка слишком молодая и про ее интерфейс мнение не успело сложиться, а вот некоторые более старые решения Микрософта вызывали бурю возмущения — вспомнить, напр., пресловутую скрепку-помошника в офисных программах. Не все решения бывают хорошими даже у лидеров.



YemSalat 21 сентября 2018 в 08:25

+4

У игр из 90х был полный или почти полный доступ к ресурсам компа (той же видюхе) Браузеры же — это «песочницы», тут у вас банально не получится все так же оптимизировать.

И че все так прицепились к «красивым» пикселям? Гуглокарты со стритвью — это красивые пиксели? Запусите на свой любимой 98ой винде тогда в Зем фаерфоксе.

НЛО прилетело и опубликовало эту надпись здесь

lokkiuni 21 сентября 2018 в 14:13

+2

0*500=0, так что конечно готов))))

staticlab 20 сентября 2018 в 22:12

0

> Автор статьи вообще не понимает, где причина, а где следствие: машину с потребелением топлива 1000 листров на 100 километсров — никто не купит. А сайт, который теряет данные, будет пользоваться большей популярностью, чем сайт «с интерфейсом из 90х». Вот и всё. Вот и вся «тайна».

Если следовать Секрету айсберга, то какой-нибудь арабский шейх всё-таки может купить автомобиль, потребляющий условные 1000 л / 100 км, если **увидит,** как этот автомобиль разгоняется до сотни за 2 секунды, **услышит,** как он при этом рычит, и при этом в нём опять же **видна** вся эта мощь ;)

khim 21 сентября 2018 в 00:22

+3

Ну дык. Вы изобрели концепцию суперкара, да.

Teemon 21 сентября 2018 в 03:40

0

Про это в статье и пишется, что автомобили это могут сделать хотя бы на уровне суперкаров. А программы как-то не особо. Покажите ворд для арабских шейхов, работающий в 100 раз быстрее майкрософтовского?))
Другой вопрос почему такого софта нет?

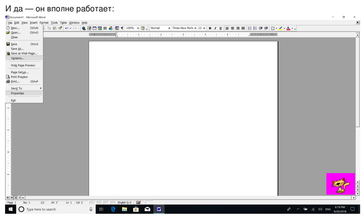
khim 21 сентября 2018 в 04:32

+5

> Другой вопрос почему такого софта нет?

Почему нет? Есть. Поставьте MS Office 2000 на современную систему и он будет запускаться за такое время, что вам покадрово видеосъёмку нужно будет изучать, чтобы заметить, что он делает это не мгновенно.

И да — он вполне работает:



Собственно это основная беда софта: суперкар выпускать выгодно, так как в нём может не быть хитрых подставок для кофе — но скорость окупит всё и его купят.

А быстрый софт выпускать невыгодно: разработка быстрого софта со всеми-всеми-всеми фичами, которые есть в той куче… добра, которое сейчас стоит на полках — невозможна (пока вы будете его разрабатывать выйдут ещё версии с ещё большим количеством фич), а с ограниченным — не нужно, так как старые-то версии не протухают!

Есть довольно большое количество софта, маленького и шустрого (тот же FAR) — но это всё некоммерческие проекты. Ибо заработать на этом нельзя!

SuperNovaWS 22 сентября 2018 в 22:50

0

И что вы в нём будете открывать, кроме вашего институтского диплома?

khim 23 сентября 2018 в 16:25

0

Вы не поверите, но, в общем, всё, что угодно. Поставьте Compatibility Pack — и вперёд.

Да, накладки возможны — ну так они и в рамках одной версии возможны и указывают, скоре на то, что OOXML — плохой стандарт. Но да, это стандарт и да — он поддерживается.

gohan 21 сентября 2018 в 00:56

+3

> Автор статьи вообще не понимает, где причина, а где следствие: машину с потребелением топлива 1000 листров на 100 километсров — никто не купит. А сайт, который теряет данные, будет пользоваться большей популярностью, чем сайт «с интерфейсом из 90х». Вот и всё. Вот и вся «тайна».

Сайт, который теряет данные? Хаха. Сайт, который пойман на воровстве данных и передаче их левым людям, до сих пор наверное топ1 по популярности в мире, если не брать поисковые сайты.

Люди — идиоты, вот и вся тайна. Есть только очень малый процент людей, которые не совсем идиоты, но и то зачастую лишь в какой-то ограниченной очень маленькой области своей компетенции.

Fracta1L 21 сентября 2018 в 10:21

−1

Идиоты это те, что не разделяют ваших ценностей? XDD

muon 21 сентября 2018 в 10:32

+1

> XDD

Только я трактую этот набор символов как «зажмуренные глаза, смеющийся рот, второй подбородок»?

SUA 21 сентября 2018 в 10:50

+1

> машину с потребелением топлива 1000 листров на 100 километсров — никто не купит.

а кроссовер за 2 тонны вместо городского компакта для покатушек на работу вполне. Разница не настолько впечатляет (всего лишь в 3 раза), смысл тот же

sumanai 21 сентября 2018 в 11:05

0

В России это вынужденная необходимость, особенно зимой, когда дороги превращаются в направления. И да, не все живут в Москве.

SUA 21 сентября 2018 в 13:03

+2

Таки нет. Даже с учетом «не Москвы» (просто в ней нагляднее проявляется из-за избытка денег у части жителей).
И написал я именно

> кроссовер за 2 тонны вместо городского компакта для покатушек на работу

поскольку в случае когда вместо дорог уже направления — нужен полноценный внедорожник а не кроссовер

sumanai 21 сентября 2018 в 15:28

0

Проблемы обычно зимой, и только со снегом, где полный привод, пусть и подключаемый только при пробуксовке, спасает, а владельцы переднеприводных машин бегают с лопатой и откапываются вручную.

novice2001 21 сентября 2018 в 13:47

+3

Смысл не совсем тот же. Кроме повышенной проходимости 2-тонный кроссовер еще и сильно безопаснее городского компакта. Учитывая количество идиотов на дорогах и мягкость наказания даже за смертельные ДТП — это совершенно нелишнее.
И полноценный внедорожник нужен в основном только для тех, кто по направлениям вместо

дорог ездит постоянно. Для периодической езды по засыпанным снегом улицам — очень годный компромисс.

crystallize 24 сентября 2018 в 15:39

+2

> Пока куча дерьма в красивой обёртке пользуется большей популярностью, чем маленькая, быстрая, но «негламурная» программа

Возможно просто вы не видите каких-то фичей которые на самом деле привлекают пользователей, а «куча дерьма» на самом деле на другой стороне.

darthmaul 22 сентября 2018 в 11:33

−1

Я так понимаю, минусующих полностью устраивает наше время, в котором веб страницы весят по 50 мб, вся электроника требует замены раз в пятилетку просто ради поддержания функционала, когда автомобили разваливаются сразу после гарантии, а дома делают из прессованной бумаги? Увы, рынок хочет лишь мусора, поярче да подешевле, это научно доказаный факт. Оригинал публикации

VolCh 22 сентября 2018 в 18:26

0

nodejs имеет полное право быть добавлена к PHP/Python. Ничем принципиально от них не отличается.

walti 20 сентября 2018 в 20:19

0

> платить вместо условной тысячи долларов за новый телефон — платить 50 000?

А придется.
Физика круче психологии.
В этом можно убедиться, ~~убедив себя, что человек может летать и шагнув за балкон~~ посмотрев на технологии литографии.
Предел кремния где-то 3нм и 5-6ГГц для десктопа.
Но раньше все упрется в тепло — навскидку, 200 Ватт для десктопа и ватт 5 для смартфона.
Больше — ~~ожоги конечностей~~ Боженька не велел.



demon416nds 21 сентября 2018 в 08:32

0

> Предел кремния где-то 3нм и 5-6ГГц для десктопа.
> Но раньше все упрется в тепло — навскидку, 200 Ватт для десктопа и ватт 5 для смартфона

исключительно если делать чипы двумерными
технологии уже сейчас (на примере МЭМС) позволяют делать трехмерные микросхемы с принудительным распределением тепла жидким теплоносителем.
у смартфонов же ограничение не столько по теплу сколько по автономности
банально батарея не тянет более мощные процессоры. Рассеять же несколько десятков ватт не настолько большая проблема.
Да и частоты больше зависят от топологии чем от техпроцесса

sumanai 21 сентября 2018 в 09:22

0

> Рассеять же несколько десятков ватт не настолько большая проблема.

Уже сейчас телефоны под полной нагрузкой не комфортно держать. А несколько десятков ватт для своего рассеивания потребуют рифления задней крышки как минимум, а то и активное охлаждение.



demon416nds 21 сентября 2018 в 09:26

0

естественно активное охлаждение нужно
но его уже батарея не потянет как я писал
а рифление не поможет тем более большинство пользуется чехлами которые это рифление закроют целиком

sumanai 21 сентября 2018 в 09:38

+5

> естественно активное охлаждение нужно

Я не хочу телефон с вентилятором.

tvr 21 сентября 2018 в 12:48

+2

> Я не хочу телефон с вентилятором.

У вас будет выбор между ~~вентилятором~~ воздушным и ~~насосом~~ жидкостным охлаждением.
«Вот видите — мы предлагаем вам выбор из двух неприемлемых для вас вариантов, а это значит — мы заботимся о вас!» ©



ValdikSS 21 сентября 2018 в 13:22

0

Теплотрубки уже начали вставлять, в Sony Xperia Z2 2014 года, например.

tvr 21 сентября 2018 в 15:07

0

Да много уже куда на самом деле.
Всякие околофлагманские игровые Asus, Xiaomi. Скоро и в бюджетный сегмент лепить начнут.

sumanai 21 сентября 2018 в 15:29

0

Систему ЖО тоже нужно охлаждать, объёмы рассеивателей и вентиляторы там зачастую больше башенных кулеров.

lrgen 21 сентября 2018 в 13:47

+1

А они уже выпускаются



demon416nds 20 сентября 2018 в 22:30

0

Чушь.
Стоимость разработки в пересчёте на устройво копеечная.
Веб-сайты по моему личному опыту прекрасно пишутся вообще без фреймворков. И таки да написанные мной spa летают даже на древних нетбуках. Чего не скажешь о любом сайте на современных фреймворках

Tantacula 20 сентября 2018 в 22:42

+7

Такого рода высказывания хорошо бы подкреплять примерами проектов, чтобы можно было оценить сложность проекта.



demon416nds 21 сентября 2018 в 07:45

0

проекты в основном закрытые
ну вот вам скриншот написанной с нуля админки
ibb.co/bVTANK
96кб не минифицированного javascripta
148кб общий вес с серверной частью



YemSalat 21 сентября 2018 в 08:30

+2

На вид конечно не дизайнерская работа, но оценить качество вашего кода по скриншоту — увольте.



demon416nds 21 сентября 2018 в 08:47

+1

зачем админке «дизайн»???
как нынче говорят юзабилити нужно,
функционал нужен
а любоваться дизайном это не к админке
примеры того как дизайн убивает удобство в админке большинства современных роутеров.
по качеству же размеры кода я написал, функционал по скриншоту прикинуть +- лапоть можно, а пускать вас в закрытую часть сайта чтобы вы убедились что это летает или открывать исходники естественно никто не будет

YemSalat 21 сентября 2018 в 09:07

+7

Ну у вас типичный случай разработчика одиночки.
Когда вы пишите код для своего проекта в 1-2 файлах и вы все помните — где что лежит, что как работает и т.д.
Но попробуйте разобраться в чужом таком 'ref' проекте — сразу поймете в чем проблема.
К тому же качество кода из примера ниже явно хромает.

ПС
Дизайн не обязательно должен быть «крутым», достаточно быть не вырвиглазным, а для этого хватает простейшего ucc.

popov654 21 сентября 2018 в 18:33

0

Я думаю, нет особой проблемы разобраться в чужом проекте, если можешь разобраться в своём — по крайней мере если вы и автор того проекта мыслите не сильно по-разному.

YemSalat 21 сентября 2018 в 18:49

0

> Я думаю, нет особой проблемы разобраться в чужом проекте, если можешь разобраться в своём

В реальных проектах под нагрузкой у вас банально не будет времени долго врубаться в чужой код и проверять что ваш ничего не сломал.

НЛО прилетело и опубликовало эту надпись здесь

YemSalat 21 сентября 2018 в 09:03

+4

Если это правда так, то тут гордится «96кб не минифицированного javascripta» думаю особо не стоит :)
Вот этот момент порадовал:

```
// Пошла магия
...
```

demon416nds 21 сентября 2018 в 09:07

0

этот скрипт написали до меня как и большую часть сайта к слову))
хотя пару незначительных правок в нем таки делал.
сейчас точно не скажешь
но человек семь до меня точно в коде отметились

PsyHaSTe 21 сентября 2018 в 12:05

0

То «сайты прекрасно пишутся», то «Да там до меня 7 человек было, вот эту грязь вообще не я писал». Вы уж определитесь :)

demon416nds 21 сентября 2018 в 12:22

0

вроде понятно все писал
но всегда найдутся желающие придраться
на данном сайте на данный момент с нуля мной написана только админка ибо старая безбожно тормозила. Все остальное планируется к замене но еще не заменено. Есть и другие сайты написанные полностью мной но они «для внутреннего пользования»
и посторонние дальше запроса пароля ничего не увидят

PsyHaSTe 21 сентября 2018 в 12:23

+3

Если на пример "сайта без современных фреймворков" не взглянуть без слез, то это очко в пользу этих самых современных фреймворков, вы же понимаете?

demon416nds 21 сентября 2018 в 12:29

-4

бедняжко контуженный материальным дизайном
вот тебе пример современных фреймворков
вот где глаза кровоточить начинают и шаринган пробуждается
ibb.co/h5wOsK

popov654 21 сентября 2018 в 18:36

0

Просто у человека нет знакомого дизайнера, который сделал бы красивый и аккуратный UI. Это можно простить. Из меня тоже дизайнер не очень (хотя мне тоже дизайн со скриншота выше кажется уж слишком аляповатым).

В целом я согласен, при наличии хорошего дизайна, прямых рук, и понимания, что ты делаешь и зачем — реально написать хороший, быстрый и надёжный код без использования фреймворков вообще. Мой скромный опыт в 2-3 проектах это подтверждает :)

PsyHaSTe 21 сентября 2018 в 18:39

0

Не пользоваться фреймворкаи это как не пользоваться IDE, не пользоваться средствами ОС, не пользоваться драйверами, а пытаться всё-всё-всё запихнуть в одно приложение. Есть хорошие принципы SRP и DRY, которые дико нарушаются таким подходом.

Grox 21 сентября 2018 в 21:01

0

Так вопросы дизайна и решаются дополнительными библиотеками шаблонов. Bootstrap или целые модули админок. Но да, они часто тащат jquery. Но его можно не использовать.

fukkit 21 сентября 2018 в 16:58

0

Крысы весело разбегаются за границы экрана, появляется горизонтальный скролл. Неудачно человек выступил)

DelphiCowboy 21 сентября 2018 в 10:25

+1

> Веб-сайты по моему личному опыту прекрасно пишутся вообще без фреймворков. И таки да написанные мной spa летают даже на древних нетбуках. Чего не скажешь о любом сайте на современных фреймворках

Только вот при устройстве на работу в требованиях для Веба, **по причине КАРГО-культа**, маячат требования знать очередной новомодный фреймворк.

GrigoryPerepechko 21 сентября 2018 в 11:46

+2

По причине нежелания полностью завязаться на одного человека. Фреймворки хоть и дают безумный оверхед, но позволяют в коде оставить по большей части только доменную логику. А в ней новым людям проще разобраться чем в тонких архитектурных абстракциях

popov654 21 сентября 2018 в 18:34

0

Поэтому гораздо проще фриланс)

yarric 20 сентября 2018 в 22:46

+4

В 90-х что-то не приходилось платить 50к за Windows 95, хотя зоопарк устройств был тот ещё: и дискеты, и сидюки, и диски увеличенной ёмкости разных форматов, далее — пяток производителей «графических ускорителей», пара производителей звуковых карт (что это?) и тд

Frankenstine 21 сентября 2018 в 07:43

0

Однако, 95-я винда стоила таки заметно дороже «десятки»: 210 баксов за коробку, против примерно 100 (и ещё можно учесть инфляцию).

SergeyMax 21 сентября 2018 в 08:07

0

Но ведь Pro версия десятки как раз стоит около 200 долларов.

Frankenstine 21 сентября 2018 в 16:51

0

95-я не тянят на прошку :) На хоум лишь.

DrPass 21 сентября 2018 в 17:35

0

В 1995-м очень даже тянула :) В офисах найти Windows NT было ещё нереально, у всех 95 или 3.11 FWG

Frankenstine 21 сентября 2018 в 20:59

+1

Вообще-то, в 95-м Windows NT 3.1 (первая с поддержкой NTFS) уже была 2 года в продаже и даже получила к этому времени третий сервис-пак. Хотя, конечно, расцвет линейки начался с Windows NT 4.0 в 1996-м. Но поскольку 95-я вышла **через** два года после NT 3.1, ваше утверждение что в офисах была 95-я в 95-м, а NT 3.1/3.5 не было, выглядит голословно и подозрительно нелепо. Более того, википедия фактически говорит что вы не правы:

> Windows NT 3.1 sold about 300,000 copies in its first year.

И это несмотря на высокие аппаратные требования на тот момент (recommended system requirements of a 486 processor with 16 megabytes of memory), неподдержку офисных пакетов (и малое число программ вообще) и высокую цену: the workstation version for $495, and the server version for $1,495. Заметьте, что сейчас Windows Server 2016 Standard стоит немного дешевле в числовом значении (без учёта инфляции), а Windows 10 Pro for Workstations дешевле Windows NT 3.1 for Workstations почти в два раза. Причём вначале Майкрософт заявлял что цена на сервер будет такой лишь пол года, а потом повысится до $2,995, но так и не была повышена (разработка, проведённая силами 250 программеров, стоила 150 мегабаксов, т.е. окупилась лишь через год, когда подоспела NT 3.5).
Конечно, в мелких оффисах до выхода NT 4.0 (и соответствующего железа) в 1996-м было вероятнее встретить 95-ю, но в крупных — очень сомнительно из-за низкой устойчивости. В сети с хотя бы сотней компов 95-я падала наверное на одном из компов каждые пять минут :)
При этом надо признать, 95-я имела отличный коммерческий успех, продавшись миллионом экземпляров в первые 4 дня после релиза.
Жаль, в вики нет данных по продажам NT 4.0.

khim 21 сентября 2018 в 21:47

+2

> Более того, википедия фактически говорит что вы не правы:
> Windows NT 3.1 sold about 300,000 copies in its first year.

Википедия говорит о том, что вы не умеете пользоваться википедией. Потому что в другом её разделе можно обнаружить, что ежегодные продажи персоналок уже в те годы достигли **десятков миллионов**… на этом фоне триста тысяч… не смешите мои тапочки…

Или на другую статью посмотрите о Windows 95:

> Sales were strong, with one million copies shipped worldwide in just four days

Действительно некоторое время после выхода Windows 95 она оставала от Windows NT 3.1. Примерно неделю. К концу года количество установок и сравнивать было нельзя.

> Конечно, в мелких оффисах до выхода NT 4.0 (и соответствующего железа) в 1996-м было вероятнее встретить 95-ю, но в крупных — очень сомнительно из-за низкой устойчивости.

В 1995м можно было чаще встретить Windows 3.1, но никак не Windows NT 3.1

> Конечно, в мелких оффисах до выхода NT 4.0 (и соответствующего железа) в 1996-м было вероятнее встретить 95-ю, но в крупных — очень сомнительно из-за низкой устойчивости. В сети с хотя бы сотней компов 95-я падала наверное на одном из компов каждые пять минут :)

Ну так Windows 3.11 была ещё глюкавее, а Windows NT была экзотикой, которую никто не видел. Так что приходилось мириться, да…

Frankenstine 21 сентября 2018 в 22:16

0

> ежегодные продажи персоналок уже в те годы достигли десятков миллионов… на этом фоне триста тысяч… не смешите мои тапочки

Если вы говорите таки не про персоналочки, а про офисы, то не забывайте, что до появления на рынке линейки Windows NT, этот рынок был на 90% «под пятой» Novell NetWare, к 95-му году уже пару лет как имевшей в активе NetWare 4 с NetWare Directory Services, аналог которого у Майкрософта (Active Directory) появился лишь через ещё пять лет. Так что если уж про офисы, причём крупные, то в 95-м можно было таки скорее увидеть эту самую нетварь, чем мастдай.

Вы просто молоды видимо, а я 95-й помню, потому что как раз школу заканчивал.

DrPass 22 сентября 2018 в 03:36

+2

> Если вы говорите таки не про персоналочки, а про офисы, то не забывайте, что до появления на рынке линейки Windows NT, этот рынок был на 90% «под пятой»

Это абсолютно верно, в офисах *крупных компаний* тех лет сетевой ОС была чаще всего как раз нетварь, реже всякие *никсы. И не только до появления линейки Windows NT, а ещё долгое время после её появления. Перелом был где-то в 1997-м году. Но есть момент, который вы упустили — хороший и очень популярный клиент нетвари как раз был встроен в Windows 95 :)

Frankenstine 22 сентября 2018 в 10:38

0

Microsoft Client for NetWare предоставлял лишь доступ к файлам и к принтерам (без авторизации в NDS) в сети нетвари. Ничего другого, например авторизации в NDS и доступ к другим сервисам, он не предоставлял. Так что с «про» ролью мастдай не справлялся, как минимум до покупки пакета «плюс» (содержавшего интернет эксплорер 1.0 и планировщик задач) и новеловского Client 32. Таким образом, сравнение 95-й винды с 10-й хоум здиш полностью оправданно.

khim 26 сентября 2018 в 22:28

0

> Таким образом, сравнение 95-й винды с 10-й хоум здиш полностью оправданно.

Это если сравнивать сетевые возможности. Но, как известно, киллер-фичей Windows в те годы была возможность запуска DOS, а вовсе не продвинутые сети. И тут у нас оказывается что по этому параметру любая, самая простецкая версия DOS попадает туда же куда и Windows Professional.

Сравнивать же Novell Netware и Windows 95 по меньшей мере *глупо* — это были операционки, которые **вообще** никак не пересекались. Вот Microsoft OS/2 (да-да, вы не поверите, но именно Microsoft OS/2), а позже и Windows NT были прямым конкурентом Novell Netware… но это совсем другая история.

mkshma 20 сентября 2018 в 22:50

+4

> А ещё придётся платить за каждый веб-сайт, потому что разрабатывать каждый раз всё с нуля, без использования библиотек и фреймворков — будет очень и очень дорого.

Не придется. Действительно технически сложных сайтов можно пересчитать по пальцам пары рук. Бэкенд если не элементарнейший, то довольно прост. Фронтенд вообще отдельная история. Его представители уже не первый год всеми правдами и неправдами раздувают стек технологий, лишь бы им платили хотя бы примерно сопоставимые деньги с работниками бэкенда. На бэкенде та же фигня: понапридумывали всякого мусора вроде микросервисов, а потом генерируют еще больше мусора вроде Докера и Кубернитис, чтобы героически решать самими же созданные проблемы. И даже не понятно с какой стороны к этой проблеме подступаться.

Можете даже смеха ради почитать вот этот увлекательнейший тред news.ycombinator.com/item?id=17915560 — половина отметившихся искренне недоумевает как это такой простейший сайт, да без фреймворков и прочего мусора запилили. Ведь должно быть все по гайдлайнам, а у них, чертей эдаких, все в одном JS файле лежит. Это все, что нужно знать про современный фронтенд.

staticlab 21 сентября 2018 в 00:22

0

Насчёт простейшего согласен. А что делать в случае более сложных проектов?

YemSalat 21 сентября 2018 в 08:36

+2

> Это все, что нужно знать про современный фронтенд.

Вы взяли сайт где люди под статьями оставляют коменты. Там думаю действительно можно ограничится одним js файликом. (Причем в первом же комменте там приводится пример того почему это не масштабируется)

Напишите фронт к какому-нибудь гмэйлу в одном файле, еще и командой из 20ти человек, раз у вас так от фронтов бомбит.

muon 21 сентября 2018 в 09:20

+3

> Напишите фронт к какому-нибудь гмэйлу

Написать интерфейс к почте сильно сложнее, чем к сайту, где люди под статьями оставляют коменты?

YemSalat 21 сентября 2018 в 18:43

+2

Ээ… Да?

SuperNovaWS 22 сентября 2018 в 22:56

0

Э… Чем?!

YemSalat 24 сентября 2018 в 15:11

0

По-моему очевидно что интерфейс гмэйла сложнее чем у хакерньюса.

muon 25 сентября 2018 в 07:40

+1

Сложность интерфейса не представляет ценности.
Ценность представляет сервис — почта.
Почта — не сложный сервис.
Почта не требует сложного интерфейса.
Если у простого сервиса сложный интерфейс, то это недостаток, а не достоинство.

VolCh 25 сентября 2018 в 09:51

0

Ценность представляет эффективность решения реальных задач с помощью сервиса. Задача «доставить сообщение адресату(ам)» — это техническая задача, а не реальная. Для технической задачи достаточен telnet клиент (если говорить о SMTP/POP3 и ко). Но вы сами будете пользоваться почтовым сервисом, который из пользовательского интерфейса предоставляет только соотвествующие порты? Ну то есть вообще нет ничего больше.

muon 25 сентября 2018 в 10:31

0

Вы не сформулировали реальную пользовательскую задачу.
Обычно сам пользователь формулирует задачу как «чтобы почта работала». Ему знакома сущность «письмо», но не «TCP-порт». С помощью telnet-клиента редкий бухгалтер отправит письмо, а с помощью SquirrelMail — запросто.

VolCh 25 сентября 2018 в 18:38

0

Это вы её не сформировали перед тем как рассуждать о достинствах и недостатках, и часто под «чтобы почта работала» оказывается имеется в виду гораздо больше чем просто создание, отправка и чтение писем в простом интерфейсе.

YemSalat 25 сентября 2018 в 12:51

+2

То что вы описываете — это базовый простейший интерфейс для почты.
Но у всех пользователей разные запросы.
Например, я хочу иметь возможность галочками отметить нужные письма из списка и перетащить их мышкой в другую папку. Загружать аттачменты перетаскиванием файлов. Иметь нормальный wisywig и так далее.
Базовый интерфейс всего этого не покрывает. Поэтому гмэйл для меня удобен — а squirrelmail это когда нету альтернатив.

muon 26 сентября 2018 в 03:18

0

> галочками отметить
> перетащить их мышкой
> нормальный wisywig

Это может быть нужно и для писем, но и для комментариев. Всё ещё непонятно, чем почта сложнее сайта, где люди под статьями оставляют коменты.

VolCh 26 сентября 2018 в 10:27

0

Зачем вам комментарии таскать мышью? Или отображать один комментарий в нескольких «папках»?

muon 27 сентября 2018 в 02:06

0

> Или отображать один комментарий в нескольких «папках»

Это называется «теги». Они удобные, хотя я их и не просил.

> Зачем вам комментарии таскать мышью

Затем же, зачем вам перетаскивать письма.
Я могу придумать множество юзкейсов и для работы с комментариями, вы — с письмами, но показать принципиально большую сложность почты всё равно не получится.

YemSalat 27 сентября 2018 в 18:29

0

В ваших условиях невозможно сравнить сложности вообще чего-либо т.к. требования постоянно меняются.

muon 28 сентября 2018 в 08:00

0

Ну так я и не утверждал, что один класс интерфейсов сложнее, чем другой.
Наверное, если хочется порассуждать на эту тему, надо описать объекты, с которыми нужно работать. И прикинуть возможные действия.
Что в почте, что в «сайте с комментариями» объект — это набор текста, картинок, вложений, с отправителем и получателем в метаданных. На мой взгляд, объекты одного порядка сложности. А в мейлинг-листах они вообще срастаются в одно.

novice2001 21 сентября 2018 в 09:22

+2

Зато вы берете уникальные сверхпопулярные и высоконагруженные проекты.
Только вот 99,999% остальных сайтов в аналогичном стеке не нуждаются, но используют, зачастую ещё и очень криво. В результате все безбожно тормозит.

YemSalat 24 сентября 2018 в 15:11

0

Пару примеров таких сайтов в студию.

CheatEx 25 сентября 2018 в 14:52

+1

Хабр

YemSalat 25 сентября 2018 в 18:54

0

Ну хабр то как раз скорее в «старом» стиле.
Тут, вроде как, никакие фреймворки на фронте не используются (подключен jQuery 1.8.3) и большая часть рендерится на сервере.

CheatEx 26 сентября 2018 в 00:09

+1

У меня в этой теме 3 кадра в секунду. В Fallout 4 — 144.

YemSalat 26 сентября 2018 в 07:54

+1

Мы говорили о сайтах сделанных на современных фреймворках. Фронт хабра написан на jQuery и каких-то кастомных скриптах и умудряется тормозить.
Может если бы они использовали современный фреймворк и, например, сделали ли бы ленивую подгрузку/отображение комментариев — то так сильно не тормозило бы.

sumanai 26 сентября 2018 в 20:14

+1

Ленивая подгрузка это такой способ заставить тормозить на всём протяжении просмотра страницы, а не только n секунд вначале.

PsyHaSTe 27 сентября 2018 в 21:49

0

Если это так, то я боюсь её уже включили.

А вообще виртуализация отображаемой области — король на рынке экономии производительности и убирания лагов со времен Дельфи.

sumanai 27 сентября 2018 в 21:51

0

По идее этим должны заниматься сами браузеры.

Druu 28 сентября 2018 в 03:04

0

> По идее этим должны заниматься сами браузеры.

Они не могут, это нарушит спецификацию.

mkshma 21 сентября 2018 в 15:21

+3

> Напишите фронт к какому-нибудь гмэйлу в одном файле

Это же невообразимо сложно подгрузить json'ку с заголовками с сервака и отобразить их построчно, припорошить кучей иконок, а по нажатию на заголовок уже подгрузить само письмо. Это элементарнейший веб-интерфейс, в котором ничего не может работать медленно. У Яндекса работает быстро, у Мыла, прости господи, работает быстро, навороченный и кастомизируемый интерфейс ProtonMail работает быстро. И вот ирония, у гугла есть HTML версия, которая, ну вы поняли — работает быстро. Так как же вот эта элементарщина в основной версии у них умудряется лагать сильнее, чем их же убогие карты? А все просто: огромная команда разработчиков гугла просто оправдывает свое существование. Сначала им, видите ли, не понравился простой и функциональный дизайн, поэтому напрягли команду дармоедов испортить интерфейс, потом напрягли пару сотен дармоедов испортить фронт, потратили на это год, а может два и теперь примерно столько же будут героически превозмогать. А потом новый фреймворк, над вермеодеями дизайнерами снова нависнет угроза увольнения и цикл повторится. Та же фигня у них с картами. Наступил коммунизм и мы теперь можем рендерить страницы на RTX2080 с рейтрейсингом, поэтому почему бы нам не воткнуть кучу 3D в КАРТЫ! А, нет, коммунизм не наступил и люди пытаются пользоваться этим на Intel HD4000 с 4 гигами оперативы и, о неожиданность, оно лагает.
Веб стал слишком перегружен. Где достаточно открыть новую вкладку — мы начнем рендерить в той же. Динамическая подгрузка вместо пагинации, раздутый дизайн, куча анимаций, хотя они нафиг не упали — зачем все это? Это не имеет никакой полезной нагрузки, так зачем все это запихивать на страницу? А потому что иначе мы уже завтра увидим сокращение штата в каждой крупной конторе процентов эдак на 30. Но работа есть работа, свое существование надо оправдывать.



YemSalat 24 сентября 2018 в 15:13

+1

> Это же невообразимо сложно подгрузить json'ку с заголовками с сервака и отобразить их построчно, припорошить кучей иконок, а по нажатию на заголовок уже подгрузить само письмо. Это элементарнейший веб-интерфейс,

Согласен, только это не интерфейс гмэйла.
Специально для таких как вы у них есть html-only версия, там именно так все и работает как вы написали.



muon 25 сентября 2018 в 07:42

0

А какие дополнительные достоинства есть у полной версии интерфейса?



YemSalat 25 сентября 2018 в 12:52

0

я там выше отвечал:

> … возможность галочками отметить нужные письма из списка и перетащить их мышкой в другую папку. Загружать аттачменты перетаскиванием файлов. Иметь нормальный wisywig и так далее..

mkshma 25 сентября 2018 в 13:04

0

Ею и пользуюсь. Весь остальной визуальный мусор только мешает.
//Если уж говорить о мусоре, то Хабр теперь в моем топе вплотную приблизился к лидерам: перейти из профиля к своему комментарию в этом треде занимает аж полминуты. Т.е. всего чуть менее чем 2к комментариев кладут Хабр на лопатки. Зато красивенько, да.

Whuthering 21 сентября 2018 в 10:24

0

Микросервисы все-таки задумывались как вполне рабочее и надежное решение пары вполне конкретных проблем. И без них было бы гораздо хуже. А называть «ненужным мусором» все, что вы не понимаете, звучит немного глупо:)

VolCh 22 сентября 2018 в 19:27

0

Скорее как средство замены одной проблемы на другую и в целом с понижением формальной надежности, но повышением в некоторых случаях практической, когда падение маловажной или просто ненужной сейчас части системы приводит лишь к деградации, но не полному отказу.

Alexmaru 21 сентября 2018 в 00:36

+2

а тут вы неправы. И у эппла, и у гугла доходность унеслась в стратосферу, но штат людей от это и качественно, и количественно меняется слабо.



Berkof 21 сентября 2018 в 07:01

+4

Почитайте труды доктора Демминга… Это такой американский учёный, который пытался всем вокруг сказать идеи толкать, что мол хватит лепить абы как, давайте всё посчитаем аккуратно и выйдет, что можно сделать лучше и даже ещё и сэкономить…
Я занимаюсь разработкой ПО уже много лет и вижу (на живых примерах, т.к. сменил с десяток компаний уже), что может оказаться дешевле писать качественные программы, чем некачественные… и конечно никто не хочет делать себе плохо (ну может эмо, но мы не про них), просто люди где-то поленились (+1 контринтуитивный интерфейс), где-то поторопились (+1 порядок тормознутости программы, где человек не разобрался как в том кривом интерфейсе обновить данные и просто переполучил заново всю пачку), недальновидность (следующий разраб не стал пытаться улучшить что есть, а просто разделил это на воркеров в микросервисах и развесил на кластер, чтобы выдержать возросшую нагрузку)… Но страшно даже не это, страшно то, что работая с хорошей программой хороший программист сразу понимает что и почему тут там сделано и куда именно ему нужно внести правки, чтобы добавить нужный функционал. А с плохой так не прокатывает и ты либо тратишь гору времени на изучение всех хитросплетений костылей, либо просто втыкаешь ещё парочку (первый — чтобы сделать функционал, второй — чтобы подлепеть то, что отвалилось из-за первого, который не на месте)… В итоге ты тратишь на задачу много времени, начальство недовольно и ты решаешь, что можно и без тестов нормальных, абы как, вроде работает… А в следующий раз тебе уже и страшно выдёргивать какие-то костыли, даже если время есть, потому что тестов то нету и вдруг ты не все хитрости понял и что-то сломаешь?
Ну а у Демминга ещё ПО не было, он обсуждал, что дешевле заказать более дорогую доставку, но чтобы товар доставили ровно в 9:05 к воротам цеха, чтобы рабочие занесли материал и сразу пошли его обрабатывать, без склада, кладовщика и перетаскиваний между помещениями. А дальше он писал и вовсе чудестные вещи, что произвести качественный товар даже дешевле — если покупатель видит, что у вас товар лучше, то он и брать будет у вас, т.е. в долгосрочной перспективе реклама почти не нужна; если рабочий будет видеть, что его товары лучше всех — он будет испытывать гордость за свою работу, а не стыд — это мотивация не увольняться; если будет меньше брака — будет меньше затрат на доделку на выходном контроле и меньше возвратов от клиента и т.д.



PsyHaSTe 21 сентября 2018 в 12:12

+1

> А с плохой так не прокатывает и ты либо тратишь гору времени на изучение всех хитросплетений костылей, либо просто втыкаешь ещё парочку (первый — чтобы сделать функционал, второй — чтобы подлепеть то, что отвалилось из-за первого, который не на месте)… В итоге ты тратишь на задачу много времени, начальство недовольно и ты решаешь, что можно и без тестов нормальных, абы как, вроде работает… А в следующий раз тебе уже и страшно выдёргивать какие-то костыли, даже если время есть, потому что тестов то нету и вдруг ты не все хитрости понял и что-то сломаешь?

У меня на прошлой работе так было. Только там было еще веселее — выдернутые костыли требовали вернуть обратно, ибо "вдруг что сломается", а в качестве верного решения предлагалось закостылить вот этот конкретный кейс из джиры.

Мой любимый случай был когда мне упала задача на дыры в API, я нашел больше сотни дыр, которые позволяли одним пользователям управлять аккаунтами/данными/… других. Подумал немного, сделал один аутентифицирующий фильтр, который закрывал разом все возможные дыры, и отправил на ревью. Ревью оно не прошло — "слишком много изменений, ты же всю платформу затронул, кто это тстировать будет?!?!". Итоге — решение не принято, все эти дыры так и висят до сих пор, насколько я могу судить.

VolCh 22 сентября 2018 в 21:47

0

«Кто это тестировать будет» ещё нормально по сравнению с «я не знаю как это тестировать»

senglory 23 сентября 2018 в 02:38

0

Что, в общем, правильно. Т.к. с автора изменений и надо в первую очередь спрашивать план тестирования.

Druu 23 сентября 2018 в 03:52

0

Это с чего бы это вдруг?

Здесь же речь не о новом функционале, а о том, что не должно сломаться то, что и так работало — с-но, уже должны иметься тесты, которые проходит текущая реализация. И добавлять к ним ничего не надо.



PsyHaSTe 23 сентября 2018 в 03:58

0

Нужно проверить каждую АПИ по двум сценариям — правильный запрос (200 ответ) и неправильный запрос (должен зарубиться фильтром). Проверили — запушили, дыры закрылись, профит.

VolCh 23 сентября 2018 в 18:14

0

Тесты, показывающие ожидаемые изменения предоставлены были. Реплика констатировала, что никто не знает как провести регресс-тест, что изменения не сломали где-то что-то в сильно связанной архитектуре. Автор небольших изменений должен полный план регресса писать?

duodvk 25 сентября 2018 в 19:11

0

Это сильно напомнило Генри Форда.



LLarean 21 сентября 2018 в 11:27

0

Задавался тем же вопросом, что и автор, но мне кажется, что производительность технологий значительно обгоняет скорость разработки софта.
Мне проще работать на unity, которая даст доступ к необходимым инструментам и позволит реализовать идеи заказчика, чем конструировать инструмент или подбирать что-то, что будет реализовывать аппаратный потенциал.

denisromanenko 21 сентября 2018 в 12:08

+2

Дело не в том, чтобы писать для каждого куска железа чистый код с нуля.
Тем более что это невозможно.
Дело в том, чтобы не добавлять немного своего дерьма на каждом шаге разработки (компилятор-среда-ядро-система-окружение-фреймворк-ПО) к общей куче дерьма, которое выльется на пользователя.



FadeToBlack 21 сентября 2018 в 13:59

0

Почему-то вы вместо решения проблемы пытаетесь ее раздуть. Не нужно писать 100500 фреймворков, делающих одно и то же, нужно один раз сделать один нормальный. Так везде. Но люди настолько заглодили все сущностями, что и не хватает времени как следует с ними разобраться. Первый текстовый редактор был написан десятки лет назад, наверняка, он не был быстр. Но проблема осталась — я все еще жду свой компьютер, пока он что-то там сделает, хотя я просто набираю текст.

VolCh 22 сентября 2018 в 21:56

0

Нельзя сделать один нормальный фреймворк, а если можно было бы, то его сделать частью языка надо было бы. А ещё сделать один нормальный язык и никто не уйдёт обиженным.



FadeToBlack 23 сентября 2018 в 08:15

+1

Очевидно же, что проблема не в том, что нельзя, а в том, что люди не умеют договариваться. А на обиженных воду возят. Я имею ввиду, что нельзя просто так взять и обидеться на хороший язык программирования.

Osnovjansky 25 сентября 2018 в 13:54

0

Именно что нельзя. Для разных задач оптимальными будут разные фреймворки/языки

C++ хороший язык, но я на него обижен, что слишком много неявных преобразований и других способов «выстрелить себе в ногу»
C# тоже хороший язык, на него я обижен за отсутствие множественного наследования и автоматическое управление памятью
Области оптимального применения этих языков пересекаются, но совсем не совпадают и совместить их, скорее всего, не реально



PsyHaSTe 25 сентября 2018 в 14:30

0

Перефразируя:

Не люблю C++ за ручное управление памятью
Не люблю C# за автоматическое управление памятью

Вы уж извините, но тут третьего пока не придумали :)

Бтв, множественное наследование реализации — известное зло, а множественное наследование интерфейсов в C# есть.



Am0ralist 25 сентября 2018 в 14:33

0

> Не люблю C++ за ручное управление памятью
> Не люблю C# за автоматическое управление памятью
>
> Вы уж извините, но тут третьего пока не придумали :)

Срочно нужен C-AI, с управлением памятью с помощью ИИ!



0xd34df00d 25 сентября 2018 в 20:12

0

И организацией хипа в виде блокчейна.



Am0ralist 25 сентября 2018 в 21:55

0

И организацией хайпа с последующем переводом на эти технологии всех прочих языков, включая asm!

Osnovjansky 25 сентября 2018 в 15:04

0

Ручное управление памятью в C++ это не совсем правда — вы вполне можете, если это вам нужно, пользоваться собственными аллокаторами, включая автоматическую сборку мусора и сжатие свободного места (правда, подозреваю, код будет довольно громоздким в написании)

Что касается множественного наследования реализации, когда-то читал книгу по ATL — очень понравилась идея, когда нужный базовый функционал соответствующих интерфейсов добавлялся как классы-примеси и оставалось только добавить собственный специфический код (причем обычно сравнительно немного)
Воспользоваться, правда, не успел — похоронили раньше чем могло понадобиться.

Мне кажется, в некоторых случаях, наследование классов-примесей, может помогать получить более компактный код, чем использование классов шаблонов алгоритмов. Могу и ошибаться — соответствующего опыта не имею.



PsyHaSTe 25 сентября 2018 в 15:25

0

> Ручное управление памятью в C++ это не совсем правда — вы вполне можете, если это вам нужно, пользоваться собственными аллокаторами, включая автоматическую сборку мусора и сжатие свободного места (правда, подозреваю, код будет довольно громоздким в написании)

Если всё строить на умных указателях — то вы *почти* в безопасности. Плюс шаред_птры построены на атомиках, поэтому от многих будет еще и негатив в стиле "разбазаривает процессорное время".

Но чаще везде какие-то полуавтоматические меры, которые постоянно взрываются так или иначе. По крайней мере реальный код, багрепорты и прочее говорят об этом.

> Что касается множественного наследования реализации, когда-то читал книгу по ATL — очень понравилась идея, когда нужный базовый функционал соответствующих интерфейсов добавлялся как классы-примеси и оставалось только добавить собственный специфический код (причем обычно сравнительно немного)
> Воспользоваться, правда, не успел — похоронили раньше чем могло понадобиться.

Ну, смотрите. У вас есть множественная реализация интерфейсов. Интерфейсы идут с методами-расширениями или дефолтной реализацией. То есть поведение (некоторое) наследуется. Есть наследование реализации от одного базового класса — тоже всё ок.

Вся разница — нету множественного наследования виртуальных методов. Но если вдруг такое понадобилось, то я скорее предположу проблему в архитектуре. Разобраться потом с этим будет крайне тяжело.

sumanai 26 сентября 2018 в 20:16

0

> Воспользоваться, правда, не успел — похоронили раньше чем могло понадобиться.

Вы будете смеяться, но в PHP 5.4 и новее как раз так и сделано, можете пользоваться

))

sentyaev 25 сентября 2018 в 16:46

0

> Бтв, множественное наследование реализации — известное зло

Многие так говорят, но если разобраться, то не могут привести ни одного аргумента почему это плохо. Единственное что могут привести как пример — ромбовидное наследование, но эта проблема давным давно решена, например в Python.

Я понимаю, что случаев когда множественное наследование необходимо не так уж и много, да и написание и поддержка такого решения не самое тривиальное занятие, но это не значит, что нужно искусственно ограничивать язык.

Я это не к тому, что нужно использовать множественное наследование всегда.
Я как раз за то, что нужно стараться композицию объектов, но бывает, что наследование лучше, а иногда бывает, что множественное наследование нужно.
Т.е. как обычно все зависит от задачи, а из-за искусственных ограничений приходится придумывать паттерны и костыли.

PsyHaSTe 25 сентября 2018 в 18:10

0

Есть подходы, которое вообще любое наследование запрещают под соусом «используйте делегирование». Они, надо понимать, вообще ничего в жизни не понимают?

sentyaev 25 сентября 2018 в 18:54

0

Смотрите, сначала вы говорили, что множественное наследование — зло.
Теперь вы приводите пример, что кто-то говорит, что наследование — зло.

Я не говорю, что эти утверждения не верны. У меня вопрос другой, зачем самому себе искусственно ограничивать возможности моделирования предметной области?

Просто это выглядит так, что кто-то увидел какой ад можно запроектировать с использованием множественного наследования, или создать страшную иерархию классов которую невозможно расширить не сломав всю систему и решили: да ну его, просто запретим и проблем у нас не будет.

В моем видение код должен быть написан как можно проще, если можно обойтись без наследования — отлично. Но если проект у вас не тривиальное CRUD приложение, то рано или поздно придется строить сложные иерархии, т.к. это будет проще, чем страдать от того, что кто-то решил, что наследование — зло, запретить, множественное наследование — зло, запретить.

Некоторые считают, что переменная должна состоять только из одного слова и не более 8 символов.

PsyHaSTe 25 сентября 2018 в 19:57

0

> Смотрите, сначала вы говорили, что множественное наследование — зло.
> Теперь вы приводите пример, что кто-то говорит, что наследование — зло.

Я просто хотел начать с более простого примера. Ведь если отсутствие множественного наследования это по вашему ошибка, то отсутствие наследования вообще наверное выглядит как бред и искусственное ограничение. Дальше, порассуждав о причинах подобного, было бы проще обсуждать отсутствие наследования только реализации. Как с гнутой палкой — чтобы её выпрямить, нужно перегнуть в противоположенном направлении.

> В моем видение код должен быть написан как можно проще, если можно обойтись без наследования — отлично. Но если проект у вас не тривиальное CRUD приложение, то рано или поздно придется строить сложные иерархии, т.к. это будет проще, чем страдать от того, что кто-то решил, что наследование — зло, запретить, множественное наследование — зло, запретить.

в ФП тоже строят сложные схемы, но наследования там нет.

> Я не говорю, что эти утверждения не верны. У меня вопрос другой, зачем самому себе искусственно ограничивать возможности моделирования предметной области?

Это вопрос из разряда «зачем делать модификаторы доступа public, зачем самому себе искусственно ограничивать возможности» :)

sentyaev 26 сентября 2018 в 13:05

0

> Дальше, порассуждав о причинах подобного, было бы проще обсуждать отсутствие наследования только реализации.

Я уже ниже писал, что в Java это уже завезли, как дефолтную реализацию метода интерфейса, в C# тоже скоро завезут, в Python и C++ и так есть. Есть Mixins и Traits, есть Extension methods.

> в ФП тоже строят сложные схемы, но наследования там нет.

Так там и классов нет (если говорить о чисто функциональных языках). Мы же все-таки о ООП языках.

0xd34df00d 29 сентября 2018 в 02:17

0

> Так там и классов нет (если говорить о чисто функциональных языках).

Есть, и среди них есть даже некоторое подобие наследования.

InterceptorTSK 8 октября 2018 в 14:16

0

Вот причем тут ооп-языки и реализация интерфейса? Реализация интерфейса к ооп вообще не имеет никакого отношения. И да, дефолтная реализация — это скорее всего совершенно не нужный синтаксический сахар. Ну реализован дефолтно как-то там метод интерфейса. Ну в классе-наследнике если нет реализации — скопируется реализация и все. А оно надо вообще? Это опять костыль какой-то…

У вас есть базовый класс A, к нему пристегнут интерфейс iA, у этого интерфейса метод defm(). Есть производный класс Б, т.е. Б:A, и соответственно Б:A,iA, и т.к. деф.метод скопируется полностью, то будет Б.defm(). У другого производного В от А будет такая же история, т.е. B.defm(). Вызовы всех трех методов как раз таки нарушают ооп. Потому что метод на самом деле один, и он A.defm(), а все остальное — его копии. И зачем это копирование метода? Что бы его вызывать из Б и В уже щяс достаточно Б.defm() и B.defm(). Могу предположить для «инлайна» реализации defm() в классы-наследники — но зачем? Тут уж наверное проще было бы завести какое-нить ключевое слово хоть то же «inline», т.е. если метод имеет инлайн, то он и заинлайнится в наследников, а не будет висеть только у родителя. Оно было бы понятно как и что. Но ооп-то тут причем? Если нынешняя реализация деф.метода — это костыль для может быть увеличения производительности, но это разрушает ооп. Ключевое слово было бы понятнее, и не разрушало бы ооп.

sentyaev 9 октября 2018 в 01:03

0

> Потому что метод на самом деле один, и он A.defm()

Метод один и он в iA, но не в A.

> И да, дефолтная реализация — это скорее всего совершенно не нужный синтаксический сахар.

Ваше мнение понятно, но видимо те кто делает ЯП с вами не согласны.

> Ну в классе-наследнике если нет реализации — скопируется реализация и все.

Какой ЯП вы имеете ввиду?

0xd34df00d 25 сентября 2018 в 20:26

0

> У меня вопрос другой, зачем самому себе искусственно ограничивать возможности моделирования предметной области?

Чем больше ограничений, тем больше шансов статически отвергнуть семантически некорректную программу.
Чем больше разумных ограничений, тем проще писать семантически корректные программы.

> Но если проект у вас не тривиальное CRUD приложение, то рано или поздно придется строить сложные иерархии, т.к. это будет проще, чем страдать от того, что кто-то решил, что наследование — зло, запретить, множественное наследование — зло, запретить.

Предположим, вы хотите представлять ноды в AST-дереве языка вроде плюсов.

Значимая часть нод имеет понятие глобального уникального идентификатора USR (например, объявление или определение функции, параметра, класса и всего такого), значимая часть нод имеет понятие USR, на который они ссылаются (например, выражение вида Class::type), часть нод имеет понятие модификатора доступа (private/public/protected для членов классов), часть нод имеет понятие списка детей (тип некоторых известен статически, некоторых — нет), и так далее.

Кроме того, реализации, например, первых двух (с USR) при этом одинаковы для всех нод, поэтому наследовать лишь интерфейс расточительно.

Кроме того, вы бы хотели параметризовать AST некоторым дополнительным полем, содержащим аннотации соответствующих проходов вашего компилятора/анализатора/etc — от места в исходнике (которое имеет смысл не для всех нод) до некоторых выведенных анализатором данных (что тоже имеет смысл не для всех нод). При этом, естественно, завязывать заранее типы для AST на возможные аннотации не хотелось бы. Ну и ещё хотелось бы проверять статически, что вы, скажем, тайпчекеру в рекурсивном вызове скормите только аннотированные им ноды.

Как тут построить иерархию классов?

sentyaev 26 сентября 2018 в 12:53

0

Вы почему-то из темы разработки приложений перешли к теме разработки компиляторов.

0xd34df00d 29 сентября 2018 в 02:18

0

А компилятор не приложение?

CRUD вам тривиально, компиляторы (на самом деле нет, аналог clang-tidy) слишком сложно. Вам не угодишь!

0xd34df00d 25 сентября 2018 в 20:17

0

Множественное наследование даже интерфейсов в языках с сабтайпингом ломает замкнутость сабтайпинга относительно объединений, а это имеет *эффекты*.

PsyHaSTe 25 сентября 2018 в 21:55

0

Эффекты насколько я понимаю просто позволяют декларировать некоторое множество «грязных» действий функции, разве нет? Если да, но немного не понимаю, причем тут это.
А еще не вижу, как ММ ломается в этом случае. У нас всегда есть статически известная функция, которая вызывается, либо виртуальная функция, которая тоже однозначно задается.

0xd34df00d 29 сентября 2018 в 02:21

0

Я про те эффекты, которые специальные, как в фильмах, когда что-нибудь взрывается.

Какой тип у выражения cond ? { x = 10, y = 20 } : { x = 30, z = 40 } ?

А какой тип у последнего выражения здесь?

```
interface I1 {};
interface I2 {};

class C1 implements I1, I2 {};
class C2 implements I1, I2 {};

cond ? new C1() : new C2();
```

PsyHaSTe 29 сентября 2018 в 11:25

0

Ну тип тут явно не выводится сам по себе. С тайпклассами было бы то же самое вроде.

0xd34df00d 29 сентября 2018 в 18:09

0

В смысле не выводится? Вы можете написать этот код (с обычными пританцовываниями про public static abstract singleton factory), и он соберётся, значит, статический тип у этого выражения есть. Более того, он будет равен чему-то вроде T where T extends I1 & I2 (я джаву не знаю, сорри за синтаксис).

А с тайпклассами оно просто не соберётся.

```
Prelude> class I1 a where
Prelude> class I2 a where
Prelude>
Prelude> data C1 = C1
Prelude> data C2 = C2
Prelude>
Prelude> instance I1 C1 where
Prelude> instance I2 C1 where
Prelude> instance I1 C2 where
Prelude> instance I2 C2 where
Prelude>
Prelude> f x = if x then C1 else C2

<interactive>:12:25: error:
    • Couldn't match expected type 'C1' with actual type 'C2'
    • In the expression: C2
      In the expression: if x then C1 else C2
      In an equation for 'f': f x = if x then C1 else C2
```

PsyHaSTe 29 сентября 2018 в 19:46

0

Я про то, что нет однозначного типа для C1 | C2, а возвращать тип-сумму неявно языки не позволяют, что хорошо.

0xd34df00d 30 сентября 2018 в 19:14

0

> Я про то, что нет однозначного типа для C1 | C2

Ровно потому, что есть множественное наследование. Если бы его не было, то такой тип существовал бы всегда.

PsyHaSTe 30 сентября 2018 в 20:36

0

Так нет же, с тайпклассами мы можем объявить свой трейт для типов C1 и C2 и теперь он тоже должен в него резолвится. В случае обычного наследования у нас по месту объявления классов уже понятно, какие супертипы у них могут быть общие.

Druu 2 октября 2018 в 10:49

0

> Какой тип у выражения cond? { x = 10, y = 20 }: { x = 30, z = 40 }?

В тс будет { x: number, y: number, z?: undefined } | { x: number, z: number, y?: undefined }.

> Вы можете написать этот код

Да вообще-то нет, типов-объединений ведь нету в джавах. Можно кастануть к Object обе ветки и будет результат Object или чот такое. Может, в каких-то случаях каст к супертипу будет сам по себе.

> Более того, он будет равен чему-то вроде T where T extends I1 & I2

Неправильно. Он будет T where T extends (I1 & I2) | (I1 & I2) и уже потом сократится по идемпотентности |. Но если | в языке нет то и не получится результата, если только не закостылить вот один этот конкретный кейз.

VolCh 26 сентября 2018 в 10:04

0

А можно пример когда множественное наследование нужно? Именно нужно, а не типа «в этом классе у нас код повторяется и из этого класса, и из этого — нужно отнаследоваться от обоих». Множественное наследование очень яркий признак нарушения принципа единственной ответственности во всех случаях, когда я его реально встречал. И никакой необходимости его нарушать я не видел.

SirEdvin 26 сентября 2018 в 11:58

0

А как же быть с миксинами? Когда в класс нужно добавить какой-то функционал.

0xd34df00d 29 сентября 2018 в 02:22

0

В плюсах в этом случае обычно костыляют CRTP (и это по ряду причин предпочитается над множественным наследованием реализации).

Ну и можно поговорить о том, что миксины тоже зло.

sentyaev 26 сентября 2018 в 12:37

0

Получается я спрашивал почему множественное наследование — зло, а вы теперь меня спрашиваете: А зачем оно вам нужно?

Как пример рассуждения о множественном наследовании: blog.cleancoder.com/uncle-bob/2015/01/08/InterfaceConsideredHarmful.html

А вообще, множественное наследование интерфейсов — это тоже множественное наследование. Да и в Java, например, уже появилась реализация метода по умолчанию в интерфейсе, в C# тоже обещают добавить. Эта штука конечно имеет ограничения, но по сути интерфейс теперь может иметь реализацию по умолчанию. Получается, что даже создатели языков программирования потихоньку движутся в направлении добавления множественного наследования в мэинстримные языки. Python имеет поддержку множественного наследования и она очень грамотно

сделана используя C3 linearization (https://en.wikipedia.org/wiki/C3_linearization)

Если взять топ ООП языки такие как Java, Python, C++, C#, то в первых трех множественное наследование реализации на данный момент поддерживается полностью, либо с ограничениями (Java), а в C# скоро добавят.

Зачем рассуждать с позиции нужно/ненужно, это один из способов моделирования. Плохой или хороший, сложный или нет — это уже другой вопрос.

popov654 21 сентября 2018 в 16:44

0

> А ещё придётся платить за каждый веб-сайт

Кому, посетителю? Заказчик и так платит.

progmal 21 сентября 2018 в 17:13

0

Можно написать более оптимизированный софт и меньшем количестом разработчиков. «А в реальном мире пользователю важнее получить приложение за 1$ которое будет работать» но будет работать а не вырубать от перезгрузки и нехватки оперативки телевизор или телефон, в котором всего памяти 16Г = занято 1г + свободно 300мб.

crystallize 24 сентября 2018 в 09:17

–1

Вы задним числом поменяли вопрос и написали удобный ответ.
Не надо нанимать никого нового, нужно уволить прежних дураков вместе с менеджерами и нанять небольшое количество, но квалифицированных.

Osnovjansky 25 сентября 2018 в 13:58

+1

«нанять небольшое количество, но квалифицированных»
Они у вас, как мухи, самозарождаются из воздуха? Или, всё-таки, должны получить свою квалификацию, поработав на каких-то проектах перед этим?
Тогда какие-то проекты должны быть написаны обычными слабоквалифицированными программистами

crystallize 27 сентября 2018 в 08:43

0

Для этого существуют учебные заведения и любительские проекты, не надо всё это тащить в софт которым будут не только пользоваться массово, но ещё и деньги за это платить.

crystallize 29 сентября 2018 в 19:19

0

Вообще конечно это совершенно несусветное заявление, что

> перед тем как начать делать свою работу качественно, работник имеет право некоторое время выпускать брак, пока не научится

. В любой другой профессии это подвело бы человека если не под статью, то по крайней мере под заключение о профнепригодности с последующим увольнением. Но ИТ-шникам всё сходит с рук, толи потому что никто не понимает что возложенная на них ответственность не меньше чем в любой другой профессии, толи потому что никто в принципе не понимает чем они занимаются.

PsyHaSTe 29 сентября 2018 в 20:05

0

> В любой другой профессии это подвело бы человека если не под статью, то по крайней мере под заключение о профнепригодности с последующим увольнением

Нет, в любом другой профессии точно такое же отношение к новичкам.

crystallize 30 сентября 2018 в 18:44

0

Произведенную ими некондицию продают под видом и по цене качественного товара?

PsyHaSTe 30 сентября 2018 в 20:34

0

Да

Osnovjansky 29 сентября 2018 в 20:32

0

А то что бывают токари разных разрядов, до 6-ого, если мне память не изменяют, и аттестовываются и получают они их только получив опыт работы, вы в курсе?

Низкоквалифицированный = умеет делать работы низкой сложности
Квалифицированный = обычно имеют в виду «имеет высокую квалификацию и может делать работы высокой сложности» (здесь я не про токарей, а опять про программистов)

А проблема брака везде решается выходным контролем — даже квалифицированные могут ошибаться + возможен брак материалов (а здесь я уже про любое производство, включая программы)

VolCh 30 сентября 2018 в 18:04

0

Ещё один принцип решения проблемы брака — допуски до работ. Грубо, есть работы к которым разрешено допускать только токарей 6-го разряда, может 5-го, но с большим опытом работы.

crystallize 30 сентября 2018 в 18:49

0

Низкоквалифицированный труд это не значит гнать брак. А нормальный «выходной контроль» как раз не пропустит всю дичь упомянутую в статье.

sergeperovsky 24 сентября 2018 в 13:37

+4

Штат разработчиков расширят? Все, кто делает продукт для массовой аудитории знают, что вложения в рекламу и маркетинг дают большую отдачу, чем вложения в разработку.
Проблема не в стоимости, проблема в особенностях рынка. Нужно успеть первым. Пользователи НЕ ЛЮБЯТ ПЕРЕУЧИВАТЬСЯ. Ваше приложение может быть гораздо лучше. Но если кто-то уже занял нишу, потеснить его непросто. Поэтому скорость разработки стала для многих видов ПО самым главным. А из этих областей идея верховенства скорости над качеством распространилась на всю отрасль.
Я начинал не 15, а страшно представить, 50 лет назад. Алгол, Фортран и программирование штырьками на Промини со 100 ячейками памяти. В те времена программисты были учеными. Не было еще инженерной дисциплины программирования. Очевидные сейчас приемы придумывались у меня на глазах. Зачем исторический экскурс? В книге 1961-го года издания доказывалось, что языки высокого уровня не имеют экономического смысла: машинное время, потраченное на трансляцию стоит много дороже времени, которое экономит на этом программист. Основной проблемой почти любого программиста было поместить в компьютер задачу, которая ни по времени, ни по памяти помещаться туда не может.
Привычку экономить биты и такты я потом годами по капле из себя выдавливал. Экономное программирование нужно для критически важных фрагментов. В остальных случаях оно замедляет разработку. Другое дело, что после ученых пришли инженеры, а потом кодеры. И отличить критически важные фрагменты от остальных они не могут, и экономить ресурсы совсем разучились. Отсюда и описанные в статье проблемы.

crystallize 29 сентября 2018 в 19:24

0

> Пользователи НЕ ЛЮБЯТ ПЕРЕУЧИВАТЬСЯ

И тем не менее в последние 8 лет практический каждое приложение или веб-сайт поменяли дизайн, и скорее всего не однажды.

sergeperovsky 30 сентября 2018 в 03:44

0

Меняют дизайн, в двух случаях.
1.Когда не могут добавить полезный функционал, а нужно как-то обосновать полезность перехода с версии 14 на версию 15. Тогда привлекается группа дизайнеров и продавцы начинают расхваливать их творчество.
2.Когда новый функционал требует нового интерфейса. Например, для программы расчета трубопроводной сети добавили к табличному интерфейсу картографический. Стало возможным вводить огромные сети с минимальными ошибками, автоматически определять длину участков, строить пьезометрические графики и продольные профили, выводить на карту результаты расчетов и т.д.
В обоих случаях пользователи ворчат. Но во втором они видят пользу.

khim 30 сентября 2018 в 10:23

+1

Вашими бы словами… Вот какой новый функционал нельзя было воплотить в GMail образца 2004го года? Или в MS Office 95го?

Я вас разочарую: в 9 случаях из 10 новый дизайн обновляется по одной и только одной причине. Называется: «онлитеги решили, что пора обновить дизайн».

Как правило обосновывается это тем, что дизайн устарел и уже «не модный». Или тем, что он «эклектичный и если вы всё переделаем, то всё будет в одном стиле» (при этом не объясняется почему 10 предыдущих переделок дизайна не привели ни к чему подобному).

Вот в прошлом веке зачем меняли дизайн — было легче понять. Скажем Turbo Pascal 1-3, 4-5, и 6-7 сделаны по разному из-за чисто технических ограничений: Turbo Pascal 7 на машиках, где

работал Turbo Pascal 5 «не взлетит», как и Turbo Pascal 5 на 8-битных машинках, где работал Turbo Pascal 1. Хотя переход с Turbo Pascal 5 на Turbo Pascal 6 в этом смысле уже сомнителен: в принципе ничто не мешало добавить перекрывающиеся окна и в дизайн Turbo Pascal 5.5, так что этот переход несколько ближе к пресловутой «ленте», однако он почти не затронул «моторные навыки», только визуально чуть-чуть всё перекрасили, так что, в принципе, большим «разрывом шаблона» он не был.

Но вот в XXI веке, после того, как компьютеры стали достаточно мощными, чтобы поддержать многооконный GUI-интерфейс… я вот сходу не назову ни одной фичи ни в одном продукте, которую нельзя было бы вписать в старый дизайн… вы вот можете показать хоть один редизайн, который реально принёс возможность что-то принципиально новое делать, чего нельзя было делать в старой — хоть в одной **реальной** программе (а не в гипотетической программа рассчёта трубопроводов)?



Am0ralist 30 сентября 2018 в 12:49

0

Базис Мебельщик 7 и Базис Мебельщик 8. Хотя поймет только тот, кто работал в обеих. Так что аргумент «хоть в одной реальной программе» — считайте опровергнутым) Хоть в одной — но будет.

khim 30 сентября 2018 в 18:45

0

Я с ними не работал, но на YouTube есть видео. Из которых видно, что как редизайна-то там и нету! Вот как раз подход к редактированию — сильно изменился и можно обсуждать к лучшему или к худшему — но «нового стильного, молодёжного»-дизайна там нет. Скорее наоборот — видны попытки уменьшить отличия.



Am0ralist 30 сентября 2018 в 20:44

0

Ну не скажите, оформление с а-ля софт 90-х изменился на а-ля начало нулевых) Ну и так по мелочам изменения были. Они очень консервативны, да.
Думаю, до плиток они еще лет через 5 доберутся.
А так, да, внутренний редизайн взаимодействия пользователя с программой — изменили капитально. Да, там осталась возможность работать по старому, но новый стиль взаимодействия был в разы продуктивнее.

sergeperovsky 1 октября 2018 в 02:27

0

Программа для проектирования и эксплуатации трубопроводных сетей не гипотетическая. Это совершенно конкретный пример: 30 лет у программы был табличный интерфейс, а добавление картографии позволило резко расширить функционал и целевую аудиторию. Многие специализированные CAD-системы получили 3D окно, что также упростило пользование и позволило решать ряд дополнительных задач.
Много раз обруганный переход на «ленточное меню» для некоторых программ имел смысл. Когда количество используемых команд велико, высокая вложенность стандартного меню замедляет работу. Если экраны стали достаточно большими, почему бы не сделать как можно больше команд доступными в один клик?



Alexeyslav 1 октября 2018 в 09:24

0

А там уже другая крайность — попробуй найди иконку нужной команды среди сотни похожих… особенно когда они переезжают с места на место как живые.
Хоткеи? нет, не знаем…
Большой экран берут чтобы в него поместилось больше редактируемого документа, а не команд.

sergeperovsky 1 октября 2018 в 19:20

0

Дублирование меню и хоткеев появилось не зря. Разным людям удобно работать по разному. Кто-то помнит только ctrl-C и ctrl-V и то предпочитает вызывать контекстное меню, а кто-то помнит два десятка кнопок.
Вообще, настройка интерфейса для действительно сложных программ дело разумное. Вот только нужно иметь возможность свои настройки таскать с собой в маленьком файле, а то, действительно, приходится искать в чужих настройках нужную команду среди сотен.
Впрочем, в основном мы согласны: чаще интерфейс меняют ради изменений, а не ради практической пользы.



Alexeyslav 2 октября 2018 в 08:38

0

Если часто работаешь с какой-то функцией и нужен быстрый доступ не сложно назначить свой хоткей удобный для пользователя. Другое дело, люди почему-то не хотят этого делать… а так же не хотят использовать мощную систему макросов.

VolCh 2 октября 2018 в 10:16

0

Во-первых, не всем не сложно, удобного интерфейса назначения хоткеев нет в, как минимум, винде и гноме. Главный недостаток имеющихся — просто нет списка уже имеющихся для данного контекста (включая ОС, оконный менеджер и т. п.)
Во-вторых, некоторые сценарии работы вообще не подразумевают использование клавиатуры или её использование исключительно для ввода текстов, вплоть до переключения между полями ввода в форме мышью, а не, например, табом.

Про макросы вообще молчу, «мощные системы» больше похожи на язык программирования.



Alexeyslav 2 октября 2018 в 15:14

0

Программируемость там является дополнительным бонусом, но не обязательна. Ты можешь не видеть текст макроса от создания и до его смерти после многократного использования.

APXUB 1 октября 2018 в 11:12

0

С чего вдруг $50000? Вы считаете наняв группу с дополнительных, скажем, 140 человек, они за пару лет заработают $50 триллионов? Если скромно учесть моб.ОС на 1 млрд единиц.

elCreator 1 октября 2018 в 11:12

0

тормоза как раз из-за расширения штата дешевыми неквалифицированными разработчиками. Тот же Ангуляр, да и популярные фреймворки вообще взлетели, т.к. дали возможность бизнесу распараллелить простые задачи между людьми, умственные способности которых без фреймворка, делающего программирование простым, не позволили бы решить задачу.

holy_desman 20 сентября 2018 в 15:10

+8

Вот построим ИИ — программиста, он нам быстренько всё оптимизирует.

dukei 20 сентября 2018 в 15:14

+7

Не, не быстренько, он же будет плохо написан. Сначала ему придется себя оптимизировать. И это займет бесконечное время, так что на ПО у него времени уже не останется.

Akon32 20 сентября 2018 в 15:49

+5

За бесконечное время и (особенно!) бесконечные деньги мы и сами сможем всё оптимизировать.

AstarothAst 20 сентября 2018 в 17:15

+7

Позволю себе процитировать классика:

> Дейв Вудбери и Джон Хэнсен, неуклюже в своих скафандрах, с волнением наблюдали, как огромная клеть медленно отделяется от транспортного корабля и входит в шлюз для перехода в другую атмосферу. Почти год провели они на космической станции A-5, и им, понятное дело, осточертели прокочушие фильтрационные установки, протекающие резервуары с гидропоникой, генераторы воздуха, которые надсадно гудели, а иногда и просто выходили из строя.
> — Все разваливается, — скорбно вздыхал Вудбери, — потому что все это мы сами же и собирали.
> — Следуя инструкциям, — добавлял Хэнсен, — составленным каким-то идиотом.
> Основания для жалоб, несомненно, были. На космическом корабле самое дефицитное — это место, отводимое для груза, потому-то все оборудование, компактно уложенное, приходилось доставлять на станцию в разобранном виде. Все приборы и установки приходилось собирать на самой станции собственными руками, пользуясь явно не теми инструментами и следуя невнятным и пространным инструкциям по сборке.
> Вудбери старательно записал все жалобы, Хэнсен снабдил их соответствующими эпитетами, и официальная просьба об оказании в создавшейся ситуации срочной помощи отправилась на Землю.
> И Земля ответила. Был сконструирован специальный робот с позитронным мозгом, напичканным знаниями о том, как собрать любой мыслимый механизм.
> Этот-то робот и находился сейчас в разгружающейся клети. Вудбери нервно задрожал, когда створки шлюза наконец сомкнулись за ней.
> — Первым делом, — громыкнул Вудбери, — пусть он разберет и вновь соберет все приборы на кухне и настроит автомат для поджаривания бифштексов, чтобы они у нас выходили с кровью, а не подгорали.
> Они вошли в станцию и принялись осторожно обрабатывать клеть демолекуляризаторами, чтобы удостовериться, что не пропадает ни один атом их выполненного на заказ робота-сборщика.
> Клеть раскрылась!
> Внутри лежали пятьсот ящиков с отдельными узлами… и пачка машинописных листов со смазанным текстом.

Hardcoin 20 сентября 2018 в 15:14

+2

> Вы бы купили машину с расходом 100 литров на 100 километров? Как насчёт 1000 литров?

Разумеется. Ведь машина на десять литров требует большой штат инженеров для её поддержки, к тому

же ценник на неё в десятеро выше.

Alexey2005 20 сентября 2018 в 16:45

+4

Идея для бизнес-модели производителей авто: продавать машину в 100 раз ниже себестоимости, но сделать так, чтобы она могла ездить только на фирменном бензине, который стоит в 100 раз больше обычного.

aknew 20 сентября 2018 в 16:49

+20

Что-то ваша машина сильно некоторые принтеры напоминает по такому описанию

engine9 20 сентября 2018 в 18:47

+2

Сейчас уже и в аккумуляторы фотокамер чипы встраивают.

nikolayv81 21 сентября 2018 в 17:44

0

> 10 лет уж точно, особенно японцы преуспели (panasonic) а ещё они з.ч. (платы) выдают только оф. сервисам в обмен на сданные вышедшие из строя по цене 30-40% от стоимости устройства в момент старта продаж :)

s-kozlov 28 сентября 2018 в 09:23

0

Они могут заряжаться только фирменными электронами?

Osnovjansky 28 сентября 2018 в 11:08

0

Как ни странно, тут есть и очевидное техническое обоснование.
1. Массовый продукт должен работать в руках любого идиота
2. Юзвери будут ставить самые дешевые аккумуляторы
3. У дешевых аккумуляторов высокое внутреннее сопротивление — камера будет постоянно просаживать напряжение аккумулятора — в результате, перезагружаться.
Через несколько секунд/минут напряжение на аккумуляторе будет восстанавливаться и т.д. по кругу
4. Юзвери будут волить — камера говно — и с этим придется разбираться техподдержке и маркетологам

engine9 29 сентября 2018 в 10:12

0

Разумно, но со старыми камерами такого не наблюдалось. Мне кажется, это способ ограничить жизненный цикл устройства и выдоить лишнюю копейку с пользователя.

Alexeyslav 1 октября 2018 в 09:26

0

Полагаю, что и потребление энергии у старых камер было небольшим. А новые камеры, с мощным процессором и матрицей высокого разрешения требуют много энергии для работы, скорей всего ещё и кратковременным но мощным потреблением.

springimport 20 сентября 2018 в 17:18

+8

Оглянитесь на рынок консолей.

UrbanRider 20 сентября 2018 в 17:46

+6

+++Выпускают типо эксклюзив. Хотя архитектура сейчас на обеих топ консолях x86 и ничего не мешает порту на PC. Но нет, купите нашу не очень дорогую консоль и заплатите овердофига денег за игру, которую ты пройдешь за 2 дня.

Tsimur_S 21 сентября 2018 в 15:31

0

если ничего не мешает то почему провалился порт batman arkham city(на самом деле я думаю он не единственный но самый известный)? фризы, падение фпс, тысячи рефандов в стиме что аж пришлось заморозить продажи в стиме по причине проблем с перфомансом. Может это все же не самая простая операшка?

UrbanRider 21 сентября 2018 в 15:40

+2

Ну не самая сложная, т.к. архитектура не меняется. Консоли сейчас не мощнее среднестатистического компа. Где-то на уровне среднего игрового ПК.

То что оно там все тормозило опять же отсылает нас к прочтению статьи и тотальному негодованию.

mayorovp 21 сентября 2018 в 17:00

0

Если код нормальный — то довольно простая. А если у NaN брать младшие 16 бит и что-то с ними делать...

sergeperovsky 1 октября 2018 в 02:31

0

Машины на водороде не принесли прибыли.

DrPass 20 сентября 2018 в 17:41

0

Эта аналогия не верна ни в отношении автомобилей, ни в отношении софта.

VBKesha 20 сентября 2018 в 15:14

+16

> Как инженеры, мы можем и должны, и сделаем лучше.

Как инженеры то да, можем, должны, сделаем. Но решения принимают не инженеры, новые «красивые и удобные» дизайны придумывают и одобряют тоже не инженеры, и сверху спускают планы за сколько что нужно выполнить тоже не инженеры. ИМХО инженеры уже давно исполнительное звено, и просто делают то что от них требуют. Время инженеров осталось в начале 2000х, теперь время менеджеров…

muove 20 сентября 2018 в 15:44

+14

Кто как, по моему опыту большинство раздолбайства в проектах было допущено нами самими а не ВЕЛИКИМИ и УЖАСНЫМИ менеджерами. И время у нас было, и деньги, но мы такие.
ЗЫ совсем не обобщаю, все только про меня самого

Mabusius 20 сентября 2018 в 17:17

+5

Так менеджеры для того и приставлены, чтоб контроливать халтуру. А они не контролируют. Так что все равно менеджеры…

kreatr 20 сентября 2018 в 15:18

+6

кажется местами похоже на стиль Линуса :)

arthi7471 20 сентября 2018 в 18:32

+9

Линус как раз сейчас в отпуске.

dartraiden 21 сентября 2018 в 02:00

0

Линус, в какой-то мере ответственен за описанное в статье. Ведь в основе Android лежит ядро Linux, а в ядро в каждом релизе включается всё больше и больше всякого. По сути, в ядро готовы принять что угодно, если это не ломает существующие механизмы и есть люди, которые готовые это поддерживать.

khim 21 сентября 2018 в 02:30

+5

> По сути, в ядро готовы принять что угодно, если это не ломает существующие механизмы и есть люди, которые готовые это поддерживать.

Вот уж чего нет, того нет. Любая фича, которую добавляют в ядро, в первую очередь оценивается на тему: а мы точно получим от этого существенный выигрыш, стоящий усложнения ядра? В результате какой-нибудь overlayfs включили в ядро в 2014м году. Притом что первая версия (ну… плюс-минус…) под именем IFS — вот она… 20 лет, однако…

А что включается «без вопросов» — так это драйвера… но тут всё просто: если у вас нет соотвествующей железяки — то и от наличия драйвера вам не горячо и не холодно…



Am0ralist 21 сентября 2018 в 10:49

0

Знаете, я помню, читал вроде как в ядро включили поддержку модульного смартфона от гугла… сразу после того, как проект этого смарфтона был *приостановлен*.
Был немного удивлен, так сказать.

khim 21 сентября 2018 в 14:20

0

А я — нет. Там и поддержка каких-то монстров от SGI, которых выпущено было 100 штук была. И всяких исторических компьютеров от IBM и много чего ещё.

Ваша поддержка сколько занимала в памяти в случае отсутствия такого телефона? 100 мегабайт? Или «нисколько»?



Am0ralist 21 сентября 2018 в 14:43

0

Да я даже не знаю, что в нее включалось, просто вспомнилось к словам «любая фича, которую добавляют в ядро, в первую очередь оценивается на тему: а мы точно получим от этого существенный выигрыш, стоящий усложнения ядра?» и то, что фича по сути как раз вроде никому стала не нужна…

mayorovp 21 сентября 2018 в 15:13

0

Так и ядро не усложнилось же.

Fracta1L 21 сентября 2018 в 08:17

+4

Вы ещё скажите, что в тормозах Андроида ядро виновато XDD



0xd34df00d 21 сентября 2018 в 22:18

0

Только там ненужные вещи можно, ну, выключать при сборке.

Laney1 20 сентября 2018 в 15:30

+4

помню, как-то попытался собрать проект на node.js, который последний раз обновлялся года за 3 до этого. Получил огромную кучу нетривиальных ошибок, даже со старой версией ноды, и быстро расстался с этой глупой идеей.

Еще недавно читал про адок с USB Audio в новых смартфонах. Ни черта ни с чем не совместимо, зато отказались от "устаревшего" джека.

Авторам Go все эти годы приходится иметь иметь дело с обиженными экспертами, которым не хватает в языке дженериков, концептов и прочей мути. О том, что введение всего этого раздует спецификацию и сложность языка в несколько раз, эксперты естественно не задумываются. Как и о том, зачем оно вообще нужно.

Автор статьи возможно даже не осознает, насколько он прав.



Suvitruf 20 сентября 2018 в 15:40

+2

Вот у меня наоборот, проекты на ноде многолетней давности собрались на самой последней версии nodejs без единой ошибки.

mkshma 20 сентября 2018 в 16:21

+15

> Авторам Go все эти годы приходится иметь иметь дело с обиженными экспертами, которым не хватает в языке дженериков, концептов и прочей мути

Потому что без этого мутью становится сам код. Либо пиши версию функции под нужный тип каждый раз, либо прокидывай через интерфейс и кастуй… Ну бред же.

Xandrmoro 20 сентября 2018 в 17:03

+14

> которым не хватает в языке дженериков, концептов и прочей мути

Дайошь копипасту, ага.



PsyHaSTe 20 сентября 2018 в 20:55

0

> Rust пока что выглядит, как чисто сисадминский язык :)
> Дженерики, по-хорошему, это задача для искусственного интеллекта, а у меня его нету :)

(c)

sand14 20 сентября 2018 в 18:01

+1

> Авторам Go все эти годы приходится иметь иметь дело с обиженными экспертами, которым не хватает в языке дженериков, концептов и прочей мути.

Дженериков действительно не хватает (если их в консервативную джаву добавили…), и вроде обсуждается, чтобы добавить их в Go
А вот с остальным вопросы — наверное, действительно не нужно, т.к. идеи и реализации спорные, и будут противоречить основной идее Go как "простого" языка.



DarthVictor 20 сентября 2018 в 18:52

+6

> Авторам Go все эти годы приходится иметь иметь дело с обиженными экспертами, которым не хватает в языке дженериков, концептов и прочей мути.

Дженерики и нормальная система типов скорее и продукт ускоряют и код упрощают. Единственное что они увеличивают — порог входа.

PsyHaSTe 20 сентября 2018 в 20:57

0

Если посмотреть то, как они их сделали — такое ощущение, что лучше бы вообще не делали. Раньше у го была хотя бы ниша «пишу не читая, и оно работает из коробки», а теперь вообще что-то непонятное.

Bonart 20 сентября 2018 в 21:40

0

Наверное, именно поэтому во второй версии го есть генерики и обработка ошибок.

JekaMas 21 сентября 2018 в 09:25

+1

Судя по draft фич golang, то "раздует" — это 4 новых слова в спеке языка. Вы о чем-то не том...

Laney1 21 сентября 2018 в 10:18

0

задумывались о том, что последует за этими 4-мя словами?

- переписывание и значительное усложнение препроцессора языка. Как показывает практика (новости типа "в фейсбуке написали собственный препроцессор C++, работающий в 2 раза быстрее стандартного") — нифига не тривиальная задача.
- полное переписывание стандартной библиотеки. Несколько лет работы как минимум.
- "ой, а у меня тут дженерик не компилируеться, компилятор ругаеться непонятными словами!". Придется вводить в язык концепты или что-то аналогичное. Еще несколько лет работы.
- "ой, а тут надо писать тип длиной полтора километра, мне это не нравиться!". Ага, языку определенно не хватает продвинутой системы вывода типов.

Дальше продолжать? Кто будет все это делать, Тейлор с Пайком? А как насчет тулинга и прочих популярных библиотек — они что, сами себя перепишут? Или вы хотите повторения истории с переходом на Python 3?

А нафига вообще все это в Go, если есть Java? Как тут правильно сказали, уже давно с дженериками и полным фаршем. Или Rust — вернуть туда указатель Gc, и все дела. Или D — кстати, где он? Или куча других языков со всеми "современными" фичами.

JekaMas 21 сентября 2018 в 11:52

0

Ваше утверждение было об «экспоненциальном» увеличении спеки. И мой комментарий именно об этом.
Зачем вы переподите дискуссию в область поддержки и развития кода? Я эту тему не затрагивал.

Laney1 21 сентября 2018 в 12:09

0

> Зачем вы переподите дискуссию в область поддержки и развития кода?

затем, что любая спецификация бессмысленна без кода, который ее реализует. А что произойдет с кодом, я кратко пояснил. И именно это я и имел ввиду.

И да, слова "экспоненциальный" я нигде не употреблял, это вы что-то выдумали

JekaMas 21 сентября 2018 в 12:13

0

поэтому в кавычках.
вы использовали «в несколько раз» — хорошо бы показать, что добавление 4х слов в спеку сделает это «несколько раз». ну или это ваши фантазии.

p.s. ещё полезное упражнение попробовать официальный путь generics через генераторы и пожить с поддержкой этого… годик. мнение резко меняется.


0xd34df00d 21 сентября 2018 в 22:26

+1

В C++ не дженерики, в C++ темплейты, тайпчекающиеся после мономорфизации, а не до (и концепты пытаются чинить это). Тайпчекайте до мономорфизации как тайпклассы в хаскеле или трейты в расте, и будет всё хорошо.

VolCh 22 сентября 2018 в 22:18

0

У экосистем джавы и гоу очень разные подходы. Можно сказать несовместимые.


solver 20 сентября 2018 в 15:32

+7

В целом правильно всё. Но автор передергивает конечно)

> В браузерах накопилось столько пограничных ситуаций и исторических прецедентов, что никто не осмелится писать движок с нуля.

А как же Firefox? Они не просто написали движок с нуля. Но даже рахработали новый более безопасный язык для этого…

tangro 20 сентября 2018 в 15:36

+7

А помните была такая компания Опера, пилила свой движок, гордилась этим. А потом тихо сдалась, призналась, что не тянет. И того всех мощностей человечества на сегодняшний день хватает на 3±1 движка браузера (и то же самое и с ОС, движками БД и т.д.)


Alexeyslav 20 сентября 2018 в 16:32

+4

Не мощностей а денежных средств.
Проблема не в количестве погромистов, а в их качестве. И в качестве архитекторов, которые этим заправляют…

Lexicon 20 сентября 2018 в 22:06

+3

Особая ирония — современные методологии управления.

Начальника нет, ответственность общая, все равны, все компетентны, все архитектора. Тимлидами становятся номинально, не неся за собой никаких реальных навыков руководства.

Блоги компаний "как мы управляем проектами/как мы делаем собеседования" множатся в геометрической прогрессии, но, похоже и до единства и до здравого смысла еще далеко.

На собеседованиях все такое же ощущение, будто основная обязанность тимлида — владеть Jira, а никак не умирающее желание пихать(порой совершенно бесполезные) задачки превращает собеседования в игру на случайность, в которой не выигрывает никто.

Конечно мир, где все люди равны одинаковы может выглядеть интересно, но ведь это не реальный мир. Задумываешься — "может мне кажется", но когда, скажем, на мобильном хабре(без обиды, хабр, ты исправляешься) неприлично долго висит кнопка "проголосовать", бесполезная для ~95% аудитории сайта, а прокрутка комментариев политических статей позволяет сгибать пальцем металлические пруты, это ощущение резко пропадает.

VolCh 22 сентября 2018 в 22:34

0

Так кнопка-то не работает: «что-то пошло не так»


ImKremen 20 сентября 2018 в 21:50

+2

А вы хотя бы спецификации W3C возьмите.
Это огромное, сложное и недоработаное ТЗ с большущими легаси проблемами в архитектуре. Самое интересное что в его разработке активно участвует не так и много людей, ибо порог вхождения также огромен.
Полностью переписать с нуля, согласовать и оптимизировать такую систему невероятно сложно.


Kant8 21 сентября 2018 в 11:24

0

Только сдалась она не потому, что не тянет, а потому что корпорация добра изо всех сил убивала ее по юзерагенту.


nikolayv81 21 сентября 2018 в 17:52

+1

Так причина то на поверхности, другая компания получив некоторую долю начала добавлять расширения на любой чих, без которых ломалось отображение в Opera, а т.к. ресурсы были несопоставимы(причём не только финансовые) то результат очевиден.


Massacre 20 сентября 2018 в 16:41

+5

Заодно убив поддержку нормальных аддонов, ну и раздув использование памяти до «высот» Хрома за счёт той же мультипроцессности.


Crandel 20 сентября 2018 в 21:49

+2

Хватит писать этот бред про память. Сколько я бы вкладок не открывал, никогда больше 3-4 гиг не сжирало(при 32 гигах на моем ноуте) Единственное исключение — вкладки с юдубом, которые постоянно выжирают память. В хроме же за час использования — 6-8 минимум, без юдуба

НЛО прилетело и опубликовало эту надпись здесь

VolCh 22 сентября 2018 в 22:35

+1

на 32-bit винде?)

khim 20 сентября 2018 в 21:58

+2

> Сколько я бы вкладок не открывал, никогда больше 3-4 гиг не сжирало(при 32 гигах на моем ноуте)

У меня только основной процесс 6 гиг занимает, хотя у меня вкладок не больше пары сотен сейчас открыто. Хотя некоторые вкладки открыты больше месяца назад, это да.

> Единственное исключение — вкладки с юдубом, которые постоянно выжирают память.

На рабочем компе я YouTube не открываю…

fukkit 21 сентября 2018 в 13:41

+1

> вкладок не больше пары сотен сейчас открыто

Вот это меня удивляет. У себя не могу насчитать более, ну максимум, 20.
Что вы делаете с таким количеством одновременно открытых вкладок? Читать их все одновременно, да и попеременно, вы едва ли способны. Держите на случай веерных отключений Интернета? Связь нестабильная? Закладками не пользуетесь?
Не первый раз вижу такие комментарии от людей, бывает еще пишут 500 вкладок. Теряюсь в догадках.
Просветите? (интересно)

khim 21 сентября 2018 в 14:26

0

Тематические окна браузера с 5-10 вкладками в каждой. В этом окне я смотрю какой корпус для Raspberry PI купить — тут 5 вкладок. В этом — комментирую статью Хабре, там открытые странички из неё и ссылки, которые ответы на неё пойдут. И так далее.

Когда «работа» закончена — окно закрывается. Arena Allocator. Но некоторые вещи могут делаться неделю или месяц.

fukkit 21 сентября 2018 в 16:34

0

Мне *кажется*, что они Вам действительно не нужны все **одновременно**, но некоторые обстоятельства делают неудобной их загрузку из ссылок по мере необходимости:
1) Неудобный интерфейс закладок
2) Проблема, как раз описанная в статье: сайты стали слишком жирными, ждать их загрузки из Интернета — никаких нервов не хватает.

Касательно доков/спеков, особую боль вызывают сайты с доками, грузящие тонны js (и не функционирующие без js!), в основном, для того, чтобы отобразить анимированный аккордеон оглавления.
В такие моменты хочется осыпать автора малоприличными выражениями из числа наиболее сочных, имеющихся в русском языке, но переживаешь молча, стиснув зубы.

VolCh 22 сентября 2018 в 22:40

0

Конечно не нужны, но закладки, хистори и всякие плагины ридитлэте неудобны, плагины навигации по вкладкам куда продвинутей. А главное к интерфейсу закладок я и требований чисто для себя не могу сформулировать, чтобы они не превращались во вкладки.

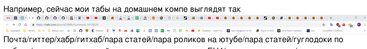


PsyHaSTe 21 сентября 2018 в 14:55

+5

Я лично в процессе написания какой-то фичи. Это могут быть вкладки с поисковыми запросами в гугл, документация, ответы с SO, примера на медиуме, обсуждение подводных камней на реддите... Всё это мне нужно в течение работы, т.к. я могу переключаться между ними, если во время реализации я понял, что мне нужно X, а я это видел в статье/ответе/... в одном из табов.

Например, сейчас мои табы на домашнем компе выглядят так



Почта/гиттер/хабр/гитхаб/пара статей/пара роликов на ютубе/пара статей/гуглодоки по работе/описание стратегий за некоторые страные в EU4/еще пара роликов/смешные картинки (открыл для просмотра комментариев)/снова EU4/снова картинки.

Даже на домашнем компе я могу открыть полсотни табов, очень простым способом: захожу на сайт картинок, если вижу интересное обсуждение, я его открываю в новом табе, и продолжаю листать общую ленту. Прокрутив ленту до того места, которое я вчера (или в прошлый раз) уже видел, я закрываю ленту и начинаю последовательно смотреть комментарии к понравившимся картинкам.



popov654 21 сентября 2018 в 19:02

0

Я так со статьями в Википедии часто делаю :)

> Всё это мне нужно в течение работы, т.к. я могу переключаться между ними, если во время реализации я понял, что мне нужно X, а я это видел в статье/ответе/... в одном из табов.

Но это жрёт дофига оперативки. Я в этом случае банально пользуюсь историей просмотра, благо, ссылки не успевают спуститься далеко вниз (у меня 3,75 Гб RAM всего доступно, а браузер — хром, и потребляет он немало).



PsyHaSTe 21 сентября 2018 в 19:23

0

> Но это жрёт дофига оперативки. Я в этом случае банально пользуюсь историей просмотра, благо, ссылки не успевают спуститься далеко вниз (у меня 3,75 Гб RAM всего доступно, а браузер — хром, и потребляет он немало).

Обойтись можно, но неудобно.
Поэтому я еще в 2014 году купил себе 16 гигов памяти и с тех пор не знаю бед. И SSD тоже. Вложился один раз, теперь серфинг не доставляет мучительных ощущений :)

popov654 21 сентября 2018 в 19:39

0

Тоже скоро планирую апгрейд) Вообще привычка — сильная штука. Я когда работаю на Win7 x64, у меня гораздо больше лимит (на самом деле, мне не удавалось выбрать больше 75-78 процентов даже на очень ресурсоёмкой операции анализа здорового бинарника в IDA Pro). Стоит, правда, всего 8 гигабайт. Но даже на этой ОС я всё так же открываю не больше 6-7 вкладок. Привык уже, ничего не могу с собой поделать)

С ностальгией теперь вспоминается 2012-ый год, когда я в Opera 11.64 на WinXP мог открыть 50-60 вкладок вк, и оно даже не тормозило. И вылетало только на 70-75 вкладке. С Хромом и обновлённым вк так уж не получается... Хотя хром потенциально может выдержать больше, но с моим объёмом памяти своппинг начинается уже на 30-35 вкладке, а дальше — ужасные тормоза при смене вкладок, листании фотографий и страх, что браузер вот-вот вылетит.

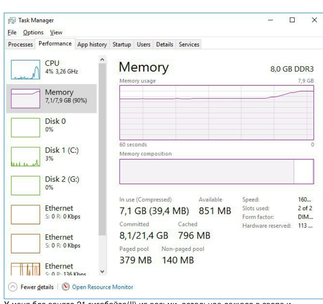
PsyHaSTe 21 сентября 2018 в 19:45

+1

Ну вот с той картиной что я выше показал хром у меня ест 1700 мегабайт.

Что касается 75% памяти — то винда всегда старается всю свободную память занять, ибо пользователю пофиг, 5% или 50% свободно, а вот если что-то будет запрефетчено и открыто на пару секунд быстрее, то ему будет счастье. Поэтому чем больше свободной памяти, тем больше всяких кэшей, и тем быстрее и комфортнее работа.

16 гигов занять сложно, а вот 12 у меня периодически получалось.
Например на прошлой работе мне дали убогий ноут с 8 гигами памяти, где мне приходилось постоянно запускать тяжелые докер-образы и несколько студий (дебажить 2-3 проекта параллельно, микросервисы, чтоб их). Так вот — скрин в студию



У меня бло занято 21 гигабайта(!!) из восьми, остальное лежало в свопе и постоянно шерстило диск, всё дико лагало и не вызывало желания что-то оптимизировать (всё равно не пройдет ревью, и меня спросят за зря списанное время), а вот желание кого-то убить появлялось…



popov654 21 сентября 2018 в 20:01

0

Ну, если запустить несколько виртуалок (Docker — это же виртуальные машины, если я правильно вас понял), то конечно можно очень быстро выбрать всю память. И IDE довольно много жрут, да.

Кстати, если запустить виртуалку в VMware, и в её настройках указать лимит памяти, превышающий мой размер RAM, и при этом гостевой ОС удастся достичь лимита по физической памяти хостовой ОС, то Win XP просто зависает намертво. Помогает только ребут :)

VolCh 22 сентября 2018 в 23:15

0

Докер в этом плане не виртуальная машина, а запускатель обычных процессов ОС с особыми флагами. Есть небольшой оверхед на управление, но сами процессы жрут ровно столько же памяти как с дефолтными опциями, а отдельное ядро не запускается.

Massacre 20 сентября 2018 в 23:55

0

Это, конечно, зависит и от вкладок, и от битности хрома и FF (у 64-битного, почему-то, процесс GPU в разы больше весит), но вот у меня Pale Moon если выжирает гиг, то что-то здесь не так (скорее всего, открылся Facebook). А тут вы 4 гига нормой считаете…

И да, вкладки с видео мало памяти жрать не могут, надо ж его где-то кэшировать…

Crandel 21 сентября 2018 в 00:12

0

> А тут вы 4 гига нормой считаете

Где я писал, что это норма? Я писал, что это максимум памяти, в который я упирался в фаерфоксе. Сейчас у меня открыт десяток вкладок и кушает все это добро вместе с системой(арч + i3), вайбером, емаксом всего 2100 мб оперативной памяти

simpleadmin 21 сентября 2018 в 10:04

0

Прямо сейчас:

▸ firefox

9 вкладок ~ 15 ГБ
Это далеко не предел, может и один процесс выжрать 10+ ГБ.

Crandel 22 сентября 2018 в 01:18

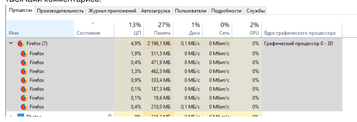
0

Я же писал, юдуб выжирает очень много памяти, возможно другие сайты себя тоже так ведут, а я ими не пользуюсь, поэтому как я ниже приводил скриншот, огнелис с десятком вкладок вместе с системой занимает 2200.

DarthVictor 23 сентября 2018 в 16:17

0

У вас какой-то бракованный фаерфокс (ну то есть сайты бракованные, конечно).
Пример: 22 вкладки из них 6 — ютьюб с запущенным видео и одна — эта страница с 1500 тысячами комментариев.



Надо глянуть нет ли у Лисы опции убивать вкладку после роста потребления памяти выше заданного порога.

simpleadmin 23 сентября 2018 в 16:38

0

> У вас какой-то бракованный фаерфокс

Приобретался у официального дилера. По гарантии обратно не принимают.

> (ну то есть сайты бракованные, конечно).

▸ ff

Перечисляю открытые вкладки
1) www.freebsd.org/cgi/man.cgi?route
2) Google Search Console
3) analytics.google.com
4) analytics.google.com
5) habr.com/post/231277
6) tison.ru/ip
7) tison.ru/ip
8) cloudpbx.rt.ru/login
9) видимо, родительский процесс
код 6-й и 7 -й состоит из одной строчки «return 200 » в nginx

sumanai 23 сентября 2018 в 16:41

0

Дополнения?
К сожалению, мне нечего рассматривать в Google Search Console по причине как отсутствия аккаунта в гугле, так и отсутствия аналитики на моих сайтах, но загрузив всё это в дополнение к полутора десяткам моих вкладок, я получит в два раза меньший объём использования ОП.

simpleadmin 23 сентября 2018 в 20:15

0

Нет, вообще ни одного дополнения. Абсолютно голый ff, без дополнений и синхронизации.
На свежеустановленном ff открываю любую вкладку (можно просто настройки) и через 2-3 недели она сжирает 3-4-5 ГБ

sumanai 23 сентября 2018 в 23:20

0

> через 2-3 недели

Разве что это.

IRainman 23 сентября 2018 в 17:05

0

Возможно что это жрёт 64 битный FF, у меня принудительно стоит 32.

В целом рекомендую открыть about:memory, нажать кнопку Measure и посмотреть что там действительно жрёт

P.S. я знаю, что 64 битный более безопасен, но мне ок.

sumanai 23 сентября 2018 в 17:12

0

Не думаю, что дело в битности. У меня тоже 64 бита.

BigBeaver 21 сентября 2018 в 00:11

0

Хватит писать этот бред. Сколько бы памяти у меня не стояло, при открытом браузере никогда больше полутора гигов не остается свободным.

Crandel 21 сентября 2018 в 00:17

+2

> Хватит писать этот бред. Сколько бы памяти у меня не стояло, при открытом браузере никогда больше полутора гигов не остается свободным.

▸ htop

maisvendoo 23 сентября 2018 в 00:23

0

Это удар ниже пояса для поклонников решений от корпорации Microsoft. Так не честно)

▸ Мой текущий статус

khim 21 сентября 2018 в 00:28

0

Это у вас просто памяти мало. У меня на десктопе с 64GiB, как правило, половина свободна всегда. Даже после того, как Chrome месяц проработал…

Whuthering 21 сентября 2018 в 10:35

0

У Хрома это намеренная политика работы — «кэшируем все что кешируется, и мониторим свободную память и использование свопа, если память нужна другим приложениям или её мало, то освобождаем».

Fracta1L 21 сентября 2018 в 10:48

+1

> мониторим свободную память и использование свопа, если память нужна другим приложениям или её мало, то освобождаем»

Разве это не работа операционной системы?

Whuthering 21 сентября 2018 в 12:43

0

Страницу памяти для ОС можно пометить как используемую и не используемую. Варианта «используем, но если вдруг надо, можем отдать, только вызовите все деструкторы и уведомите наблюдателей» не существует.
Что до мониторинга свободного места и уведомлений приложениям об исчерпании памяти, то насколько я помню, такой функционал есть только в Android, да и то кривой работает пока. Поэтому этим приходится заниматься самим приложениям.

Fracta1L 21 сентября 2018 в 13:07

+1

Я не эксперт в ОС, но насколько я знаю, управление памятью несколько более сложное, чем две метки «используемая» и «неиспользуемая»

Whuthering 21 сентября 2018 в 23:36

0

Это само собой, но я говорю про управление со стороны приложения. А там у нас есть malloc() и производные, free(), горстка алишек для разделяемой памяти и больше никакой магии. Варианта «вот эта память нам пригодится, но не то чтобы сильно нужна, сообщите, если кому другому надо вдруг» я ни в Win, ни в POSIX не встречал.

khim 22 сентября 2018 в 03:18

0

> Варианта «вот эта память нам пригодится, но не то чтобы сильно нужна, сообщите, если кому другому надо вдруг» я ни в Win, ни в POSIX не встречал.

В POSIX этого нет, он на большие системы рассчитан. А в Windows — есть, конечно. Вернее… было. Можете почитать — статья большая, если хочется сразу к сути ищите по слову *discardable*.

Есть также в Android и в MacOS X/iOS. В MacOS (классической) тоже вроде было, но я даже не знаю где доку на неё в онлайн найти…

Sing303 1 октября 2018 в 11:11

0

Странно, куча памяти свободной (даже до 10 не дошло)
Это ещё VS висит с большим проектом
▸ Диспетчер, Chrome, 3 дня, 20 вкладок

BigBeaver 1 октября 2018 в 11:39

0

Просто мне жалко покупать еще 10 гигов просто ради браузера. Все остальное и на 2-4 без браузера живет лучше, чем сейчас на 12 с браузером.

CheatEx 25 сентября 2018 в 15:02

0

Проблема в том что когда открываешь одну вкладку с хабром потребление примерно такое-же.

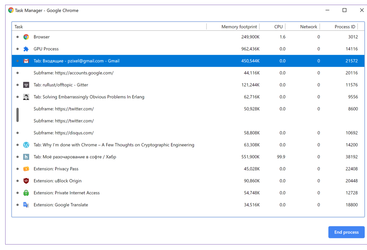
Crandel 25 сентября 2018 в 15:03

0

Какое именно?

PsyHaSTe 25 сентября 2018 в 15:29

0



SuperNovaWS 26 сентября 2018 в 20:57

0

Вас не вводит в когнитивный диссонанс браузер, который выжирает 3-4 гига памяти?! Это, на секундочку, столько, сколько хватает для работы среднего сайта, включая ОС, БД, веб-сервер и генерирующий код, например.

Crandel 26 сентября 2018 в 21:03

0

Нет, потому что я сравниваю с хромом, в котором те же сайты могли съедать всю доступную оперативку (может сейчас ситуация поменялась, но 2 года назад так было). Конечно, хотелось бы, чтобы ресурсов тратилось меньше, но тогда никакого параллелизма не будет, а мне очень нравиться текущий движок огнелиса.

Whuthering 21 сентября 2018 в 10:42

−2

И абсолютно правильно сделали.
Мультипроцессность сделали в первую очередь ради стабильности и безопасности. Программ без ошибок не бывает и не может быть в принципе (люди и их мозги не идеальны), а цена ошибки слишком высока. На потребление памяти кстати отразилось не то чтобы сильно — большая часть кода и данных шарятся между процессами.
С расширениями все ещё интереснее было: наконец-то- удалось избавиться от тонн старого говна во внутренностях браузера и миллионов костылей. Старая архитектура расширений была на редкость непродуманной и позволяла слишком много (даже патчить интерфейс на лету), в результате чего создавала дыры в безопасности, позволяя легко завалить браузер неосторожным движением, приносила лютую попаболь при переносе расширений на новую версию, и главное, препятствовало распараллеливанию задач (в эпоху многоядерных процессоров, лол). Дальше так было продолжать нельзя, и теперь все вздохнули с облегчением.

sumanai 21 сентября 2018 в 11:01

0

> и позволяла слишком много

И это было прекрасно.

PsyHaSTe 21 сентября 2018 в 12:16

0

Нет, не было.

sumanai 21 сентября 2018 в 20:12

+2

Нет, было.
Прекрасные дополнения, которые могли действительно изменять браузер. Classic Theme Restorer, который имел тысячи настроек и мог как воспроизвести интерфейсы старых версий, так и создать совершенно новый, уникальный и удобный конкретно для меня. Дополнения, делающие древовидные вкладки, делали своё дело, разворачивая оригинальное дерево вкладок, а не рисуя в окошке рядом ещё одно. Дополнения могли изменять уть адресную строку, делая кликабельными её сегменты. Добавлять свои панели. Менять страницы самого браузера, делая их удобнее.
Всего это больше нет.

PsyHaSTe 22 сентября 2018 в 03:08

+1

Да, а я помню крашащиеся приложения, Потому что кто-то куда-то не туда записал что-то, синие экраны (которые на домашнем компе после установки семерки, а затем 8 и 10 я не видел), отрубающийся звук и прочие прелести… Да, классик тема определенно того стоит.

sumanai 22 сентября 2018 в 11:28

+1

> приложения

Мы всё ещё про браузер и его дополнения? Какие синие экраны, какие проблемы со звуком? Перечисленные мною дополнения были самыми стабильными, я не припомню проблем с ними. Не знаю, сколько трудов программистов было на это потрачено, да и не важно. Главное — они позволяли делать мне интерфейс браузера таким, каким хочу его видеть я, а не дядьки из управления разработкой.

Massacre 21 сентября 2018 в 13:18

0

Вопрос только, кому теперь нужен получившийся альтернативный Хром? По крайней мере, у FF с позволяющими «слишком много» расширениями была своя ниша. Да и однопроцессность хорошо сдерживала аппетит в плане памяти, в отличии от того, что происходит на скринах выше.

Crandel 22 сентября 2018 в 01:23

0

Мне нужен. Я сильно пользуюсь контейнерами, у меня все гугловские сервисы в одном контейнере, фейсбук и линкедин в другом, а другие сайты не могут меня отследить по супер кукам или кнопкам соц.сетей и составить мой психологический портрет. Да и настроить его можно все равно гораздо шире чем хром.

Massacre 22 сентября 2018 в 04:54

0

Попробуйте Vivaldi.



Crandel 22 сентября 2018 в 13:12

0

Зачем? Меня огнелис полностью устраивает. Вивальди закрытый проект, Его невозможно форкнуть, полностью зависишь от разработчиков.

sumanai 20 сентября 2018 в 21:17

0

Они писали его лет 6, и то до конца не внедрили, кажется до сих пор куски вносят.



dartraiden 21 сентября 2018 в 02:03

+2

Да, они написали движок с нуля, а теперь по частям неспеша переносят его в Gecko. CSS-движок уже перенесли, на очереди WebRender.



PsyHaSTe 21 сентября 2018 в 02:13

+1

Да, и на нем уже клевые проекты пилят, например вот.

kaleman 20 сентября 2018 в 15:36

+14

Это еще не катастрофа. Вот подождите выйдет ваша ос на JS тогда по-другому запоете. Тогда чтобы работать в редакторе понадобится 16-ядерный процессор и 32 гига оперативки.
С учетом того, что весь софт сейчас пытаются запихнуть в браузер, осталось запихнуть туда и операционную систему.



Neikist 20 сентября 2018 в 15:42

0

Гугл в обратную сторону движется. Ну, его часть по крайней мере. Ядро фуксии с нуля пишут, упомянутый в статье Xi, флаттер (кроссплатформа, но компилится в натив, хоть и уровень абстракции местами повышает, но больше для андроида и иос, а для фуксии нативным фреймворком должен быть)

Xandrmoro 20 сентября 2018 в 17:08

+9

> Гугл в обратную сторону движется.

... именно поэтому гмэйл тормозит даже хуже, чем студия с решарпером на проекте на 15млн строк, хотя казалось бы.



Neikist 20 сентября 2018 в 17:09

0

> Ну, его часть по крайней мере



vitvakatu 21 сентября 2018 в 17:27

0

Xi — это не проект Google, хотя и лежит у них на гитхабе. Об этом есть дисклеймер в самом низу readme

KaneUA 21 сентября 2018 в 08:26

0

> Это еще не катастрофа. Вот подождите выйдет ваша ос на JS тогда по-другому запоете.

> С учетом того, что весь софт сейчас пытаются запихнуть в браузер, осталось запихнуть туда и операционную систему.

Сейчас на рынок выходит KaiOS — операционная система, в которой нельзя писать небраузерные приложения. Она позиционируется как ОС для **слабых** телефонов.



arlelf 23 сентября 2018 в 22:21

0

Ахахахаха!

Whuthering 21 сентября 2018 в 23:41

+1

у LG webOS почти весь юзерспейс на HTML+js, и ничего, бегает бодрячком на телевизорах с дохлым армом и менее чем гигом памяти.

Akon32 20 сентября 2018 в 15:42

+15

Какая демагогия.
Если автор хочет программировать под DOS — пусть программирует под DOS. Но *внезапно* окажется, что на дискете с DOS нет ни голосовых помощников, ни словарей и шрифтов на все языки мира, ни сотен драйверов под сотни возможных устройств, ни реализации десятков сетевых протоколов, ни надёжных и быстрых файловых систем, ни работы с защитой памяти, etc, etc; и, боюсь, даже стандартная библиотека C++ на дискету не влезет. Да что там, сейчас только *заготовочные файлы* OpenGL занимают 300КБ. При этом систему нужно было регулярно переустанавливать.
А сейчас у нас на "дисках" гигабайты всякого хлама, и этот хлам более-менее нормально работает. Конечно, работает не идеально, проблемы бывают, но вот с этим:

> нынешняя ситуация — полное дерьмо

я не могу согласиться.

Зато сейчас, с десятками ГБ памяти на борту, я могу спокойно загружать в ОЗУ mesh'и с десятками миллионов треугольников и задействовать какой-нибудь OpenCL, когда того требует задача. Даже с учётом того, что несколько ГБ ~~и миллионы тактов~~ отъедает ОС, это намного удобнее, чем использовать "экономные" системы прошлого.

muove 20 сентября 2018 в 15:57

+4

Удобнее не значит лучше, ресурсы не вечны к сожалению, а мы их тратим на сисечки и кошечек

Akon32 20 сентября 2018 в 16:27

+3

Если "удобнее" позволяет решать какие-то новые задачи, или упрощает решение старых, то как раз это значит "лучше". И решение большой доли задач как раз экономит какие-то ресурсы.



PsyHaSTe 20 сентября 2018 в 21:09

+1

Почему «к сожалению»? Мы тратим ресурсы на то, за что платят деньги. А деньги платят за то, что необходимо людям в первую очередь. Значит мы удовлетворяем первоочередные потребности, а заниматься чем-то другим было бы издевательством над людьми.

Люди: Мы хотим кошечек!
Государство(?): Нет, кошечки это низменная потребность, на самом деле ты хочешь запуск ракеты на Макс. На, держи кусочек ракеты, это теперь твоё! Ну что, рад?
Люди:...

Спрос-предложение отличный механизм, где люди сами решают, что они хотят получить. Поэтому мне совсем непонятны претензии к «кошечкам». И никакие ресурсы к слову не тратятся, ибо никаких невозобновляемых ресурсов на это не тратится, а при изменении приоритетов индустрия перестроится. Понадобится завтра отказаться от всего легаси x86 и перейти на RISC-V? Ну значит так и сделают. Раз не переходят, значит никто этого не хочет, а заставлять — значит ущемлять других людей ради того, что кому-то кажется «более достойной целью».

Emulyator 20 сентября 2018 в 21:50

−2

Спрос/предложение — отличный механизм, когда люди сами решают, осознанно, предварительно подробно ознакомившись о плюсах, минусах, альтернативах. Современные технологии продаж успешно держат количество таких людей на уровне безопасном для прибыльной реализации световых и воздушных колебаний остальным массам.



Am0ralist 20 сентября 2018 в 23:10

−4

При таком поголовном подходе, нужно понимать, мы б все ещё сидели в пещере. Прячась от тигров и мамонтов.



PsyHaSTe 21 сентября 2018 в 00:32

+4

Очень слабая аргументация, в духе запрета интернета «вы же не хотите, чтобы дети стали наркоманами». Как раз текущая организация социума привела к тому, что мы имеем.

Am0ralist 21 сентября 2018 в 11:02

–1

Нет, по моему мнению, все то, что требуется массам слабо способствует развитию прогресса. Прогресс творят как раз таки ученые, инженеры-новаторы и прочие с сильным томлением духа.
А массы хотят две вещи, о чем известно ещё с древнего Рима.
Поэтому опираться на то, чтоб удовлетворять «основные потребности» — это как раз погребать отрасль с кошечках и сиськах. Ракеты народу были не нужны никогда, но чтоб вы сами сейчас делали без спутников? Или без ламповых ЭВМ, которые тоже народу то особо были не нужны?
Так что социум, конечно, кормят, вот только отнимают еще кучу кусков и направляют тем, в чьи задачи вовсе не входит кормить оный. Во всяком случае, не в ближайшие годы.

PsyHaSTe 21 сентября 2018 в 12:22

+4

Что понимать под прогрессом? Если это великая цель, которой все должны стремиться по вашему мнению, а они этим не занимаются, то это скорее говорит о том, что вы чего-то не понимаете.

Как раз индустрия "котиков" создала удобные портативные ПК (вместо тех ужасов со стиками под названием клк), быстрые сетевые протоколы (чтобы котики быстрее грузились), адаптивные фреймворки… Это всё чушь и ниунужна?

К слову про космос — он в известной мере переоценен. Оказалось, что там делать особо нечего, особенно если говорить про межзвездные перемещения. Легче заселить дно океана, чем Макс или там Венеру. Зачем вкладываться в неоптимальные проекты? Просто потому что @Am0ralist так захотелось?

Am0ralist 21 сентября 2018 в 13:15

0

> Если это великая цель, которой все должны стремиться по вашему мнению

Где, где я это сказал?

> а они этим не занимаются, то это скорее говорит о том, что вы чего-то не понимаете.

Почему не понимаю? Понимаю. Прогресс творят единицы. Доли процента. Еще десятки его обслуживают. Ну а прочие пожинают плоды и стенают, что раньше было лучше. Ну или котиков с сиськами лайкают.

> Как раз индустрия «котиков» создала удобные портативные ПК (вместо тех ужасов со стиками под названием клк), быстрые сетевые протоколы (чтобы котики быстрее грузились), адаптивные фреймворки… Это всё чушь и ниунужна?

Неа, индустрия это создала не для котиков. Просто ради котиков их оптимизировали. Просто вы утверждаете, что условные котики самоцель. А для меня — это явная сдача. От тех денег которые корпорации и государства потратили не ради котиков.

> К слову про космос — он в известной мере переоценен.

Спутники не нужны? вот эта вся навигация, спутниковые съемки земли — это всё не нужное барахло?

> Зачем вкладываться в неоптимальные проекты?

Зачем вкладываться в бронзовый топор, когда нам хватает каменного? Зачем изобретать колесо, если можно волоком?
Затем, что совокупно массы не видят смысла ни в чем новом. Еще раз: ЭВМ массам были не нужны. Ракеты обществу были не нужны. Да те же айфоны обществу были не нужны. Пока это всё не появилось по факту. И вот вы пользуетесь сейчас плодами того же развития космических программ и рассуждаете, что ракеты были не нужны, раз люди им были не рады.
Классно, что

PsyHaSTe 21 сентября 2018 в 15:01

0

Что такое прогресс? Почему "вчера люди не могли смотреть котиков из туалета", а "теперь могут" не прогресс?

> Спутники не нужны? вот эта вся навигация, спутниковые съемки земли — это всё не нужное барахло?

подмена тезиса, про ненужность никто не говорил. Про переоцененность — да, я говорил. Точнее это было в 60-70е. Сейчас люди поняли, что там делать особо нечего, и занимаются чисто практическими проектами: прогнозы погоды, там, интернет, GPS, вот это все. Никаких дурацких планов о создании городов на орбите, заселении венеры и т.п.

Am0ralist 21 сентября 2018 в 16:34

+1

> Сейчас люди поняли, что там делать особо нечего, и занимаются чисто практическими проектами: прогнозы погоды, там, интернет, GPS, вот это все. Никаких дурацких планов о создании городов на орбите, заселении венеры и т.п.

Никаких Масков с полетом на Марс, никаких Наса с исследованиями дальнего космоса, сплошь энгриберды на афйонах занимают умы абсолютно всего человечества… А уж сколько сэкономили, отказавшись от таких дурацких планов, как разработка термояда, коллайдера и прочих бесполезных с точки зрения всего человечества вещей!

> подмена тезиса, про ненужность никто не говорил.

> Люди: Мы хотим кошечек!
> Государство(?): Нет, кошечки это низменная потребность, на самом деле ты хочешь запуск ракеты на Макс. На, держи кусочек ракеты, это теперь твоё!
> Ну что, рад?
> Люди:…

а ведь спутников и прочего не было бы без ракет. Людям ракеты были нужны? Нет. Прогресс сейчас без ракет достиг бы таких высот? С чего бы? Без них тогда сейчас не было бы того, что я перечислил. Но ведь людям тогда это же было не нужно. И?

> Почему «вчера люди не могли смотреть котиков из туалета», а «теперь могут» не прогресс?

Это сдача с прогресса для всех. А сам прогресс он именно в том, что я перечислял и начинался задолго до того, как появилась возможность вообще подумать, что котиков можно где-то на экране смотреть, а уж тем более — в туалете.

PsyHaSTe 21 сентября 2018 в 17:54

+1

> Никаких Масков с полетом на Марс, никаких Наса с исследованиями дальнего космоса, сплошь энгриберды на афйонах занимают умы абсолютно всего человечества… А уж сколько сэкономили, отказавшись от таких дурацких планов, как разработка термояда, коллайдера и прочих бесполезных с точки зрения всего человечества вещей!

Вы не передергивайте. термояд — крайне полезная штука, очень надеюсь, что её реализуют. А вот полет на Марс — нет.

И да, никто не говорит про запреты. Хочет маск на марс — пусть летит, он заработал это право в прямом смысле этого слова. Это его деньги, он может ими распоряжаться как считает нужным. А получил он эти деньги занимаясь полезными вещами — автомобилями, платежами и прочим.

> а ведь спутников и прочего не было бы без ракет. Людям ракеты были нужны? Нет. Прогресс сейчас без ракет достиг бы таких высот? С чего бы? Без них тогда сейчас не было бы того, что я перечислил. Но ведь людям тогда это же было не нужно. И?

Людям нужны были ракеты, чтобы воевать. GPS с той же целью был разработан. Если бы они не были нужны, ни ракет ни спутниковы бы не было. Но они как ни странно понадобились. А вот пулеметы, стреляющие точными приборами не понадобились, и их нет, даже если кто-то считает, что прогресс меряется в них.

user343 21 сентября 2018 в 18:52

–2

> Хочет маск на марс — пусть летит, он заработал это право в прямом смысле этого слова.

Не совсем. Сож.жёт пресловутые "сотни нефти", а выхлопы его ракет в нашей атмосфере останутся. Для общества и экологии лучше, чтобы он свои деньги вложил например в антикварные картины.

PsyHaSTe 21 сентября 2018 в 19:26

+4

Ну это-то может быть. Но считать чужие деньги — последнее дело.

Я тут как раз с одним коммунистом знакомым разговаривал, на тему того, что если есть ученый, который изобрел способ на 10% повысить прирост любой еды, а в качестве оплаты просит 5% от неё же. То ест мир получает +5% к производимой еде (бесплатно), человек получает 5% мировых запасов, все довольны. Но тут прецедент, что ученый не хочет рассказать свой секрет за медальку и почетное место на стене своего НИИ, хочет жить на старости без бед. На что мне человек с серьезным лицом говорил, что надо попробовать произвести идеологическую обработку, а если откажется — то клещами доставать.

Как он тратит свои деньги — его дело. Забирать их со словами «ты их потратишь неправильно, я потрачу правильно» — последнее дело. К сожалению, многие любят этим заниматься. Под видео с краш тестами всяких айфонов полно комментарией «ЛУЧШЕ БЫ МНЕ ОТДАЛ». Печально, но что поделать.

Am0ralist 21 сентября 2018 в 23:04

0

> Вы не передергивайте. термояд — крайне полезная штука, очень надеюсь, что её реализуют. А вот полет на Марс — нет.

Потому что исключительно вы так решили?
Еще раз, вы сами сказали, что народ не хотел ракеты. И?
И термояд он не хочет. Ему вообще начхать на науку процентов так 90%. А вот ученым начхать на сиськи и котиков, чаще всего.
Но прогресс толкают — именно последние. А сиськи и котики достаются прочим — на сдачу.

> Если бы они не были нужны, ни ракет ни спутниковы бы не было.

Людям, которым сейчас нужны сиськи и котики — они никогда не были нужны.
Тем, кому это нужно было — их были такие же доли процентов

PsyHaSTe 22 сентября 2018 в 03:11

0

Термояд люди хотят, потому что все хотят машинки на тысячу лошадиных сил, в 10 раз меньшие счета за электричество и бесконечные батарейки. Ну вот хотят.

А на марс они не хотят.

— Короче, ваше разделение на классы «Понимающих», которые хотят приблизить будущее, и быдляков, которые не понимают своего счастья понятно, но так сказать не совсем верно.

Am0ralist 22 сентября 2018 в 10:07

0

> Короче, ваше разделение на классы «Понимающих», которые хотят приблизить будущее, и быдляков, которые не понимают своего счастья понятно, но так сказать не совсем верно.

это ваше разделение. я не утверждал ничего подобного, и не навешивал подобных ярлыков. хотя бы потому, что люди разные и под разные задачи заточены.
А вот ваше утверждение, что люди хотят термояд (видимо вы судите по себе, так как большая часть про него толком ничего и не знает, более того, когда ученые его начали разрабатывать — большая часть не хотела даже не то что айфонов, но даже сотовой связи — ибо ее еще не было), но не хотят на марс — оно *понятно, но, так сказать, не совсем верно*

popov654 21 сентября 2018 в 19:14

0

> быстрые сетевые протоколы

Так они создавались в эпоху на 10-20 лет раньше котиков. Вы ведь про HTTP, SSL, TCP/IP, IPv6, Ethernet? Им лет уже бог знает сколько)

> вместо тех ужасов со стиками под названием кпк

КПК были по тем временам достаточно быстрыми, и умели больше, чем умеют современные смартфоны. Стилусом можно было писать и даже рисовать простенькие вещи. Пальцем такого не сделаешь. Да и софт для КПК был куда богаче по функциям, как уже писали выше. Даже 3D игры были с довольно сносной графикой и геймплеем. Для КПК. Под процессоры в 64-128 МГц, ага. Сейчас под Андроид таких игрушек-то нет…

PsyHaSTe 21 сентября 2018 в 19:27

+1

То есть айфон — это очередной шаг в сторону деградации, а вот кпк на XP были торт, я все правильно понял?

popov654 21 сентября 2018 в 19:47

+1

Она была Windows CE/Windows Mobile 2003, и потом ещё были Windows Mobile 5.0 и 6.0. Я пользовался первыми двумя из названных выше. Вторая даже звонки и смс неплохо умела, с красивым UI.

Айфоны вообще противоречат моей религии, я могу только за современные Андроиды говорить. Ижно это шаг назад, да. Есть некоторый прирост в скорости работы и в том, что экраны теперь нечувствительны к давлению (и то на морозе это минус, нужны специальные перчатки, а если хочешь использовать стилус — то тоже специальный).

P.S. Со скоростью я тоже уже не очень уверен. У меня есть смартфон на Андроиде 2.3, при покупке на нём всё летало. Но я принёс его домой, установил 10-15 приложений, прошло полтора-два года. Приложения потолстели с новыми версиями, плюс добавилось много фоток и скринов, которые система периодически хочет переиндексировать — и некогда шустрый телефон превратился в кошмарный лагодром, от которого я ужасно страдаю уже три года (а поменять не на что, нет на рынке моделей с клавиатурой). То есть я его тупо замусорил.

КПК из нулевых так замусорить было, несмотря на более слабую начинку, куда сложнее. Там количество установленного софта и наличие большого числа фоток на SD карте ничем не грозило.

P.P.S. Облачные сервисы вроде Dropbox — это как раз несомненный плюс. Но клиенты под все вот эти же сервисы и под социальные сети вполне могли быть реализованы и для КПК тех лет, при чём даже при тех аппаратных мощностях(!). Правда, с загрузкой фотографий были бы сложности из-за малого объёма RAM, пришлось бы на сервере генерировать маленькие версии фотографий с меньшим разрешением — впрочем, для экранов 240x320 и не нужно грузить фотографии в Full HD.

ValdikSS 22 сентября 2018 в 10:59

+1

> КПК из нулевых так замусорить было, несмотря на более слабую начинку, куда сложнее. Там количество установленного софта и наличие большого числа фоток на SD карте ничем не грозило.

Да ладно, программы бросали свои экзешники в автозагрузку, все лагало и висело в фоне, оперативки не хватало.

0xd34df00d 22 сентября 2018 в 17:05

0

Ну так отключите сканирование и индексирование всего и вся, будет у вас КПК на стероидах.

SirEdvin 21 сентября 2018 в 22:55

+1

> Так они создавались в эпоху на 10-20 лет раньше котиков. Вы ведь про HTTP, SSL, TCP/IP, IPv6, Ethernet? Им лет уже бог знает сколько)

Сколько там лет HTTP/2? TCP 1,3? Protobuf?

third112 25 сентября 2018 в 16:54

0

К слову про космос — он в известной мере переоценен. Оказалось, что там делать особо нечего, особенно если говорить про межзвездные перемещения.

Это тут уже обсуждалось.

Am0ralist 25 сентября 2018 в 17:21

0

Вот-вот.
Прекрасный пример, когда *ненужные* траты на *ненужные народу вещи* развивали прогресс и принесли изменения в вещи, о которых до этого даже не задумывались.

third112 26 сентября 2018 в 07:28

0

Видимо Вы прочли не всю ветку по указанной ссылке, нпр.:

> До настоящего момента астрономия дала людям, нпр., навигацию, т.е. методы определения положения, нпр., корабля в открытом море. Земля сильно зависит от Солнца и от Луны. Про магнитные бури и приливы-отливы слышали?: Погода, связь и компьютерные сбои зависят от солнечной активности. Предполагают, что и землетрясения. Человечеству для высокого качества жизни крайне важно уметь предсказывать погоду и землетрясения. Для этого нужно изучать развитие звезд, космические явления.

> >Хим био исследования — без космоса почти ничего не возможно?
>
> На школьном уровне, где считается что медь растворима в азотной кислоте. А на Шатле в невесомости вырастили монокристаллы меди, которые в ней нерастворимы. Это не «просто абстрактное открытие» — это путь к созданию материалов с уникальными свойствами.

> > «От астероидов» — Вы же сами понимаете, эта тема просто смешна.
>
> Ошибаетесь. Это реальная угроза для человечества. Пока везло, пока ничего нельзя было противопоставить. Но сейчас появилась реальная возможность защитить Землю.

И т.д.

Am0ralist 26 сентября 2018 в 11:04

0

Вы это к чему, поясните?

third112 26 сентября 2018 в 11:16

0

> Вы это к чему, поясните?

К этому:

> ненужные траты на ненужные народу вещи

Вы не правы. Народу нужна связь и навигация со спутников, нужны прогнозы магнитных бурь, погоды и землятресений, нужны новые материалы и т.д. — поэтому народу нужен космос.

Am0ralist 26 сентября 2018 в 11:51

0

Перечитайте мой комментарий ещё раз плюс обратите внимание на стили текста.
После чего посмотрите, на что отвечал человек, которому ответили вы.

CheatEx 26 сентября 2018 в 11:50

0

Большая часть денег на космос это не фундаментална а вывоз и возврат нежной человечьей тушки. На деньги потраченные на апполны можно было запустить порядка 50 орбитальных телескопов. Вместо летавшего 1 раз, к счастью, бурана можно засыпать венеру автоматическими зондами.

Про химию то же самое, на земле массовые эксперименты приводятся автоматами. Не в режиме растим кристал и пробуем растворить в чём-то, а в режиме растим 500 кристалов, проверяем взаимодействие с 10 реактивов, светим в нескольких сотнях диапазонов.

Фанаты космонавтики а-ля 70е рассказывают по кругу о 10ке интересных экспериментов, но их могло бы быть 10 тысяч, раз в 10 дешевле.

AmOralist 26 сентября 2018 в 11:59

0

Неа, именно вывоз тушки поставил еще больше проблем, которые нужно было категорически решать, что потребовало другие технологии и потом активно применялось в обычной жизни.
А так бы — выводили 50 телескопов с вероятностью какой-нибудь 60/40 да и пофиг…

Druu 21 сентября 2018 в 12:25

+1

> Нет, по моему мнению, все то, что требуется массам слабо способствует развитию прогресса. Прогресс творят как раз таки ученые, инженеры-новаторы и прочие с сильным томлением духа.

Только творят они его именно для того, чтобы удовлетворить требования масс.

> Ракеты народу были не нужны никогда, но чтоб вы сами сейчас делали без спутников? Или без ламповых ЭВМ, которые тоже народу то особо были не нужны?

Массы немцев требовали бомбить саратов — сделали для масс ракеты.

AmOralist 21 сентября 2018 в 14:01

0

> Только творят они его именно для того, чтобы удовлетворить требования масс.

То есть ракеты космические сделали для того, что бы удовлетворить требования масс, которые этого не просили (если исходить из вашей и @PsyHaSTe позиций одновременно)?
Черт, извините, я так не умею.

Druu 22 сентября 2018 в 05:14

0

> То есть ракеты космические сделали для того, что бы удовлетворить требования масс, которые этого не просили

Чего это не просили? Просили, потому и было развитие космоса. Потом массам стало пофиг на космос, и он начал загнивать. Потом пришел дядя Маск, сделал космос снова сексуальным, и массы опять хотят на Марс.

AmOralist 22 сентября 2018 в 10:17

0

> Чего это не просили? Просили, потому и было развитие космоса.

Во-первых, начните с того сообщения, на который отвечал я. А то выглядит как-то глупо. Один утверждает одно, второй другое, но уже мне, отвечающего первому не его доводы.
Но по вашему ответу. *У меня воображение богатое.* Я так и представляю. Приходит Народ к Циолковскому и говорит: гони, понимаешь, теорию реактивного движения! И все такие радостные, с транспарантами.
Потом оттуда сразу идут к Королеву, прям так, в лаптях, и говорят: гони, понимаешь, ракету в космос, да с Юркой внутрях! И не смей, шельмец, другого никого туда сувать!

Или представил картинку, как в 80-е требовали айфон, дабы котики и сиськи в туалетах смотреть.
Народу чаще всего демонстрируют результаты прогресса, убеждают начать этим хотя бы немного пользоваться, потом народ начинает долго утверждать, что на лошадях то оно лучше было, или что счетами быстрее, чем пока с перфокарт данные вобьешь, а потом вдруг оказывается это настолько естественно вошло в в жизнь, что следующее поколение не понимает, как это жить без электричества, автомобилей и компьютеров.
Но прогресс твориться, когда это все только придумывают и начинают воплощать, пока это никому нафиг не упало.

Это как утверждать, что всем людям нужен квантовый компьютер сейчас. Зачем? Они даже не понимают, что это такое. Он нужен маленькой группе людей (напомню, что даже миллион на фоне 6 миллиардов общего населения — это маленькая группа людей), из которых еще меньше понимает, что же он в итоге возможно будет делать.

Druu 22 сентября 2018 в 15:25

0

> Приходит Народ к Циолковскому и говорит: гони, понимаешь, теорию реактивного движения! И все такие радостные, с транспарантами.
> Потом оттуда сразу идут к Королеву, прям так, в лаптях, и говорят: гони, понимаешь, ракету в космос, да с Юркой внутрях! И не смей, шельмец, другого никого туда сувать!

Примерно так все и было, на самом-то деле.

> Но прогресс твориться, когда это все только придумывают и начинают воплощать, пока это никому нафиг не упало.

То, что нафиг никому не упало — в результате умирает не родившись. А выживает только то, что нужно и полезно.

AmOralist 23 сентября 2018 в 14:01

0

> Примерно так все и было, на самом-то деле.

Вау, альтернативная история?

> То, что нафиг никому не упало — в результате умирает не родившись. А выживает только то, что нужно и полезно.

Что становится нужным и полезным.
То, что еще даже не известно — по сути равно никому не упало.
Как не упала теория относительности и возможности создания атомных бомб Чингисхану. Как не нужны были всем автомобили на момент их создания, а сейчас они — везде.
Вы не ставьте потребность в конкретном результате прогресса раньше самого прогресса.

Druu 23 сентября 2018 в 14:49

0

> Вау, альтернативная история?

Почему альтернативная? А по-вашему как оно было?

> Что становится нужным и полезным.

Да нет, что-то либо нужно и полезно, либо ненужно и неполезно. Никакого "становится" тут нет.

> Как не нужны были всем автомобили на момент их создания, а сейчас они — везде.

С чего же вы взяли, что не нужны? Конечно, нужны.

> Вы не ставьте потребность в конкретном результате прогресса раньше самого прогресса.

Почему не ставить, если так и есть? Именно потребности управляют прогрессом. То, что нужно — оно и развивается, то, что не нужно — отмирает.
Вы, видимом, полагаете, что кто-то наугад делает открытиЯ, а потом их пытаются куда-то присунуть? Ну это не так. Точнее — почти что всегда не так, по крайней мере если мы говорим об инженерии.
Глобальным исключением является разве что математика — но в данном случае там люди вполне осознанно занимаются всякими разными вещами чисто для своего удовольствия, потому и нет причин никаких этим вещам быть полезными.

AmOralist 23 сентября 2018 в 15:27

+2

> Почему альтернативная? А по-вашему как оно было?

явно и близко не так, как я описал. Укажите, кто же требовал с Циолковского теорию реактивного движения?

> Да нет, что-то либо нужно и полезно, либо ненужно и неполезно. Никакого «становится» тут нет.

Самолет Райтов не был нужным и полезным в момент создания. Кроме горстки людей, которым это было интересно сделать. Когда же начали летать первые самолеты — массы не набежали с требованием от конструкторов сделать вот прям щаз самолет, который сможет перевозить груз тоннами.
Закон про автомобили может напомнить?

▸ Англия

Итого, прогресс уже пришел, вещь создали, а большинству оно не было нужно и уж тем более оно не было нужно ДО изобретения.

> Вы, видимом, полагаете, что кто-то наугад делает открытиЯ, а потом их пытаются куда-то присунуть? Ну это не так. Точнее — почти что всегда не так, по крайней мере если мы говорим об

> инженерии.

Атомная бомба примерно так и получилась.
И уж тем более народ не стоял под окнами Эйнштейна и не требовал придумать теорию относительности.
Пока вы не открыли воздухоплавание и не поняли его ограничений понять, кк и куда его пристроить — невозможно.
Пока вы не придумали контролируемой ядерной реакции — невозможно придумать атомную станцию.
А уж про геометрию Лобачевского и говорить не приходится.

> То, что нужно — оно и развивается, то, что не нужно — отмирает.

> Точнее — почти что всегда не так, по крайней мере если мы говорим об инженерии.

Планшеты появились задолго до iPad. И ровно до него это были единичные вещи.
Попытки сделать новое идут постоянно и постоянно это оказывается не нужным прямо сейчас, а через десяток лет выстреливает.

Druu 23 сентября 2018 в 15:43

0

> Укажите, кто же требовал с Циолковского теорию реактивного движения?

А при чем тут вообще Циолковский?

> Самолет Райтов не был нужным и полезным в момент создания.

Самолет Райтов — не был, но сама идея некоторого средства, которое позволило бы людям летать — была привлекательна еще за пару-тройку тыщ лет до Райтов. То есть запрос был. Именно по-этому Райты и пытались самолет изобрести, что запрос на такое устройство — был.

> Итого, прогресс уже пришел, вещь создали, а большинству оно не было нужно

Ну как же не было? А зачем кто-то его покупал, если не было нужно?

> Атомная бомба примерно так и получилась.

В каком смысле так и получилось? На атомную бомбу, опять же, существовал вполне конкретный запрос. Вот ее и изобрели. Это же было инженерное решение, в проработку которого было вбухано уйма средств, как в США так и в СССР. На кой черт в это все вбухивали средства, если оно не было нужно? В то, что не нужно, деньги не вкладывают.

> Пока вы не открыли воздухоплавание и не поняли его ограничений понять, кк и куда его пристроить — невозможно.
> Пока вы не придумали контролируемой ядерной реакции — невозможно придумать атомную станцию.

Если что-то на данном уровне развития науки и техники невозможно сделать — это не значит, что оно не нужно.
Людям тысячелетиями был нужен способ быстро двигаться по воздуху, суше и морю. Людям тысячелетиями был нужен способ эффективно вытаскивать себя подобных. Людям тысячелетиями был нужен способ производства дешевой энергии в больших количествах. Людям тысячелетиями был нужен способ удобного общения на дальних расстояниях.

Тысячи лет назад эти задачи решали неэффективно — при помощи мускульной силы, мечей и стрел, лошадей, гонцов и сигнальных систем. Или вовсе никак (если речь о воздухоплавании).
И люди всегда пытались решать эти задачи лучше. В итоге сейчас эти задачи решают при помощи атомных бомб, самолетов, атомных электростанций, машин, интернетов с айфонами.
Пройдет еще тысяча лет — и будут созданы новые, более эффективные способы решения этих задач. Потому что эффективное решение этих задач людям нужно. Сейчас нужно, было нужно тысячи лет назад и будет нужно тысячи лет спустя.
Есть общественный запрос — есть развитие.

> Попытки сделать новое идут постоянно и постоянно это оказывается не нужным прямо сейчас, а через десяток лет выстреливает.

Такие случаи, действительно, бывают, но это *единичные исключения*, которые вы почему-то полагаете правилом.
В реальности же — практически все новое это результат уже существующих запросов. А то, что появляется но не требуется — в итоге умирает, навсегда умирает.
Да, *единичные исключения* действительно есть. Но это именно *единичные исключения*.

> Планшеты появились задолго до iPad. И ровно до него это были единичные вещи.

Потому что планшеты сами по себе никому не нужны, ни 20 лет назад, ни сейчас. Они бесполезны без инфраструктуры. Мобильная инфраструктура была развита с развитием, с-но, смартфонов.



Alexeyslav 21 сентября 2018 в 15:51

0

Разводили бы турбо-лошадей…



0xd34df00d 21 сентября 2018 в 22:41

+1

Только творить прогресс удобнее и продуктивнее, когда есть некоторое ощущение свободы, и когда Партия не насаждает, какой именно прогресс надо творить.

Да и то ведь, если задуматься, абстрактный матан и теоретическая физика не сильно отличаются от кошечек. Ну кому-то кошечки интересны, а мне закорючки в тетрадке рисовать нравится. Я ничем не лучше.



Am0ralist 21 сентября 2018 в 23:17

0

Так даже в ссср (до тех пор, пока построение утопии не выродилось в фарс, а государство не стало пародией) это понимали и на ученых послабляйка была.
А вообще, морковка спереди, морковка сзади будут всегда независимо от. просто в некоторых системах они будут заметнее.



0xd34df00d 22 сентября 2018 в 00:06

0

Ну и всем нравятся свои сорта морковки (хотя корреляции прослеживаются).



Ogra 22 сентября 2018 в 07:03

+2

> на ученых послабляйка была

Шаражки на ученых были, а не послабляйка.

akryukov 24 сентября 2018 в 11:10

0

СССР был достаточно долго, чтобы быть разным. Одно время — шаражки и то, подозреваю, что не всем. Другое время — послабляйка.

Osnovjansky 25 сентября 2018 в 14:34

0

А вы почитайте внимательнее, как во времена шаражек жили обычные работяги, и как жилось в шаражках. Кстати, при работе над стратегически важными темами секретность и заборы с колючкой были необходимостью во всех странах

0xd34df00d 25 сентября 2018 в 20:29

0

Я в свой время почитал про разработку YF-23. Техника меня интересовала больше, но и жизнеописания там немного были. В общем, секретность — да, колючие проволоки — нет.



Ogra 26 сентября 2018 в 07:12

+1

Секретность, это когда заборы с колючкой на вход, а не на выход.

Osnovjansky 26 сентября 2018 в 11:05

0

Да, в своем комментарии я несколько переборщил. Шарашки — явление нездоровое, хоть и имевшее исторические причины.

Но что касается колючки на выход, то, если мне не изменяет память, в свободолюбивой США Манхэттенский проект проходил без права на выход, разумеется это был добровольный выбор работавших там.

Fracta1L 21 сентября 2018 в 08:23

+6

О, началась песня про правильное использование ресурсов. Таким дай волю — через 10 лет обнаружишь, что тебе положена миска риса в день XD



PsyHaSTe 21 сентября 2018 в 12:24

+1

Целая миска риса? А что Ты Сегодня сделал для Человечества, чтобы еще требовать от него что-то?

novice2001 20 сентября 2018 в 16:01

+14

А весь этот хлам нужно тащить всем и сразу? Словари и шрифты на все языки мира, драйвера на устройства, с которыми 99.999% никогда в жизни не столкнется и все такое?
Работа с защитой памяти и фаловыми системами — это, ну, десятки мегабайт, но никак не гигабайты. Есть интернет и есть базовая концепция модульности. Должно быть ядро с необходимой ВСЕМ базовой функциональностью, а все остальное должно загружаться по мере надобности.

Neikist 20 сентября 2018 в 16:03

+6

> а все остальное должно загружаться по мере надобности

— так себе вариант. Далеко не везде и всегда этот самый интернет есть.

novice2001 20 сентября 2018 в 16:15

–5

Предполагаю, что там, где нет интернета, вряд ли кому-то понадобится драйвер уникального устройства, поддержка редкого сетевого протокола или словарь тамильского языка. Так же как распоследний рантайм .Net, ибо если нет интернета, то откуда возьмется программа, которая его потребует?

Akon32 20 сентября 2018 в 16:21

+4

Именно там, где нет интернета, это всё и требуется. Различные производства в ~~тайге~~, например.

novice2001 20 сентября 2018 в 16:25

+1

Вот именно на производстве до смерти необходимы словари всех языков мира, голосовой помощник и прочий хлам???

Neikist 20 сентября 2018 в 16:29

+3

Нет, там нужны драйвера, а также забитый по дефолту софт (чтобы его спокойно либо из образа восстановить, либо накатить с носителя, но никак не из сети)

novice2001 20 сентября 2018 в 16:35

+1

Делаете **свой** образ с нужным именно вам софтом и нужными именно вам драйверами. Хоть сразу миллионом. И накатываете куда вам нужно. Зачем весь этот мусор всем остальным???

polearnik 21 сентября 2018 в 10:52

+1

Ага и для того чтоб у вашего ребенка появился ноут вам надо будет создать свой образ системы а потом для того что бы добавить туда какуюто программку обновить его. Не хотите делать свой образ? Ну ничего страшного идите на сайт и пытайтесь выбрать подходящий из сотен тысяч образов на все случаи жизни. Вы спец который это сделает одной левой? А ваша жена нет как и тысячи людей по всему миру. Они просто хотят купить ноут и начать работать с ним

Am0ralist 21 сентября 2018 в 11:09

0

И для этого к ноутам всегда был приложен диск с драйверами под разные ОС…

freuser 22 сентября 2018 в 22:59

+1

К моему Asus K53BY был приложен диск с драйверами размером всего 1,6 ГБ под целых две ОС — Windows Vista и Windows 7. Всё. XP не может звук вообще и разрешение выше 1024x768, 10-ка тупо не становится, виснет. Причем драйвера ставятся тоже интересно: *Блютуз? вот тебе блютуз. правда, он работает через нашу фирменную программу, которая будет висеть в трее и отжирать 5% ЦПУ всегда, но ведь 5% это немного, всегда рады помочь. Графика? вот тебе графика. в процессе установки мы перезагрузимся дважды, и плюс каждую загрузку ОС ты будешь ждать дополнительно пару минут, но зато 6,2 балла по встроенному бенчмарку*
Пришлось с работающей семерки выдирать только дрова без довесков, теперь они заняли 354 МБ на диске D:.
Для сравнения, дистрибутив Linux SIDUX, который увидел все мое железо из коробки (многострадальный синезуб потребовал однострочника на шелле, чтобы загрузить модуль с другим параметром), весил чуть больше 400 МБ, и был полнофункционален для бытового использования (разве что торрент-клиент отсутствовал). Так что драйвера тоже разные бывают.

novice2001 21 сентября 2018 в 11:51

+2

Вы написали ерунду, потому что вместо того, чтобы прочитать всю ветку чужих комментариев (и содержащийся в ней контекст), прочитали только один последний, зато поторопились написать очень важный свой. Впредь, чтобы не выглядеть клоуном, не повторяйте эту ошибку.
Нигде в моих комментариях не написано, что создание образа — это единственный способ установки софта. Этот способ для конкретного специфического применения — промышленное использование вдали от сети. Где этим занимаются профессионалы, вполне способные создать необходимый образ. Что касается детей и жен, они будут по старинке скачивать и устанавливать необходимое из сети, что они успешно делают и сейчас. Но этот способ был раскритикован таежными профессионалами, которые тоже по какой-то неведомой причине, посчитали, что он должен быть единственным.
В итоге вы оба спорите не с моим мнением по этому вопросу, а с мнением выдуманным вами, но почему-то приписываемым мне.

Akon32 20 сентября 2018 в 16:36

0

Я как-то устанавливал драйвера на принтер в тайге. Думаю, если бы на компьютере не оказалось хлама типа драйверов CD-ROM или USB-флешек, мне бы это не удалось. Это, конечно, распространённые вещи, но с тем же успехом мне мог понадобиться дотнет, или vcrt определённой версии.

novice2001 21 сентября 2018 в 08:11

+2

Вот я и говорю, давайте для удобства сотых долей процента вломим несколько десятков гигабайт всем подряд. И, самое, главное, даже при таком подходе, все равно найдутся люди, на чье железо драйверов нет.
Так что, как написал @kbaa, собираетесь в тайгу — берите все, что может понадобиться с собой, а не надейтесь на чудо.
Взгляните правде в глаза: какой-то .Net вам нужен для конкретной программы. Если в тайге нет сети, то взяться она может только из сети в офисе. Ну, так проверьте требования и скачайте в том же офисе нужный рантайм. До того, как обнаружится, что на компьютере в тайге его нет. Это разумно и логично, не?
Или заранее интегрируйте в образы ОС для своих компьютеров все нужное. Не заставляйте всех терпеть весь этот ненужный подавляющему большинству хлам.

0xd34df00d 21 сентября 2018 в 22:52

0

Напомнило мне мой 2003-й, когда винмодем под тогдашним линуксом у меня не завелся, поэтому я костылял скрипт, запрашивающий у пакетного менеджера список пакетов для установки, складывавший их урлы в файл, потом скидывал этот файл на дискету, шел к компу с интернетом, скачивал там, шел обратно, подкладывал файлы куда надо, радовался жизни.

Золотые были времена! А трава, главное, такая зелёная была!

Am0ralist 21 сентября 2018 в 23:22

0

На пару лет позже я свой адсл юсбшный на slax'е заводил: трава тогда точно была зеленее… а хаутушки — укуреннее.

kbaa 20 сентября 2018 в 19:27

0

Из моего опыта — если едешь туда, где нет интернета, то или берёшь все драйвера на дисках/флешках, или потом придётся наслаждаться gprs/edge скоростями (если повезёт) или ехать второй раз
не знаю, драйверы к чему сейчас есть по дефолту, но ни к одной железки из тех, что я встречал в охранных системах — ничего естественно нет. Что уж говорить про чтото более специфичное и промышленное.

Neikist 20 сентября 2018 в 16:23

0

Почему это? Собрали коробку с софтом, и отправили в глухую тайгу, лишь изредка приезжая и забирая собранные/наработанные данные. А если накроется периферия — то искать нужно в соседней деревне за 50 км где на выбор 3 железки ни одна из которых не совпадает со сломавшейся.

novice2001 20 сентября 2018 в 16:30

+5

Ну, здорово. Чтобы кому-то в компании, где черт ногу сломит, было хорошо, давайте вломим по нескольку десятков гигабайт хлама всем вообще.
Вместо того, чтобы в отдельно взятой компании навести порядок с периферией и драйверами для нее.
Логика ОК.



Neikist 20 сентября 2018 в 16:36

0

> Есть интернет и есть базовая концепция модульности. Должно быть ядро с необходимой ВСЕМ базовой функциональностью, а все остальное должно загружаться по мере надобности.

В этой концепции компания навести порядок не сможет просто потому что вы предлагаете единственный способ доустановки модулей. Ну и насчет дров — лично меня бесит что я должен для своей материнки загружать драйвер с помощью телефона, потом подключать телефон как флешку и этот драйвер устанавливать чтобы загрузить все остальные драйвера уже с компа.

novice2001 21 сентября 2018 в 08:01

0

Серьезно? И где именно написано, что я предлагаю **единственный** способ? Просто мой способ — удобный для подавляющего большинства, ибо подавляющее большинство офисных ОС установлено именно в офисах с интернетом, а не в тайге. Это не исключает интеграции программ и драйверов в ваш личный образ. Просто это должно быть опцией, а не десятками гигабайт предустановленного хлама для всех подряд, потому что кого-то что-то там бесит.

Porohovnik 20 сентября 2018 в 22:38

0

Так можно на флешку ОС со всем функционалом, а при установки выбирать что установить, а что нет.

Borjomy 21 сентября 2018 в 10:25

+1

А для того, чтобы грузить весь код сразу — он, значит, есть

Akon32 20 сентября 2018 в 16:18

0

> А весь этот хлам нужно тащить всем и сразу?

Обычно балансируют между "тащить ВСЁ" и "загружать всё необходимое из интернета". Текущий баланс — порядка нескольких ГБ на ОС (на windows, кажется, чуть больше).

В дистрибутивах linux есть "минимальные" установочные образы с загрузкой остального по сети — мегабайт 100-200. Но установленная ОС со всеми типичными серверами занимает те же пару гигабайт как минимум. Не могу сказать, что это мало.



0xf0a00 20 сентября 2018 в 16:49

+4

2-3 гигабайта против 10-15, причем эти 2-3 гигабайта обладают большим функционалом. Парадокс.



PsyHaSTe 20 сентября 2018 в 21:16

+1

Полагаю, разница в ресурсах, картинках и прочей мишуре, а кода там примерно столько же.

Ну вот я поставил убунту 1804, выбрал язык русский, но все равно в интерфейсе половина текста на английском. Даже в самой установке все галки на русском, а вот «Include third-party software» не очень. Как бы тут не было правила 80/20, что 10-15гигабайт имеет в 4 раза (80%) больше файлов, чтобы покрыть вот эти «мелочи», а 2-3 гигабайта нет.

Плюс возможно там слой совместимости с программами, написанные в бородатые годы, ибо в 2-3"игабайтах часто практикуется совместимость на уровне исходников, а не бинарная.



0xd34df00d 21 сентября 2018 в 22:58

0

> Плюс возможно там слой совместимости с программами, написанные в бородатые годы, ибо в 2-3"игабайтах часто практикуется совместимость на уровне исходников, а не бинарная.

Именно. Плюс по одной копии библиотек для всего, не таскает каждая программа всё вплоть до рантайма с собой.



PsyHaSTe 20 сентября 2018 в 21:10

+2

Есть некая ОС, в которой всё это включено по-дефолту (скажем, Убунту).
Есть ОС, где всё надо пилить самому (тот же Альпин).

Как думаете, какую ОС чаще ставят в качестве домашней и почему?

novice2001 21 сентября 2018 в 09:47

0

Некорректное сравнение. Это совсем не то, о чем я говорю.

Simplevolk 21 сентября 2018 в 13:14

0

а какую из них выбирают в качестведocker-образа и почему?)



SirEdvin 21 сентября 2018 в 13:24

+2

Ubuntu/Debian, потому что в apline постоянно что-то не работает. Они используют musl и там куча несовместимостей и недоделок, например, инструкции search в resolv.conf не работают. А еще они там решили перейти на libressl и теперь каждый второй софт, который работает с криптографией становится туда только с бубнами.

VolCh 23 сентября 2018 в 00:16

0

Словари не знаю, но как всем предоставить интерфейс загрузки чего-то нестандартного даже если шрифта кириллического или японского нет в системе? И как загрузить драйвер сетевой карты, а видео? А если сети нет, то хотелось бы на принтере распечатать название нужного сетевого драйвера, а невидящему пользователю услышать его.



staticmain 21 сентября 2018 в 10:26

+1

> Но внезапно окажется, что на дискете с DOS нет ни голосовых помощников, ни словарей и шрифтов на все языки мира, ни сотен драйверов под сотни возможных устройств, ни реализации десятков сетевых протоколов, ни надёжных и быстрых файловых систем

Конечному пользователю от всего вышеперечисленного ни тепло ни холодно — ему не нужен «голосовой помощник», от которого проку — поставить будильник и отказаться вызывать скорую. Ему не нужны словари на все языки мира, он пользуется одним-двумя, ему не нужны десятки сетевых протоколов — гофер остался в прошлом, сейчас кроме tcp/udp http/https пользуются разве что ftp и то отдельные личности. Этими вашими mtproto пользуются отдельные приложения и то, не как отдельным протоколом, а уровнем абстракции над http, в системе поддержка этого не нужна. Сотни драйверов конечному пользователю не нужны, у него максимальный набор чипов — два десятка, это НЕ нужно по умолчанию тянуть в систему.

> и, боюсь, даже стандартная библиотека C++ на дискету не влезет.

```
-rwxr-xr-x 1 root root 1868984 янв 15 2018 /lib/x86_64-linux-gnu/libc-2.23.so
-rw-r--r-- 1 root root 1566440 июн 22 01:55 /usr/lib/x86_64-linux-gnu/libstdc++.so.6.0.21
```

Все прекрасно помещается.

khim 21 сентября 2018 в 14:31

+3

Не понял что там у вас помещается. Оба файла по размеру больше всех распространённых типов дискет.



staticmain 21 сентября 2018 в 14:44

−2

Странные у вас дискеты, меньше 1.5 МБ:
```
~$ ls -lh /lib/x86_64-linux-gnu/libc-2.23.so
-rwxr-xr-x 1 root root 1,8M янв 15 2018 /lib/x86_64-linux-gnu/libc-2.23.so
~$ ls -lh /usr/lib/x86_64-linux-gnu/libstdc++.so.6.0.21
-rw-r--r-- 1 root root 1,5M июн 22 01:55 /usr/lib/x86_64-linux-gnu/libstdc++.so.6.0.21
```

ValdikSS 21 сентября 2018 в 15:11

+9

Казалось бы, не много-то времени прошло, а люди уже не помнят ёмкость самых распространенных дискет. На дискету помещается 1.44 МБ. Есть 2.88 и ZIP-дискеты, но они не были популярны или распространены.

khim 21 сентября 2018 в 21:55

+4

> Странные у вас дискеты, меньше 1.5 МБ:

Это у вас они странные. Начинались дискеты с размера 170KB, кончились 1.44MB (причём это было обсосбый такой специальный «флоповый» мегабайт в 1024000 байта).

Дискеты в 2.88MB и ZIP-drive — это экзотика, они мало на каких компах были.

Akon32 25 сентября 2018 в 17:45

0

Те дискеты, которыми мне приходилось пользоваться, вмещали 1440КБ (~1.38МБ).



iroln 22 сентября 2018 в 16:44

+4

Мне кажется, вы не поняли, что хотел сказать автор. Он не предлагает стать "дауншифтером" и принять аскезу, используя DOS и ограничиваясь 640 килобайтами. Он всего лишь хочет, чтобы программы были быстрыми, качественными, надёжными и не бессмысленно раздутыми. И да, я сейчас пишу это сообщение в браузере Google Chrome, который недавно обновился, и теперь этот текст набирается с сильно заметным лагом и фризами на моём Core i7 и 16 Гб ОЗУ. Если это не дерьмово, то я тогда не знаю...

Druu 23 сентября 2018 в 03:56

−1

Дык, это, платите деньги, они и будут. Вы вот сколько за месячную подписку на браузер платите? Небось, 0$? А за хабр? Столько же? А чего вы тогда хотите за этот 0$?

Или вам бесплатно должны делать качественные, надежные, быстрые приложения? А сами бесплатно не хотите поработать? Нет? А чего так?



iroln 23 сентября 2018 в 09:29

+2

Вы серьёзно?! У меня есть просто огромное количество примеров софта за астрономические деньги, который ужасен настолько, что не понятно как он вообще может работать. Например, всякие штуки уровня интерпрайз типа Cisco WebEx или адовые ERP за десятки тысяч долларов. А софт во всяких "умных" телеках, которые адски тормозят, а к тому же ещё и решето с точки зрения безопасности. Всё, я даже продолжать не хочу.

На счёт хрома, Google знает всё обо всех и зарабатывает на этом, если ты не платишь за продукт, то ты сам становишься продуктом, это вроде все знают. На счёт подписок, я пользуюсь продуктами JetBrains по подписке, и что, думаете, там всё идеально? Хахахаха. Баги годами висят в багтрекере. С каждым новым релизом платформа intellij idea требует всё больше ресурсов и всё тормознее, а какого-то серьёзного прогресса по функциональности я что-то не замечаю.

Druu 23 сентября 2018 в 14:20

−2

> Вы серьёзно?! У меня есть просто огромное количество примеров софта за астрономические деньги
>
> Замечательно, то есть к тормозящему браузеру и хабру все претензии сняты?

> У меня есть просто огромное количество примеров софта за астрономические деньги, который ужасен настолько, что не понятно как он вообще может работать.
>
> Прекрасно! Если этот софт за астрономические деньги плох — купите хороший софт от конкурентов! Или предложите за те же деньги свой софт — который будет быстрым, надежным и нераздутым. И те, кто пишет плохой, тормозной, негодный софт — разорятся из-за недостатка клиентов. В чем проблема?

> На счёт подписок, я пользуюсь продуктами JetBrains по подписке, и что, думаете, там всё идеально? Хахахаха. Баги годами висят в багтрекере. С каждым новым релизом платформа intellij idea требует всё больше ресурсов и всё тормознее
>
> Так не пользуйтесь продуктами джетбрейнс или идеей. Пользуйтесь продуктами, написанными хорошими, годными инженерами! Которые пишут быстрые, надежные приложения.
> На худой конец — у вас же всегда есть вим или емакс. Почему вы не пользуетесь вимом или емаксом?



arielf 23 сентября 2018 в 22:30

0

Автор и многие совершенно не понимают — зачем версии Android на **конкретном** телефоне нужны 6 Гб драйверов на все возможные интерферометры из ЦЕРНа? Или 100500 прочих телефонов — а? В конце концов ось модульная и предустановленная — неужели нельзя забабахать норм кастомизацию — шоб на каждый телефон шли лишь нужные вещи? Зубами клянусь — реально нужных вещей в нём на гиг максимум!



mapron 20 сентября 2018 в 15:48

+2

Очень сильно напомнило suckless.org/philosophy



PsyHaSTe 20 сентября 2018 в 21:18

+1

Полный треш, а не философия. «Если у вас нет собаки, её не отравит сосед». Особо эпичные места можно цитировать, но пожалуй тут смысла этого делать нет. Просто попробуйте попользоваться софтом, написанным с такой философией, я посмотрю. Обычно разработчики просто начинают крушить всё подряд, если окажется, что в некоей софтине нет подсветки кода, автоматических рефакторингов, плагинов, автодополнения, интеграции с сотей системой версионирования, а также нельзя выбрать ровно 15 строчек и заменить в них foo -> bar. Что говорит о некоемом двоемыслии, с позиции разработчика и пользователя соответственно.

НЛО прилетело и опубликовало эту надпись здесь



PsyHaSTe 20 сентября 2018 в 22:05

0

Да, примерно это я и собирался расписать, хорошо, что теперь не придется.

Софт по такой философии я видел, но выглядел он совершенно некозабельно. Например

> (Non-)Features
> …
> * No different font styles (bold, italic, underline)
> * No fancy layout options (different font sizes, different colors, …)
> …
> * Slides with exuberant amount of lines or characters produce rendering glitches intentionally to prevent you from holding bad presentations.

то есть отсутствие фич и есть фича, как и отсутствие обработки краевых случаев



ValdikSS 21 сентября 2018 в 01:53

+7

> Веб плохой? Ну покажите класс, запилите хороший, годный, посещаемый ресурс, приносящий прибыль, который будет написан без тонны клиентского кода, и работать молниеносно (что смешнее еще и без тонны рекламы же надо им всем, и вот этого всего, чтобы только текст там).

old.reddit.com

sumanai 21 сентября 2018 в 09:37

0

Высокая посещаемость reddit.com не вызывает сомнений, но сколько из пользователей читают его именно через этот поддомен?



ValdikSS 21 сентября 2018 в 10:04

+1

Редизайн случился не так давно, менее года назад. До этого он всегда был таким, как сейчас на old.

Fracta1L 21 сентября 2018 в 10:26

0

А что в старом дизайне хорошего? В новом можно просто прокручивать главную страницу, и сразу видно, какой тред стоит открыть, а в каком очевидно ничего интересного нет, в старом виден только заголовок и картинка размером с биту, вот же весело ломать глаза себе. И это только самый очевидный недочёт. Если проанализировать, можно ещё кучу найти, думаю.

volchonokk1 21 сентября 2018 в 11:26

0

Так это не совсем то что сверху написали. Это предыдущая версия текущего реддита перед редизайном, а не специально написанный аналог с философией минимизации и оптимизации.

НЛО прилетело и опубликовало эту надпись здесь

Tsimur_S 21 сентября 2018 в 16:03

+1

> Ну так давайте, чем вам не конкурентноспособное приемущество запилить более быстрый slack или скайп, или что-угодно-еще, и показать всем как надо.

Да хотя бы телеграмм, 30 мб памяти против 300+ у слаки и 100+ у скайпа. Если бы все писали такие приложения то этой статьи бы не было.

> Беда в общем, не в сфере. А в головах людей, что дальше собственного носа и потребностей видеть неспособны.

К сожалению других людей нету, не завезли. На этих людей только и можно подействовать нытьем. Будет больше таких статей, разной степени техничности, больше людей будет задумываться о том почему старый айфон резко начал тормозить после выхода нового апдейта ios.

SirEdvin 21 сентября 2018 в 16:39

0

> Да хотя бы телеграмм, 30 мб памяти против 300+ у слаки и 100+ у скайпа. Если бы все писали такие приложения то этой статьи бы не было.

Мм… вы сейчас серьезно?
Покажите в телеграме:

- Воркспейсы
- Треды
- Интерактивных ботов (со списками, выпадающим меню и так далее)
- Возможно красиво форматировать сообщение
- Грамотную организацию вкладок и разделение потоков. Что-то в духе личка и приватные чатики на 2-3 человека отдельно, а общие каналы отдельно.
- И так далее

PsyHaSTe 21 сентября 2018 в 18:26

0

1. -
2. -
3. Такие боты есть
4. поддерживается markdown
5. -
6. ...

SirEdvin 21 сентября 2018 в 18:32

0

1. Таких ботов нет. Посмотрите как работает и выглядит MailClark или же интеграцию с jira. Telegram до такого еще довольно далеко
2. Нет. Поддерживается огрызок синтаксиса маркдауна в виде bold, italic текста, ссылок и verbatim режима при помощи кавычек. В том же slack как минимум поверх этого есть еще заголовки и куча других полезных штук. Не учитывая, насколько гибкое форматирование сообщений в слаке.

Телеграму с их "текст и кнопки" еще далеко идти.

PsyHaSTe 21 сентября 2018 в 18:41

0

1. А это разве не "интерактивный бот"?



2. Видимо разрабы решили, что этого огрызка хватит. Учитывая, что в телеграмме длинные строки выглядят ужасно, звучит логично.

SirEdvin 21 сентября 2018 в 18:43

0

На мой вкус телеграму не хватает возможности добавить выпадающий список, потому что без него длинный список вариантов работает грустно.

А так да, есть.

iroln 23 сентября 2018 в 18:57

0

Вот это всё, что вы перечислили, должно требовать каких-то огромных ресурсов что ли? Или в чем смысл коммента? А со скайпом чего не сраниваете тогда? :)

Скайпом на смартфоне я просто не пользуюсь, потому что невозможно. Тормозит, зависает, вылетает, потребляет слишком много памяти.

SirEdvin 23 сентября 2018 в 21:49

0

1. Скайп написал в microsoft, они просто пишут плохие продукты.
2. Все, что я перечислил возможно еще и потому, что slack решил не выпускат 100 и 1 продукт для разных платформ, как сделал telegram, а сконцентрировать свое внимание на одной и просто портировать результаты с нее на другие платформы с минимальными затратами. Меньше затрат на порт — больше фич. У телеграма вот все еще нет секретных чатов на десктопе.

Massacre 22 сентября 2018 в 02:06

0

Вообще, телеграм на десктопе это примерно 60мб на старте и утечки памяти вплоть до 500+, но хоть не электрон, т.е UI не лагает… Скайп 7.x был в районе 100-150мб, если учитывать браузерный компонент, да.

А слак такой раздутый не из-за дополнительных фич, а именно из-за электрона.

mapron 22 сентября 2018 в 12:00

0

Ну я лично не агитирую ни за, ни против.
По поводу «попробуйте пользоваться софтом» — с сайта проекта:
— zlib
— musl
— libev
— DeadBeeF
— mc
— MPV (мой дефолтный плеер на винде)
Это то чем пользовался я. Так-то порядочно там довольно популярного софта:
suckless.org/rocks
Вы видели исходники musl?

Что касаемо разработки DE под такой идеологие, или IDE — то я лично уже хз, тут я бы поспорил, что suckless зайдет)

SirEdvin 22 сентября 2018 в 13:36

0

> musl

Не поддерживает 1000 и 1 вещь, к который вы привыкли. Мне лично понравились проблемы с search в dns-resolve.

> Вы видели исходники musl?

Сколько там ему лет? Когда доживет до glibc тогда и поговорим. Написать классный код довольно просто, а вот поддерживать.

mapron 22 сентября 2018 в 13:40

0

Очень надеюсь до glibc никогда не доживет :) у них все же разные сферы применения. Для «микро» систем — самое то (uclibc я еще терпел из-за mmuless).
7 лет вроде поддерживает. OpenWRT на него перешел уже.
А если вы хотели сказать «когда ему будет столько же лет, сколько glibc» — то вообще не смешно.
Про список совместимости — я в курсе, спасибо, я читал внимательно, как и баги которые чел заводил в других проектах (это уже просто для себя).
Как полную замену glibc во всех отношениях никто проект и не позиционировал. Но я да, считаю, что это один из правильно сделанных suckless проектов.

SirEdvin 22 сентября 2018 в 13:44

0

Вики пишет, что 4 года назад вышла стабильная версия (

Ну а касательно никто не позиционировал — а вы уверены? Его вот в alpine утащили, который довольно популярен для docker образов, которые вполне себе на серверах крутятся.

mapron 22 сентября 2018 в 13:51

0

musl-1.0.0.tar.gz — March 20, 2014
Наверное речь об этом. Я лично с проектом познакомился еще до релиза 1.0.0, он уже тогда был вполне юзабельный)

Про позиционирование, что-то нигде не нашел про замену glibc. Да, пишут, мол general-purpose; что до использования на серверах — ну видимо майнтейнеры знают, как обходить особенности resolv.conf и прочую муть)

(давайте завяжем с оффтопом наверное, а то и так комментов полторы тыщи)

SeOd 20 сентября 2018 в 15:50

+9

Ох, если еще посмотреть на js, ноду и папочку node_modules. То там тоже боль. Особенно, если застал как было до. Но теперь:
«двести метров джаваскрипта грузят текста триста байт»
https://soundcloud.com/viruzzz-kun/kolkhoznyy-frontend

shybovycha 20 сентября 2018 в 15:53

+5

> В современных текстовых редакторах задержка при наборе больше, чем в 42-летнем Emacs. Текстовые редакторы! Что может быть проще? На каждое нажатие клавиши, нужно всего лишь обновить крошечную прямоугольную область на экране, а современные текстовые редакторы не могут сделать это за 16 мс.

Так эти ваши "современные текстовые редакторы" в браузерах крутятся же! Нынче модно стало ваять не нативные приложения, а кроссплатформенную жесть, которая тащит за собой и браузер и движок жаваскрипта.

Neikist 20 сентября 2018 в 15:56

+2

Кроссплатформа и без js и браузера может быть к счастью. Вспомнить тот же qt, kivy (python, но все лучше js+html+css), flutter.

shybovycha 20 сентября 2018 в 16:04

0

так я и не говорю что это хорошо — я говорю что это модно. нынче здравый смысл проигрывает моде / трендам чуть более, чем по всем позициям

Neikist 20 сентября 2018 в 16:05

+2

Я говорю о том что кроссплатформа это не всегда трэш из веба.

Fedcomp 21 сентября 2018 в 12:00

+2

Как раз здравый смысл. Дешевле сделать на электроне чем нанять Qt программиста.

shybovycha 21 сентября 2018 в 12:06

0

кому — "дешевле", кому — "здравый смысл" =)

yarston 20 сентября 2018 в 16:59

0

В гугл плее про демо приложение на флаттере (Flutter gallery) пишут, что жирное и лагает.

Neikist 20 сентября 2018 в 17:02

0

Да? Странно, видимо зависит от чего то. Меня эта демка после приложений которыми ежедневно пользуюсь очень порадовала.

chupasaurus 20 сентября 2018 в 17:14

0

> Вспомнить тот же qt

В котором для любителей JS вне браузеров есть QTWebkit...

Crozzy13 20 сентября 2018 в 17:55

+1

Что угодно придумают, лишь бы на плюсах не писать. Про порты Qt для других ЯП я вообще молчу

0xd34df00d 21 сентября 2018 в 23:01

0

Вы его, похоже, с QML перепутали.

chupasaurus 21 сентября 2018 в 23:48

0

Да нет, знакомтесь.

0xd34df00d 22 сентября 2018 в 00:07

0

Так это ж и есть браузер, оно не вне браузера.

xMushroom 20 сентября 2018 в 21:34

+1

Пишу на нативном WPF (хотя он тоже не нативный) и страдаю — на то, что в браузере можно сделать одним аттрибутом, тут уходит полдня. А ведь эти полдня я мог потратить на что-то более полезное…

shybovycha 20 сентября 2018 в 21:40

+1

вот именно этим и обоснованы абстракции вроде unity, electron и прочяя ересь — "чтобы разработчик думал о высоком, а не о низкоуровневой ерунде". именно об этом и весь сыр-бор

xMushroom 20 сентября 2018 в 21:46

+3

Ну вообще этим вся цивилизация обоснована. Разделение труда там, специализация и т.д.

shybovycha 20 сентября 2018 в 21:49

–1

это рептилоиды вас заставляют в это верить. на самом деле — все матрица, люди — батарейки, а мы с вами пилим кусок никому ненужного говнокода, лишь бы разделить специализированный труд =)

xMushroom 20 сентября 2018 в 21:52

+2

Если это был сарказм, то я его не понял.

shybovycha 20 сентября 2018 в 21:46

0

если грубо — то именно из-за лени программистов и нежелания разбираться в мат. части появляются уебшики, которым не нужно заморачиваться на всякие там управления памятью, оптимизации на уровне операций, да даже банальными алгоритмами и структурами данных. а потом у них жира тормозит и почта отъедает гиг памяти

staticlab 21 сентября 2018 в 00:30

+1

Кажется, Жира тормозит на стороне сервера с суровой джавой, и соискателей в Атлассиан собеседуют наверняка так же, как и везде, с вопросами о структуре хэшмапа и разворачиванием дерева.

polearnik 21 сентября 2018 в 11:12

+1

Делаете вы сайт ну прдположим на c++ Вначале все идет норм только постоянно отвлекаетесь на ошибки типа если пользователь введет не число а букву все упадет. Потом вдруг оказывается что пользователи жить не могут без пары графиков. Вы тратите кучу времени на то чтоб разобраться как рисовать графики. Потом оказывается что пользователи то хотят интерактивные графики. Вы снова тратите кучу времени на то чтоб запилить это. Потом оказывается что надо бы сменить дизайн или дать пользователямв возможность выбрать его. А у вас то он прибит гвоздями в коде.И вот в течении этих пары лет пока вы будете пилить свой довольно простенький сайт вы обязательн остолкнетесь с тем что ваше приложение-сайт работает только под виндой но пользователи хотят еще на телефонах его запускать Вы скажете что вы потратите много времени в первый раз а потом просто будете использовать примеры из старого кода. Возможно догадаетесь выделить примеры в отдельные файлы и даже делится с другими пользователями чтоб они потратили в след раз гораздо меньше времени. И даже сами будете брать библиотеки других чтоб ускорить процесс. Хотя погодите мне это чтото напоминает. и вы не учитываете еще тот факт что вы сами можете делать чтото неоптимально просто изза недостатка знаний или опыта.

P.S. Работал я с одним кадром который часто мечтал о том чтоб сделать сайт на ассемблере. Типа слушать 80 порт можно и на ассемблере. Делать запросы в бд тоже можно из асма итд. Вот бы все летало и жрало мало ресурсов



serbod 21 сентября 2018 в 11:38

0

В начале 2000-х так и было. Программы в Delphi / C++ Builder создавались из компонентов. Вставил мышкой компонент HTTP сервера, обработчиков HTTP запросов, подключения к базе данных, таблиц и запросов базы данных. Нарисовал мышкой визуальную форму с полями ввода, гридами, графиками, кнопками и панельками. В свойствах компонентов указал, с кем и как они связаны, где брать данные для отображения. И потом все это скомпилировал в файл и запустил. Получился stand-alone web server, который в браузере работает как настольное приложение, только хуже. Потому что тогда интернеты были вялые, браузеры туповатые, а дизайн делали не дизайнеры, а программеры.

Сейчас тоже так можно, причем в браузере будут не древние ActiveX скрипты с нативными формами, а модный Sencha Ext JS. Но ведь Delphi умер, вышел из моды, нет предложений на рынке.



shybovycha 21 сентября 2018 в 11:49

0

в вашем кейсе две проблеммы:

> Делаете вы сайт

и

> сайт ну прдположим на c++



shybovycha 21 сентября 2018 в 11:56

0

речь на самом деле не о крайностях вроде "если кодить — так только на Си!", а скорее о том, что приложения (даже сайты) можно писать оптимально (оптимальнее), не используя раздутых фреймворков, а решения делать не "самое простое — главное чтоб работало" (так, как выгодно бизнесу), а опять-таки, оптимально (обдуманнее).



PsyHaSTe 21 сентября 2018 в 00:49

0

Строго наоборот же, там, где в WPF можно обойтись двусторонним биндингом и сделать датасорс на коллекцию или даже БД в фронтент-разработке придется тащить и бекенд, и какой-нибудь реакт/ангуляр.

staticlab 21 сентября 2018 в 01:05

0

> и сделать датасорс на коллекцию или даже БД в фронтент-разработке придется тащить и бекенд

Очевидно, что приложение клиент-серверное, значит тащить бэкенд всё равно придётся.

> и какой-нибудь реакт/ангуляр

Но ведь фронтенд-фреймворки считаются злом. Что же делать? Писать байндинг на ванильном JS?



PsyHaSTe 21 сентября 2018 в 01:13

+1

Ну вот я не видел сайтов на ванильном JS, и надеюсь не увижу. Отказываться от фреймворка (вообще любого) это как отказываться от любого уровня абстракции. Нафиг ваш winhttp, я свою либу напишу. Нафиг ваш http вообще, я свой протокол напишу. Нафиг ваш TCP, я своё реализую…

Байндинг на ванильном JS не написать. А если начать реализовывать обертку, то получится багованный и кривой ангуляр или тот же реакт.

gearbox 21 сентября 2018 в 12:55

0

>Байндинг на ванильном JS не написать.
На Proxy и custom elements за пару вечеров на коленке ваяется аналог реакта. Но согласен — нафига, если есть реакт. Проблема то не во фреймворке.

xMushroom 21 сентября 2018 в 11:46

0

В этом смысле да, хотя если подтащить реакт/ангуляр, то снова становится нет. Но по больше части WPF куда бедней на возможности. Конкретно вчера я делал placeholder. :)

Neuromantix 20 сентября 2018 в 15:55

+4

Соглашусь как юзер, особенно как юзер разных девайсов, которые должны работать, но из-за софта, запихнутого в их МК не работают без регулярных перезагрузок. Термометр-метеостанция виснет каждые 3-4 дня, выбросил, бытовая техника, которая глючит, включает произвольные функции или тп. Какой ад ждет после увеличения числа IoT — боюсь представить.

BD9 22 сентября 2018 в 03:40

0

Это не всегда софт.
Был случай: глючил новый телевизор Sony — сбои программы. Зная про плохое питание, подключил его через стабилизатор. И глюки исчезли.
В общем, снизилось качество **всего** — попробуйте купить хорошую колбасу в РФ (и не только в РФ). Можно сделать качественно, но за это потребитель не хочет платить.

khim 22 сентября 2018 в 04:18

+1

> Можно сделать качественно, но за это потребитель не хочет платить.

Не «не хочет», а «не может». Постепенное усложнение мира в котором мы живём плюс усилия маркетологов привели к тому, что потребитель, зачастую, в магазине не может отличить хороший товар от плохого. А это, по большому счёту, приговор «свободному рынку»: свободный рынок может работать тогда, когда покупатель делает **осознанный** выбор… чего, по крайней мере на рынке потребительских товаров, уже давно не происходит…

Neuromantix 22 сентября 2018 в 09:04

+3

По поводу колбасы и выбора. Я сделал как-то сам колбасу из чистейшего мяса, специй и тп, по рецептам из дореволюционной книжки. Все, кого я этой колбасой угощал, ругались на то, что она какая-то не такая — не розовая, а темная, жесткая, пахнет перцем и тп, и после этого предлагали мне попробовать «нормальную» колбасу за 50р/палка из «магнита».

AmOralist 22 сентября 2018 в 10:23

0

> попробуйте купить хорошую колбасу в РФ

Легко, 750 рублей кг. И всего в 700 км от Москвы.

Osnovjansky 25 сентября 2018 в 15:17

0

Мой отец пару лет работал на колбасной фабрике на Кубани. В райцентре аграрного края. Колбаса, которую можно назвать хорошей, бывала только когда цыгане приводили ворованных коней. Как и из чего делалась обычная — лучше не знать и не видеть.



AmOralist 25 сентября 2018 в 15:49

0

Не, в той колбасе хотя бы видно из чего она сделана, в этом всё дело.
Но цена ее в 1,5 раза выше *обычной* сырокопченой в том же регионе и на точку ее берут не более 3 палок (0,75 кг) в день — иначе может не продастся.
Это в Мск такие цены даже на сервелат какой-нибудь встречаются (а та колбаса вроде как

продается в каких-то магазинах с фермерскими продуктами, но уже по совершенно другой цене).

kvasdopil 20 сентября 2018 в 15:57

+6

Можно бесконечно смотреть на три вещи: как горит огонь, как течёт вода и как ноет @tonsky. Хочется пожелать автору всё-таки со временем выйти из режима Grumpy Cat и начать снова получать удовольствие от любимого дела, мы все за него очень переживаем.

engine9 20 сентября 2018 в 21:13

0

Но дискуссия-то разгорелась первоклассная.

tonsky 21 сентября 2018 в 02:11

0

а автору-то как этого хочется :) но ничего, написание этой статьи позволило мне хотя бы для себя сформулировать, что меня вгоняет в депрессию и чего я хочу. Дело за малым — найти подходящий проект и подключиться

404

amarao 20 сентября 2018 в 15:57

+1

Цена абстракций. Если бы абстракций не было, вы бы использовали Initial Orders для того, чтобы загрузить вашу программу в двоичных кодах чтобы посчитать то, что нужно.

Самая первая жирная абстракция, которая случилась в компьютерах, заключалась в идее имитации интерактивности происходящего.

dernasherbrezon 20 сентября 2018 в 19:36

+2

Вряд ли. Каждый junior говорит мне одно и то же «оперативная память дешевле труда программиста». Потому что эту манцу проще говорить чем оптимизировать код и думать головой.

PsyHaSTe 20 сентября 2018 в 21:34

+5

Ну на то он и junior, тем более, что я тоже так новичкам говорю. А то есть у человека задача, а он начинает думать, что int для переменной цикла — это слишком жирно, я ведь только до 100 итерируюсь! Надо сделать byte. И идет срывать покровы, что программы такие большие, потому что ленивые разрабы где можно обойтись меньшим типом использовали больший. Да что там, сам я точно так же думал. И всех их учат обращать внимание на главное, а не на мелочи. Когда основы усвоены, то можно идти дальше.

Это примерно как в школе, где сначала долго и упорно заучивают, что из меньшего нельзя вычитать большего, а потом через год говорят «давайте вычитать», и ругать учеников, если они вдруг на старое правило ссылаться начинают.

С опытом приходит понимание (или нет. Но тут уже ничего не поделать), как расставлять приоритеты. А именно — никак. Уметь придумать несколько вариантов с разной стоимостью и дать бизнесу выбрать то, что ему больше подходит. «Денис, вот ты задачу создал, можно вот так вот закостылить, но скорее всего через год это выльется в такие-то проблемы ценой в 1М, а можно сейчас на пару недель сделать задачу на грамотное изменение архитектуры, тогда всё будет ок, но мне нужен будет Марк на пару дней для выработки плана». Главное, не сократить эту фразу до «Денис, вот ты задачу создал, можно вот так вот закостылить,, а можно сейчас на пару недель сделать задачу на грамотное изменение архитектуры и мне нужен будет Марк на пару дней для выработки плана». Разница ведь колоссальная, и как мне кажется большинство разработчиков используют вторую форму, когда подразумевают первую, а потом обижаются на логичный ответ менеджера «давайте по первому сценарию, заказчик уже месяц ждет». Сам виноват, надо было лучше объяснять.

Ну а если все равно костыли выбирают, значит бизнесу так нужнее, они вам с прибыли зарплату платят, они должны выбирать решения, вы эксперт который дает им полную информационную поддержку, но решения все же за ними.

Ну а если выбирают костыли, а потом говорят «блин, нам твой костыль в 1М встал, что ж ты собака сделал?!?», то 2 недели и поиск более адекватных коллег является оптимальным исходом :)

dernasherbrezon 20 сентября 2018 в 21:49

+1

У вас другой случай. Те примеры которые описаны в статье это как раз случаи когда суровые дядьки архитекторы принимают вполне осознанные решения. Вряд ли технологию редактора кода выбирал junior. А потом те же самые дядьки архитекторы пишут разгромные статьи «почему нашему бизнесу важно забить на использование памяти». Junior потом читают такие статьи и в силу своего опыта делают неправильные выводы.

eugene_bb 20 сентября 2018 в 21:43

+2

А каждому сеньору надо распечатать и повесить табличку когда НЕ-надо оптимизировать, а то за собой замечаю иногда что рука тянется соптимизировать пару строчек, а эти строчки делают IO-bound операции.

Latency Comparison Numbers (~2012)

L1 cache reference 0.5 ns
Branch mispredict 5 ns
L2 cache reference 7 ns 14x L1 cache
Mutex lock/unlock 25 ns
Main memory reference 100 ns 20x L2 cache, 200x L1 cache
Compress 1K bytes with Zippy 3,000 ns 3 us
Send 1K bytes over 1 Gbps network 10,000 ns 10 us
Read 4K randomly from SSD* 150,000 ns 150 us ~1GB/sec SSD
Read 1 MB sequentially from memory 250,000 ns 250 us
Round trip within same datacenter 500,000 ns 500 us
Read 1 MB sequentially from SSD* 1,000,000 ns 1,000 us 1 ms ~1GB/sec SSD, 4X memory
Disk seek 10,000,000 ns 10,000 us 10 ms 20x datacenter roundtrip
Read 1 MB sequentially from disk 20,000,000 ns 20,000 us 20 ms 80x memory, 20X SSD
Send packet CA->Netherlands->CA 150,000,000 ns 150,000 us 150 ms

braineater 20 сентября 2018 в 22:52

+1

Смотря как считать. Если нечто популярное использует много лишней памяти то по совокупной стоимости этой памяти можно купить не время работы программиста а всего программиста с потрохами.

khim 21 сентября 2018 в 00:30

0

Нечто популярное, типа кукого-нибудь Chrome, создаётся не одним программистов, а несколькими тысячи программистов. И тут уже неясно — что дешевле: заплатить за память или ещё за 100500 программистов…

braineater 21 сентября 2018 в 00:39

0

Правда ваша, к тому-же за программистов платит заказчик софта а за память — пользователи. Но тысячи обезьян программистов не всегда нужны. Инстаграму хватило несколько человек. Полагаю и в мире десктопов такие примеры имеются.

404

amarao 21 сентября 2018 в 13:02

+1

Хром не использует много памяти. Много памяти используют много сайтов. А купить такое количество веб-разработчиков… Ну, удачи.

404

amarao 21 сентября 2018 в 13:00

+4

В тот момент, когда вы согласились, что 'hello world' должно кодироваться с помощью числа 121404708502361365413651816 (а то и числа 31079605376604509545894864896), в этот момент вы уже были уверены в то, что оперативная память дёшева. С позиций разработчиков первых компьютеров, мой вот этот комментарий, в UTF-8 — это просто какое-то непростительное разбазаривание байтов. В такой объём памяти можно вместить достаточно переменных для того, чтобы рассчитать траекторию полёта на луну, а я тут объясняю, что память дешева.

На самом деле, основным потребителем памяти являются не программисты, а пользователи. Потому что пользователю всё равно сколько мегабайт занимает его аватарка — он хочет, чтобы она была красивой. Ему всё равно сколько миллиардов математических операций надо выполнить в секунду — ему хочется, чтобы страница скроллилась плавно, и чтобы в середине печати не возникало сообщения «превышена максимальная длина текста».

А ещё хочется переключать задачи мгновенно. И иметь быстрые сайты.

И всё это требует абстракций, о которых кусок кремния ничего не знает. Каждая абстракция — заведомо неэффективное использование уровня ниже (те же транзисторы в аналоговом режиме могут много больше, чем в цифровом — мы используем их самым тупым образом во имя иллюзии двоичного переключателя).

Неэффективность взамен предлагает простоту — и ради этого все эти гигабайты кода и памяти. Это просто, а значит, человек может это понять и начать использовать. Как только «просто» становится «сложно» на его базе делают следующий уровень неэффективности, чтобы снова стало «просто».

dernasherbrezon 21 сентября 2018 в 18:38

0

Я вот сейчас пытаюсь писать комментарий и делаю это быстрее чем текст появляется на экране.

К сожалению я не могу загрузить профайлер хрома. Но в посте где нет комментариев, текст набирается быстро и профайлер говорит что все хорошо. Зачем перерисовывать страницу когда я набираю текст в маленьком окошке? Это ведь не про размер букв в UTF-8.

404

amarao 21 сентября 2018 в 18:52

+3

И виноваты программисты, да? Вы в курсе, что «окошек» не существует, и что это всего лишь неэффективная абстракция над эффективным методом делать bitbit? Который, в свою очередь,

эффективная абстракция над возможностью копировать байты с одного места в другое? (Я понимаю, что в современном графическом стеке всё сложнее, но нам надо где-то остановиться).

Вот зачем вам эта неэффективная абстракция и весь этот bitblt? Почему бы не отображать символы сразу из растровых представлений в видеопамять?

Не, серьёзно, зачем вам окошки? Эти ужасные программисты тратят мегабайты на каждое окошко — если избавиться от окошек, то освободится огромное количество памяти.


Foror 22 сентября 2018 в 17:29

+1

>И виноваты программисты, да?
Да, только не браузера (хотя и браузер разрабатывают такие же олимпиадники), а олимпиадники хабра. Они возможно хорошие математики, но очень плохие программисты. Решение проблемы в данном случае заключается в удалении из DOM лишних комментариев, невидимых на странице. И этим страдает не только хабр, этим грешат очень известные западные веб-сайты, всем известных корпораций.

И причем тут вообще окошки?

EvilFox 22 сентября 2018 в 17:42

+1

> Решение проблемы в данном случае заключается в удалении из DOM лишних комментариев, невидимых на странице

Нет, решение для поле ввода заключается в чистке неиспользуемых стилей и возможно рефакторинга переусложнённых селекторов (их хватает неоптимальных), а прокрутка и вовсе чинится так.
Динамический viewport нужен только когда у вас реально много данных на странице (начиная с десятка тысяч, или даже больше), а тут на момент написания этого комментария всего 1369 комментариев.


Foror 22 сентября 2018 в 18:06

0

>Кстати я отписал таки в тп хабра, цитирую ответ: «Вы можете привести пример ресурса на котором статьи с 1300 комментариев загружаются на ваш компьютер без задержек?»
Ахаха, вот они наши олимпиадники) Я думал они просто от лени, а они даже не вдупливают, что вообще происходит)

P.S. Вы так или иначе придёте к удалению лишнего из DOM, поэтому мне непонятно ваше категорическое «Нет», с отметкой сколько нужно вешать в граммах.

EvilFox 22 сентября 2018 в 18:28

0

> P.S. Вы так или иначе придёте к удалению лишнего из DOM, поэтому мне непонятно ваше категорическое «Нет», с отметкой сколько нужно вешать в граммах.

И сколько предлагаете держать в «буфере» комментариев? всего сотню? (на сотне наверное не тормозит?) А знаете что при динамической видимой области перестаёт полноценно работать поиск по CTRL+F? То есть тут уже встают две задачи сделать динамическую область видимости (задача сама по себе не простая) и ещё эмуляцию поиска (который будет искать и в невидимых комментариях). А технический долг в виде перегруженной каскадной таблице стилей никуда не исчезнет.
Так-то я за динамическую область видимости, но только когда в ней есть потребность.


Foror 22 сентября 2018 в 19:05

0

>И сколько предлагаете держать в «буфере» комментариев?
То, что помещается на экран. Плюс держать еще несколько «сверху и снизу».
Плюс хранить все загруженные комментарии в виде объектов для подстановки на скроле.

Про поиск как-то не подумал. Для этой ситуации можно держать только TextNode на нужных позициях в DOM. Думаю не должно тормозить даже для десятков мегабайт текста (но это не точно). Один мегабайт точно не тормозит.

Еще это может решаться с помощью UI, на пикабу видел топорное решение для бесконечного скрола, путём нажатия нужной кнопки для очистки DOM.

Задача интересная, тут ресерч нужен, сам такое не делал еще.

>А технический долг в виде перегруженной каскадной таблице стилей никуда не исчезнет.
Можно исправить, я не спорю.

>задача сама по себе не простая
Так в этом и весь сыр бор, что программисты идут легким путём, а не решают проблему на корню.

sumanai 22 сентября 2018 в 19:27

0

И с полосой прокрутки сложно, особенно при изменении размера экрана. Плюс на Хабре куча кнопок для перехода на отдалённый комментарий, например, кликая по числу новых справа, можно довольно таки далеко скакать по странице.
В общем ИМХО, удаление элементом из DOM есть большой костыль.


rsi 27 сентября 2018 в 16:15

0

На мобильных платформах нормальная работа таблиц строится по принципу держим в памяти столько ячеек, сколько входит на экран. А при навигации куда либо, просто перестраиваем их. Это не так и сложно.

sumanai 27 сентября 2018 в 18:33

0

На мобильных нельзя непредсказуемо менять размер окна.


rsi 27 сентября 2018 в 18:48

0

Размер элемента в нем можно, что вызовет лайаут вложенных элементов. Вообще изменение размера окна это не такой уж и проект сайенс, просто надо пересчитать по цепочке легату для всех вложенных элементов. Это чистая математика.

sumanai 27 сентября 2018 в 19:01

0

Просто придётся делать это вручную на яваскрипте, вместо случая, когда это делает браузер на компилированном коде.


rsi 28 сентября 2018 в 05:56

0

Делай это браузер проблем бы небыли вообще. Но и вручную это сделать можно.

akryukov 28 сентября 2018 в 12:59

0

Можно повернуть набок. Изменение с портрета на ландшафт вроде бы предсказуемо, но переход заметен.

EvilFox 22 сентября 2018 в 19:35

0

> Для этой ситуации можно держать только TextNode на нужных позициях в DOM. Думаю не должно тормозить даже для десятков мегабайт текста (но это не точно). Один мегабайт точно не тормозит.

Да, можно попытаться выкрутиться таким компромиссом.

> а не решают проблему на корню.

Ну прямо на корню это в обозревателях должны решать — они сами должны выкидывать невидимое и отбрасывать все неиспользуемые стили при первой загрузке, оптимизируя их так чтобы не вычислять сложные правила каждый раз при перерисовке. В свежей лисе похоже так и работает (либо просто очень быстро работает), а вот в хроме нет.

Ещё минус динамической области видимости в невозможности пользователю сохранить страницу целиком. Нужно костылять спец. кнопку по которой всё дерево сгенерируется для удобства сохранения (кто об этом позаботится? только энтузиасты).


Alexeyslav 24 сентября 2018 в 13:50

+1

И вместе с этим «поиск на странице» превратится в «найди мне слово на экране».

Am0ralist 24 сентября 2018 в 17:21

0

При этом дома на стрингах уже на полутора тысячах не с первого раза стали прогружаться все комменты.
А на фаерфоксе — норм и шустренько.
У меня возникают подозрения, на каком браузере тестируют работу хабра…

khim 25 сентября 2018 в 18:55

0

Тем не менее 300 комментариев с LWN у меня открываются быстрее, чем 100 комментариев с Хабра. И я могу там увидеть сотню новых комментариев и ответить на них быстрее, чем увидеть на Хабре один комментарий и понять, что он меня не касается.

Очень наглядная разница между «ужасным интерфейсом 90х» и современным «отзывчивым» интерфейсом…

Foror 22 сентября 2018 в 17:01

+1

Абстракции были всегда. Проблема не в них.

Для прикладного ПО проблема в отсутствии инструментов для удобной и быстрой разработки. Ну нет сегодня аналога Делфи из начала 00-х. Есть две крайности сложная разработка на кростах с QT или простая, но очень жирная разработка на Electron. А по середине ничего нет! Ниша свободна.

В свою очередь в системном ПО скопился огромный пласт легаси. Тот же Линукс не может осилить 10 Гбит/с в стандартной поставке. Его приходиться патчить, накатывать сторонние костыли. Загуглите linux 100 Gbit/s, там всё еще веселее. И здесь полностью свободная ниша. Люди понимают о необходимости Unikernel, но не кому это пилить! Ниша свободна. И тут мы подходим к следующей проблеме.

Людей, которые смогут оптимизировать и переписать всё с нуля — можно пересчитать по пальцам. Просто нет специалистов, кто возьмётся за это неблагодарное дело. Сегодня легаси переплюнут авгиевы конюшни. Поэтому переписывать нужно очень многое, начиная с языков программирования и заканчивая операционными системами.

Emulyator 22 сентября 2018 в 17:13

0

Вот уже не думал, что кто-то посетует на нехватку инструментов для разработки. ИМХО, их больше, чем нужно, и, по отзывам, один удобнее другого.)

Foror 22 сентября 2018 в 17:30

0

Нет, это неправда или это был сарказм?

Foror 22 сентября 2018 в 17:32

0

Инструментов хватает, я даже перечислил лидеров рынка. Но остальные еще хуже.

Emulyator 23 сентября 2018 в 00:25

0

Нет, не сарказм, все так и есть, я кроме Qt да в.студии особо ничего не использовал, но читая тот же хабр, регулярно встречаю восторженные отзывы об очередном инструменте, плагине, редакторе, фреймворке и т.п., который чуть ли не сам всё за тебя делает. Глаза разбегаются, не понятно, что выбрать.)

Foror 23 сентября 2018 в 09:08

+2

>встречаю восторженные отзывы об очередном инструменте
А я встречаю восторженные отзывы о go, kotlin, но это ничего не значит. Люди по прежнему программируют на кростах, джаве и питоне. В go же хотят внедрять дженерики (а как пели, что они не нужны!). А с развитием джавы непонятно зачем нужен kotlin. По мне он повторит историю groovy.

То, что видите вы это маркетинг, действующий на неокрепшие умы. Я это всё очень давно мониторю и пока не видел технологии «который чуть ли не сам всё за тебя делает». Всё что есть на рынке отдаёт некомпетентностью и легаси.

Мне это настолько всё надоело, что я сейчас сам начинаю работу в сторону разработки нормальных инструментов для программистов.

>восторженные отзывы об очередном инструменте
Например?

Emulyator 23 сентября 2018 в 10:57

0

> Например?

Я серьезно не анализировал каждый случай, это скорее общее ощущение от «чтения заголовков». Ощущение, будто в короткий период появилось множество технологий, фреймворков, даже парадигм, новых версий сред разработки. Хлынуло в первую очередь из в веб направления: всякие ангуляры, реакты, метеоры, вуи, тайпскрипты, теневые домы, вебассембли, редаксы, нпмы, ноды.жс, какие-то стеки технологий, среды и плагины, все это поддерживающие и т.д. Я вебом не занимался никогда, но помню php и mysql всем хватало, а тут такое многообразие.)
Или взять Unity3D. Помню один из многочисленных школьников на ютубе слабо владеющий внятной речью, за пол часа накликал скелет для какой-то примитивной, но работоспособной игры. Я вообще в то время слабо представлял практические пути создания игры для мобилок, ну может глючные примеры на QML потрогал, а тут человек без особых знаний и навыков в программировании делает что-то рабочее. Вот и подумалось, это ли не «волшебная кнопка».)

Druu 23 сентября 2018 в 14:27

0

> Ощущение, будто в короткий период появилось множество технологий, фреймворков, даже парадигм, новых версий сред разработки.

Да всегда так было. Просто раньше — чуть медленнее. Сейчас вырос темп жизни, с ним вырос и темп, ну, вообще темп. Всего.

> Хлынуло в первую очередь из в веб направления: всякие ангуляры, реакты, метеоры, вуи, тайпскрипты, теневые домы, вебассембли, редаксы, нпмы, ноды.жс, какие-то стеки технологий, среды и плагины, все это поддерживающие и т.д.

Ну давайте-то будем честными — имея нормальную базу и опыт все это фигня и изучается влет.
Вот я умел в типы, упарывался функциональщиной, тыкал тот же хаскель палочкой — в итоге когда я пришел в команду, где писали на тайпскрипте, ту уже через две недели делал для используемых библиотек врапперы на суровой типовой акробатике, в которую половина людей из команды (с опытом год+ тса) тупо не могла.
Потому что я пролистал спеку тайпскрипта, увидел там везде знакомые термины из теории, и уже понимал как это можно использовать и в каком виде, не вдаваясь в подробности. Дальше немного практики, чтобы руку набить — и дело сделано.

Emulyator 23 сентября 2018 в 15:00

0

Да понятно, что все «охи и ахи» во многом просто возрастное бурчание. Даже помню как сам в юности посмеивался над дяденькой который на фортране под дос разрабатывал, измеряя скил программиста количеством строк кода которые тот за смену выдает. А я мышкой клепал формочки в MSAccess и считал, что время писать буковки скоро совсем уйдет.

Amöralist 23 сентября 2018 в 15:11

0

> Вот и подумалось, это ли не «волшебная кнопка».

Кнопка то волшебная, только кто сказал, что оно бесплатно достается?)

Druu 23 сентября 2018 в 14:26

+2

> Люди по прежнему программируют на кростах, джаве и питоне.

Вы так говорите, как будто кросты, джава и питон сами не развиваются. Мейнстримные языки в итоге аккумулируют в себе наиболее полезные фичи, которые отрабатываются на всяких котлинах, хаскелях и прочая.

khim 23 сентября 2018 в 16:31

0

> а как пели, что они не нужны!

Про дженерики разработчиками всегда говорилось «они полезны, но сильно усложняют язык, потому, может быть, в будущем».

Про бесполезность пели фанбои, многие из которых вообще на Go программировать не умеют…

CaSper 20 сентября 2018 в 15:58

+4

Прочитал статью с удовольствием, полностью согласен с автором. Тенденция печальная и думаю что она не изменится.
Людям поддерживающим эти идеи остается только проявлять активность, писать отзывы разработчикам, поддерживать материально такой софт.

yarric 20 сентября 2018 в 23:23

+1

Тут всё-же видится довольной скорый хэппи-энд — когда производители упрутся в физические ограничения чипов и программистам придётся снова вспоминать, чем цикл лучше мап-редьюса.

khim 21 сентября 2018 в 00:33

+1

Про этот «скорый» хэппи-энд вспоминали уже в конце прошлого века, а воз и ныне там. Flash пока только до 64 слоёв добрался, так что ещё есть куда расти… Ещё на полвека, я думаю, хватит.

Fedcomp 21 сентября 2018 в 12:14

0

Map + reduce могут быть такими же эффективными как и цикл.

yarric 21 сентября 2018 в 12:17

0

Цикл реализуется одной-двумя процессорными инструкциями, а сколько оверхеда тащат вызовы функций?

PsyHaSTe 21 сентября 2018 в 12:42

+1

А кто вам сказал, что там будут вызовы функций?

yarric 21 сентября 2018 в 13:56

0

А кто сказал, что нет?

PsyHaSTe 21 сентября 2018 в 15:05

0

Любой нормальный оптимизатор.

Вот есть хороший плейграунд: https://play.rust-lang.org/?version=stable&mode=release&edition=2015

Там слева вместо Run выберите show assembly и покажите вызовы функций core::slice::<impl core::iter::traits::IntoIterator for &'a [T]>::into_iter, core::iter::iterator::Iterator::map и прочих...

Druu 23 сентября 2018 в 14:38

0

А если функция будет виртуальной или фунаргом? :trollface.jpeg:

Fedcomp 22 сентября 2018 в 12:32

0

в компилируемом языке могут быть zero cost, по сравнению с циклом, как в Rust например.

swelf 20 сентября 2018 в 16:00

+1

Наверно когда ты не инженер и все эти аспекты низкой скорости принимаешь как данное, тебя не волнуют внутренние механизмы работы, то тебе просто хочется котика полайкать. А таких 99%. Но ведь вся эта раздутость в конечном итоге двигает развитие железа, позволяет его покупать по разумным ценам людям которым оно и правда нужно. Всякие научные расчеты, моделирование и все такое.

Ну и вобще, все просто хотят «красиво и удобно», что приносит в жертву «быстро и эффективно», а еще быстро и эффективно часто требует навыков, которые люди не хотят получать.

Если говорить в взбе, ничего ведь принципиально не изменилось с момента, что я помню, начало 2000х. Тот же текст, те же ссылки и картинки. Да теперь не надо качать видео и музыку, чтобы послушать/посмотреть. Но прожорливость браузеров выросла прилично. Все ради красоты и скругленных углов у div'ов?

Некоторое время назад из взб страниц начали делать полноценные приложения(взб2.0?). Занимательно, что для телефонов, все сейчас хотят насадить пользователю свое standlalone приложение взад.

Akon32 20 сентября 2018 в 16:09

0

> Если говорить в взбе, ничего ведь принципиально не изменилось с момента, что я помню, начало 2000х. Тот же текст, те же ссылки и картинки. Да теперь не надо качать видео и музыку, чтобы послушать/посмотреть. Но прожорливость браузеров выросла прилично.
>
> Тот же текст, те же ссылки и картинки.

Что?? Сейчас один CSS чего стоит. А ещё есть JS с сотней API (например, webgl, video, ajax). И JIT-компилятор JS. Можете взять старый интерпретатор и современный, и сравнить производительность.

novice2001 20 сентября 2018 в 16:20

+1

Проблема не в CSS, а в идиотском его использовании. JS сразу со всей сотней API в подавляющем большинстве случаев не нужен.
В подавляющем большинстве случаев как раз будет достаточно текста и картинок. Безо всякого JS вообще.

swelf 20 сентября 2018 в 16:57

+2

> Сейчас один CSS чего стоит

ну это и не принципиальное изменение, Сайты стали красивей? что изменилось то?

> А ещё есть JS с сотней API

только в нем нет необходимости на 99% сайтов.

Да, взб развивается, но посмотри просто на большинство сайтов, ты(собирательный образ пользователя) чаще всего читаешь какой-то текст с сайта. Необязательны для него тонны js и css. И даже не в них дело, процесс браузера рисующий страницу с 20к текста, парой картинок и немного js и css может жрать десятки и сотни мегабайт.

SirEdvin 20 сентября 2018 в 16:59

0

> ну это и не принципиальное изменение, Сайты стали красивей? что изменилось то?

Вы вообще сидите в интернете? Вот так выглядели сайты в 1996. Изменилось ли что-то с того времени?

swelf 20 сентября 2018 в 17:14

+2

Эм, нет? Картинки, текст, ссылки) Ну там дизайн так себе, но разве сегодня нельзя такой сделать и плохая верстка это показатель технологичности? Раньше нельзя было нейтральней сделать вырвиглазные цвета?
О том я и говорю, сделаем чуть чуть красивей и увеличим потребление ресурсов на порядки.

PsyHaSTe 20 сентября 2018 в 22:07

+2

thebestmotherfucking.website

Krapivnik 21 сентября 2018 в 13:47

0

И даже на нем аналитика PiWiK на js висит

НЛО прилетело и опубликовало эту надпись здесь

novice2001 21 сентября 2018 в 09:28

0

Ну так буквально недавно была статья. Где один сверхпопулярный сайт — википедия — прекрасно обходится без js вообще.

Kwisatz 22 сентября 2018 в 16:56

0

А если нужно страничку распечатать, да в ландшафте, то ничего за последние 10 лет не изменилось.

novice2001 20 сентября 2018 в 16:09

0

Если приносят в жертву «быстро», то «удобно» не получается никак. Если я после выделения фрагмента жму Ctrl-C, Ctrl-N и несколько секунд жду появления окна, это никак не назовешь ни удобством, ни эффективностью.
И вообще, вряд ли типичный потребитель вообще точно знает, чего он хочет. Иначе пришлось бы к прийти к выводу, что он хочет раздутое и тормозное ПО, коего сейчас большинство.

swelf 20 сентября 2018 в 16:47

0

> Ctrl-C, Ctrl-N

это я отношу к быстро и требует навыка, а удобно и не требует навыка это выделить мышкой, ПКМ,

либо меню -> копировать и тп.

novice2001 21 сентября 2018 в 08:20

+1

Вы, конечно, можете относить бело к черному, но сути это не меняет.
Только вот, если человек работает, то «удобно» — это именно «Ctrl-C, Ctrl-N», а не тыкание мышкой по пунктам меню.
Внезапно, «выделить мышкой, ПКМ, либо меню -> копировать и тп.» есть в любой программе, это не достижение новых тормозных версий.

swelf 21 сентября 2018 в 12:12

0

Пример немного выбивается из того, что я на самом деле имел ввиду. Не нормально конечно когда текстовый редактор делает копировать/вставить(что там ctrl-N?) несколько секунд. Не могу на вскидку привести норм пример. Ну я например вимом пользуюсь, но для большинства людей он имеет 2 режима: пикать и портить текст. Хотя мне например вполне удобно.

Речь скорей в целом об индустрии, да и не только, о жизни в целом, все просто хотят кнопку «сделать зашибись» и не важно что там за ней спрятано и сколько ресурсов понадобится для реализации, тогда приложив немного усилий можно самостоятельно получить этот результат намного быстрей.

novice2001 21 сентября 2018 в 12:17

0

Я бы понял, если бы действительно была кнопка «сделать хорошо», и она действительно делала хорошо. Тогда можно было бы смириться и с большим размером и с тормозами. Проблема в том, что такой кнопки нет, или есть, но хорошо она все равно не делает. Но аппетиты программ растут так, как будто они сами делают не просто хорошо, а прямо-таки превосходно.



SirEdvin 20 сентября 2018 в 16:26

0

Программирование — это не инженерная специальность. У нас как бы нет четкой математической базы или подходов в духе "молоток — для гвоздей, отвертка для шурупов", как и нет спецификаций или их игнорируют абсолютно все.

Вместо этого у нас куча универсальный инструментов, вкусовщина и хайп)

Silvatis 20 сентября 2018 в 18:09

+4

Внезапно, есть и методология подсчета сложности и оптимальные алгоритмы на все случаи жизни.

Но проблема в том, что в тройке скорость-качество-цена ныне принято вычеркивать второе.



Losted 20 сентября 2018 в 20:33

0

И, что характерно, на текущий момент это оправдано для большинства типов бизнеса, так как стоимость ошибок относительно низка в среднем по больнице. Хотя и бывают исключения типа падения Амазона в Amazon Day (или как его там). Процесс, при котором можно быстро поправить ошибку оказывается более эффективен процесса, при котором эта ошибка будет обнаружена до выката в прод.

А в областях, где качество более важно (медицина, космос, авиация и т.д.) и процессы другие, но просадка по скорости.

yarric 20 сентября 2018 в 23:35

+1

> просадка по скорости

Многие до сих пор на WinXP сидят — и ничего, всё летает, так что если бы новые версии выходили раз в 10 лет — большинство бы сказали «спасибо», поскольку не понадобилось бы ломать привычные паттерны использования из-за того, что компании срочно понадобилось обосновать продажу новой версии.



Losted 21 сентября 2018 в 16:18

0

Так тут вступает в действие другой механизм: стоимость поддержки и стоимость внедрения фичи. Иногда от легася надо избавляться чтобы внедрять новое



SirEdvin 20 сентября 2018 в 20:36

−6

Что, и правда есть? Я надеюсь вы сейчас не скажите мне про подсчеты через O(n), которые асимптотические и не применимы в реальности?
Или есть какая-то другая методология?

eugene_bb 20 сентября 2018 в 21:59

+3

Некоторое время назад прочитал интересную мысль, что программирование всё еще находится в эпохе ремесленичества, а не индустриализации.

Если принять это, то всё становится на свои места и разочарования и страдания выглядят надуманно.

Слишком молодая отрасль, плюс эпоха экспоненциального роста, а спрашивают с неё как со зрелой инженерии.

yarric 20 сентября 2018 в 23:29

0

Есть общая теория построения систем и кибернетика, вместо чего в мейнстримовом программировании какая-то псевдонаучная муть от гуру, типа шаблонов проектирования.



SirEdvin 20 сентября 2018 в 23:39

0

Общая теория построения систем абсолютно такая же муть только с другой стороны, потому что не учитывает практически ничего, кроме своих абстракций. В итоге на бумаге у них все круто, а как только приступаешь к реализации, оказывается, что в реальной жизни очень много подводных камней.

А если попытаться учесть все детали и сделать полную модель — вы в ней потоните скорее всего.

yarric 21 сентября 2018 в 00:02

+2

Тем не менее пока ничего лучшего не придумали: самолёты летают, спутники кружатся, процессоры считают.

staticlab 21 сентября 2018 в 00:35

+1

А уязвимости в механизме спекулятивного выполнения инструкций процессора — это «кибернетика» виновата или «шаблоны проектирования»?



CodeRush 21 сентября 2018 в 00:48

+3

Виноваты умолчания.
Никто не захотел думать (не знал, не стал заморачиваться, не влезал в строки, не влезал в бюджет, не влезал в целевую производительность, причины точные не знает никто), что процессор можно атаковать с этой стороны. Почему раньше никто не переживал по поводу того, что kernel.dll отображается по фиксированному адресу, а теперь это большая проблема? Почему никто не пытался предотвращать исполнения страниц памяти с данными, а теперь все стараются всеми силами? Почему аппаратура начала вести теневые стеки, подписывать указатели и тегировать динамическую память, а раньше об этом никто не думал?
Все потому, что программирование — оно в основном про то, как выполнить задачу бизнеса при определенном наборе входных данных и выдать относительно приличный результат за относительно же приличное время. Написание программы, работа которой возможна при любых входных параметрах, противостоящей утечкам данных из любых общих ресурсов, тайминг-атакам и всему тому, что появилось за 50 лет противостояния брони и снаряда — это совсем другая задача за совсем другие деньги.

BD9 22 сентября 2018 в 03:57

0

> Почему никто не пытался предотвращать исполнения страниц памяти с данными, а теперь все стараются всеми силами?

В старющем Фортране уже давно отказались от возможности принимать данные как код (и наоборот) (COMMON block вроде бы).
Другие огрехи так же были известны ранее — ничто не ново под луной.



SirEdvin 21 сентября 2018 в 00:51

−1

И причем тут общая теория построения систем? Сбои то все равно происходят, самолеты падают, спутники сходят с орбиты, в процессорах, как уже выше написали — дыра на дыре.

yarric 21 сентября 2018 в 02:57

0

Не так часто, как ворд. Да и много ли самолётов и спутников упало из-за программных ошибок?

VolCh 23 сентября 2018 в 01:00

0

Сколько в процентах падает самолётов и вордов?

khim 21 сентября 2018 в 03:06

+3

> в процессорах, как уже выше написали — дыра на дыре.

Вы действительное видите разницы между 4-5 ошибками, обнаруженными за 10 лет и примерно таким же количеством ошибок, которые в современных OS находятся за день?

ForexBroker 1 октября 2018 в 11:11

–1

А вы подсчитайте сколько в современном мире всего работающих OS? А что было бы, если бы такое же количество было спутников, которые выполняли тоже разнообразное количество задач? Мне кажется, что статистика была бы «близкой».



Kriger91 21 сентября 2018 в 11:15

+1

А толку-то от этого железа по разумным ценам, если я набирая в браузере этот коммент вижу как запаздывает появление текста? Где польза от 16Гб ОЗУ, 4 ядра Intel Core i7, SSD?

mayorovp 21 сентября 2018 в 13:19

0

И, что самое веселое, тут никакой js не виноват. Тормозит сам браузер, потому что ему забыли объяснить что содержимое textarea не влияет на размеры этой textarea, а значит и reflow всей страницы делать не обязательно…



Kriger91 21 сентября 2018 в 13:58

0

Не совсем понял при чём тут js(хоть он и противен естеству). Моя ремарка лишь про то, что при текущем подходе работа на компьютере не становится эффективней, а все достижения в железе нивелируются говнокодом и борьбой с «преждевременной оптимизацией». При этом, никому не хватает времени, чтобы сделать работу хорошо. Зато оно всегда находится, чтобы всё переделать. Дважды. А потом купить новое железо, потому что старое не тянет.



PsyHaSTe 21 сентября 2018 в 15:27

0

Почему не влияет? А если там есть какой-нибудь textarea:contains("special"){...}?

mayorovp 21 сентября 2018 в 15:30

0

Так его нет же…

IRainman 22 сентября 2018 в 13:29

+1

Напишите, пожалуйста, багрепорт авторам браузера. Я серьёзно.



0xf0a00 20 сентября 2018 в 16:16

+10

Я тут со своим Delphi в сторонке постою)
Вообще полностью согласен с автором, но войну добра и бабла всегда выигрывает бабло.



Am0ralist 20 сентября 2018 в 23:25

–1

Lazarus. Тот же самый по сути проект без шаманства вида «как собрать в размер дельфи» занимал раза в три больше места.
Ну и вообще, утилита уровня хеллоу ворд на 3,5к строк — 20 мб…
Не, знаете, как-то все грустно…



PsyHaSTe 21 сентября 2018 в 00:57

+1

Может дело было не в бобине? Вот тут на примере раста объясняется, но применимо к любым бинарям.



Am0ralist 21 сентября 2018 в 11:56

0

Конечно, дело во мне, даже не спорю. Я ж не знаю, что и как мне надо делать, если пишу утилиту а-ля хеллоуворлд (на самом деле было — чтение текстового файла, немного простой математики и работа с сервисом ООо, плюс графическое окошко уровня — прямоугольник серый, парочки полей для выбора, парочка кнопочек).
Вот только просто понять, что куда ткнуть и что отключить, дабы оно не весило столько, сколько весило вначале… Ну то есть я должен об этом — задуматься. А потом, получив все таки те же 20 метров, что были на делфи у того, кто проект начинал, я должен еще разок задуматься и пойти куда-то оптимизировать еще больше…

Но вот я подключил десяток библиотек, типа с сложной математикой. Вызвал по паре раз из каждой по одной функции (дабы не пытаться написать свой не правильный велосипед), но в итоге что попадет в скомпилированную программу? Одна функция из библиотеки (ну или несколько, если та вызывает еще что внутри) или вся библиотека?



PsyHaSTe 21 сентября 2018 в 12:52

0

с LTO — одна функция.



Am0ralist 21 сентября 2018 в 14:36

0

Ну вот а под lazarus советы по уменьшению — это утилиты strip и UPX к *.exe применить полученному (ну ладно, там есть отдельно несколько настроек в IDE), но до сжатия delphi у меня все равно тогда выигрывал… про LTO для него не встречал.

Но вообще это всё взгляд именно абсолютно новичка. Нужно было кое-что — сделал, поужасался размерами, чуть исправил но всё равно…



PsyHaSTe 21 сентября 2018 в 15:25

0

Собственно, я могу рассказать, что дебаг-сборка моего приложения на расте (небольшой бот на для телеграма) занимал 150мегабайт, скомпилированный в релиз с lto — 6 мегабайт… Так что действительно техника очень полезная. И можно действительно получить гигантский исполняемый файл, где непонятно, куда он вообще столько ест :)

Druu 23 сентября 2018 в 14:39

0

> скомпилированный в релиз с lto — 6 мегабайт…

Кек, целых 6 мегабайт какойто бот. А если бы этот бот был на ноде и весил тех же целых 6 мб интересно что бы сказали товарищи вроде автора обсуждаемой статьи? :)

khim 23 сентября 2018 в 16:36

0

> А если бы этот бот был на ноде и весил тех же целых 6 мб интересно что бы сказали товарищи вроде автора обсуждаемой статьи? :)

Что если сравнивать боты на разных языках, то нужно не забывать и про райтам? Сколо-сколо занимает контейнер, который может бота на node.js поднять? На этом фоне размер самого бота уже мало что значит, извините.

Druu 23 сентября 2018 в 16:42

0

Давайте еще ОС тогда посчитаем? Бот же не в воздухе запускается? :)

khim 26 сентября 2018 в 22:33

0

Несомненно если эти боты работают под разными OS, то нужно сравнивать ещё и OS, да. Если же под одной — зачем сравнивать то, что не меняется?

PsyHaSTe 23 сентября 2018 в 16:50

0

Ну самодостаточный бинарь и скрипт — все же разные вещи. А то так можно и на дуднете написать 100кб приложение, и забыть про гиг фреймворка на винте.

0xf0a00 21 сентября 2018 в 09:28

+1

Настройками проекта, что в Delphi, что в Lazarus можно выкинуть лишнее и файлы станут нормального размера.

serbod 21 сентября 2018 в 10:53

+1

Даже если выкинуть отладочную информацию из экзешника, современный бинарник стал «толще» за счет эволюции базовых библиотек, рассчитанных на кросс-платформенность и unicode. Но, при этом сохраняется совместимость со старыми версиями ОС и железа, что требует дополнительных проверок и «оберток». В любом случае, по объему это несравнимо с Android Support Library, которая идет в составе современных андроид-приложений.

third112 25 сентября 2018 в 17:37

0

> Я тут со своим Delphi в сторонке постою)

Разрабатываю алгоритмы и т.о. Delphi-7 под Win XP для меня оптимален. Для web использую Knoppix. Остальные ОС под VM.

SirEdvin 20 сентября 2018 в 16:21

+9

Читаю статью и комментарии и мне кажется, люди катастрофически не представляют, как весело программировали в 90-х и какие были веселые проблемы.

ОС за вас ничего не делает? Вы представляете сколько костылей, оптимизаций и драйверов предоставляет для вас ОС, что бы вы могли работать независимо от железа и не зубрить спеки? Сравнивает современные редакторы кода и древний emacs? Я вам напомню, что emacs работает в терминале все-таки :)

Функциональность не растет? Вы вообще пишите хоть что-нибудь? Возьмите любой продукт из 90-х, он написал ужасно, выглядит ужасно, и работает так себе. Современный софт содержит кучу анимаций, обновлений в фоне, вебсокетов и прочего. Автор сетует на то, что "мы ожидаем, что никто не поменяет данные пока пользователь смотрит страницу"? Еще лет 15 назад это была аксиома.

Касательно машин это очень прохладная история, хотя бы потому, что КПД двигателей внутреннего сгорания все-таки ниже 100%.

Ну и касательно веса — большая часть веса приложений это чаще всего ресурсы. И что вы предлагаете с ними делать?

0xf0a00 20 сентября 2018 в 16:33

+5

Клавиатура гугл — 153Мб, Smart keyboard Pro — 5,9МБ. Расскажите какие же там ресурсы напиханы что разница в почти 30 раз? Карл 30 раз! Естественно я использую смарт клавиатуру, она не лагает, не занимает много места, у нее лучше оформление чем у гугловой(внезапно). Серебряная пуля для меня среди клавиатур на телефон, нету этих смайликов и прочего го*на, разрешения требует только те что реально необходимы (6 штук всего)… просто клавиатура и все.

SirEdvin 20 сентября 2018 в 16:39

+1

1. Возможно, я не прав, но клавиатура гугл весит всего 30Мб.
2. Ну какие-то ресурсы. Папка с ресурсами занимает 50% веса.

```
57812  .
27736  ./res
17616  ./res/xml
13960  ./lib
13956  ./lib/x86_64
8220   ./resources.arsc
6940   ./lib/x86_64/libjni_delight5decoder.so
6384   ./classes.dex
2700   ./lib/x86_64/libhmm_gesture_only.so
2336   ./res/layout
2216   ./lib/x86_64/libmozc.so
2096   ./lib/x86_64/libhwrword.so
1828   ./res/raw
1480   ./res/raw/third_party_licenses
1056   ./res/drawable-xxhdpi-v4
1016   ./res/drawable
988    ./res/drawable-xhdpi-v4
968    ./res/drawable-hdpi-v4
932    ./META-INF
456    ./META-INF/MANIFEST.MF
```

0xf0a00 20 сентября 2018 в 16:58

0

Я ведь предоставить пруфы всегда рад. Да каюсь, не открывал детальную информацию о приложении, просто в меню посмотрел, не 153, а 134, но блин на эту разницу 4 клавиатуры влезет)

▸ GBoard

▸ Smart keyboard pro

AlexZyl 20 сентября 2018 в 17:20

0

У меня слегка иная картина: те же 130мб, но приложение — 75, данные — 50

0xf0a00 20 сентября 2018 в 17:28

0

У меня из приложений от гугл только клавиатура(которую только что удалил) и хром. Возможно сказывается отсутствие гугл плея и всего что он с собой тянет.

SirEdvin 20 сентября 2018 в 20:50

0

Ну, вот я привел ссылку на apk, которое вести 30 МБ, в распакованом виде 57.

Мне кажется, лишний вес может появится только из-за файлов ресурсов, разве нет?

0xf0a00 21 сентября 2018 в 09:34

+2

Да что вы молитесь на эти «ресурсы»… это универсальное слово которым можно все оправдать? Это так не работает. Конкретно, по факту что там может быть? Каждая кнопка клавиатуры в разрешении 4к? Любимый фильм лида?
Понятия не имею, что это за сайт, что там за апк и почему, я вам привел скриншот официального приложения установленного из плэй маркета.

PsyHaSTe 21 сентября 2018 в 12:54

0

Ну например картинка с наложенным на неё красивым отфотошопленным текстом на каждом языке. Я i16n никогда не занимался, но например когда я смотрел размер дума, в нем 40 гигабайт из 60 как раз занимали ресурсы — текстуры, звук, ...

0xf0a00 21 сентября 2018 в 15:28

−1

Ммм… сравнивать игру ААА класса на продажи которой напрямую влияет качество и количество контента и клавиатуру, умно умно, но это не точно…

PsyHaSTe 21 сентября 2018 в 15:35

0

Никто не сравнивает, вам говорят, что картинки и звук *могут* занимать столько места.

Например, локализованный логотоип на всех языках.

0xf0a00 21 сентября 2018 в 15:59

0

30 раз.
ваш дум весил бы 620гб, а не 60.
больше холиварить я не намерен.

Akon32 20 сентября 2018 в 16:53

0

> Серебряная пуля для меня среди клавиатур на телефон, нету этих смайликов и прочего го*на, разрешения требует только те что реально необходимы (6 штук всего)… просто клавиатура и все.

Возможно, вам смайлики совсем не нужны, но видимо кто-то посчитал, что это киллер-фича. Их даже в юникод впихнули. И какой-то части пользователей они полезны.

0xf0a00 20 сентября 2018 в 17:05

+4

Да да вот именно по такому принципу и идет теперь разработка ~~ювите его~~ может кому то нужно, а может и не нужно, пихай всем, ***** что качество софта упадет… главное здесь и сейчас пользователя окучить.

Saur4iG 20 сентября 2018 в 18:26

+3

А может остальным это и нужно? Вы можете не поверить, но никто не будет использовать приложения без анимаций и красивостей…
Или просто так в прошлом году apple на презентации какую показывали?) Люди хотят видеть подобное и им полностью плевать сколько памяти это занимает и как работает под капотом

prs123 20 сентября 2018 в 18:31

0

Более того, если вы будете выдвигать приложение в AppStore, то без всех этих фишечек, анимаций и красивых иконок ваше приложение скорее всего и модерацию то не пройдёт

sumanai 20 сентября 2018 в 22:12

0

> модерацию то не пройдёт

Серьёзно?

prs123 21 сентября 2018 в 11:12

0

Более чем. Существует огромный ворох т.н. гайдлайнов, в которых это все описывается. Ручная модерация в основном и занимается проверкой соответствия этим гайдлайнам

Antidipressant 21 сентября 2018 в 09:29

0

А Вы в курсе, как сейчас молодежь называет людей, которые вместо эмоджи используют смайлики в виде скобок? Если нет — погуглите :) А то не хочу выражаться :)

vis_inet 21 сентября 2018 в 11:53

0

Заинтриговали!)

InterceptorTSK 21 сентября 2018 в 12:25

0

А вы в курсе как называют молодежь в любое время? Хоть щяс, хоть 100 лет назад хоть 300… И да, гугол тут не нужен вовсе…

Antidipressant 21 сентября 2018 в 12:31

0

Какое это имеет отношение к теме?
Мой посыл был в том, что сейчас «быстрое» приложение «без рюшечек» основной массе не интересно. По этому, то что у Вас будет, к примеру, простой мессенджер, который умеет только отправлять текстовые сообщения и ничего больше, без анимаций с самым простым дизайном — вряд ли им будет пользоваться широкий круг людей.
Ах да, забыл, без аналитики вам будет очень и очень сложно развивать продукт в «правильном» направлении. А если еще у Вас основной бизнес строится по на рекламе, то дополнительный, скрытый функционал приложения может быть толще самого приложения.

IRainman 22 сентября 2018 в 13:26

+1

Я вообще использую Hacker's Keyboard, там вообще чуть больше 1 МБ и удобные для меня фичи.

novice2001 20 сентября 2018 в 16:41

+1

Да-да, вот именно, куча анимаций и прочей фигни, которые не имеют к основному функционалу приложения ни малейшего отношения, зато увеличивают его вес и тормоза при использовании.

SirEdvin 20 сентября 2018 в 16:45

+2

Да, особенно real-time обновления не имеют никакого отношения к функциональности приложение. И удобные оповещения, наглядная навигация тоже.

0xf0a00 20 сентября 2018 в 17:15

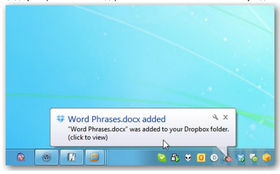
+1

Давайте манипулируя абстракциями дойдем до того что в любое приложение надо будет впирать функционал целой ОС и еще чуть чуть.

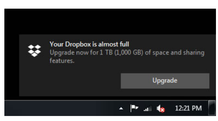
JTG 20 сентября 2018 в 18:54

+19

Тут вот товарищи из Dropbox как раз завезли удобные оповещения. У меня есть ноутбук с Windows 7, где у Dropbox была стандартная плюшка с баблом в области уведомлений:



Но потом, слава автообновлению, подвезли версию, в которой, наколхозили некий «polyfill», на который наверняка потратили кучу человеко-часов, и пытающийся маскироваться под уведомления в стиле Windows 10



Давайте же посмотрим как улучшился UX. Было:
— Крестик для закрытия
— Уведомления подчинялась настройкам уведомлений системы
— Кнопа для отключения надоедливых уведомлений была прямо на бабле
— Бабл закрывался по таймауту
— При уведомлении об изменении файла, клик открывал папку Dropbox с выделенным файлом

Стало:
— Крестика нет (Возможно, он появится при наведении? Но это не точно)
— Уведомление висит и не закрывается
— После некоторых обновлений это колхозное уведомление начинало перехватывать фокус. Может выбросить из полноэкранного режима
— При уведомлении об изменении файла, клик больше не открывает не папку. Теперь запускается браузер с сайтом Dropbox

Разумеется, это частный случай, но очень показательный. Десктопный софт стремительно тупеет.

SirEdvin 20 сентября 2018 в 20:43

0

Хм… опять же, а причем тут программисты? Думаете, они такие "а, поломает то, что неплохо работает без видимой на то причины"?

А если заказчик сказал "должно быть на всех устройствах одинаково", то тут уже ничего не попишешь.

novice2001 21 сентября 2018 в 09:33

+3

А здесь неважно кто именно виноват, важно что под видом полезных обновлений продвигается неведомая и никому кроме менеджеров-идиотов ненужная фигня. И так не только в дропбоксе, а везде.

SirEdvin 21 сентября 2018 в 11:26

−2

А вы вот прямо уверены, что не нужная? Или не нужная именно вам, так как вы не попадаете в ЦА этой фичи, а, скажем, 60% пользователей эта фича как раз то, что нужно?

novice2001 21 сентября 2018 в 12:34

0

А вы можете доказать что нужная и именно 60% аудитории?



SirEdvin 21 сентября 2018 в 12:47

–1

Я предполагаю, что такая крупная компания как Dropbox не делает что-то просто так, они считают что это или снизить издержки или повысит прибыть.

Вы не согласны?

Ну и доказать, что что-то нужно какому-то проценту аудитории возможно только постфактум.

novice2001 21 сентября 2018 в 13:21

0

Я предполагаю, что любой компании кажется, что когда она что-то делает, то она сделает это обязательно для пользы. Но «кажется» и «на самом деле» — это очень разные вещи, иначе компании не разорялись бы.



SirEdvin 21 сентября 2018 в 13:38

0

А вы можете предложить что-то другое? Принятие решений в условиях неопределенности это довольно сложная задача и там можно выбрать только самый оптимальный результат, а не лучший.

novice2001 21 сентября 2018 в 13:59

+1

Так и не надо приводить это в качестве аргумента. А то в одном комменте вы пишете «раз компания это сделала, значит они считают, что это поможет», а потом «ну так получилось, другого способа-то нет».
Другого способа конечно нет, но именно поэтому сам факт принятия того или иного решения никак не значит, что оно будет полезно хоть кому-нибудь вообще. Самый эпичный пример — кинопоиск 2.0.
Не надо делать всякий бред и говорить «смотрите, вот как хорошо!». А когда аудитория говорит «Нет, не хорошо», ей отвечают «Вы дураки, у нас тут дезигнеры, они лучше знают как для вас лучше».

IRainman 22 сентября 2018 в 13:20

+1

> Десктопный софт стремительно тупеет.

Потому что он не десктопный, что Dropbox, что Skype 8 типичное браузерное наколенное поделие которое тащит за собой ещё и вебкит для работы.



engine9 20 сентября 2018 в 17:35

+4

Тут еще и безграмотные (или просто ленивые) дизайнеры руку прикладывают. Иногда расковыриваю чужие иконки или 3Д модели, так они часто бывают крайне избыточные.

Недавно переделывал 3Д модель которая должна рендериться в размер 320x640 пикселей, по причине того что заказчик жаловался, что она как-то тормозит. Так мой предшественник сетку из автокада вывел и прямо загрузил чуть ли не 1 миллион полигонов. Даже не *заморачиваясь*.

Час работы и ужал лишнее до тысячи, не потеряв ни капли в гладкости рендера, зато фпс вырос в разы и скорость загрузки.

Что-то мне подсказывает, что все эти анимационные свистоперделки делают аналогичным образом. И палец о палец не ударят чтобы оптимизировать вектор.

Mabusius 20 сентября 2018 в 18:06

0

Был у меня проект на Зенде первом. И были там очень заковыристые хлебные крошки с которыми я ковырялся очень долго пытаясь заставить их быть такими как надо. И где то через год наконец дошли у меня руки посмотреть откуда же идут исходные данные для этих крошек… Оказалось, что мой предшественник забил на хлебные крошки, которые у него были, и написал новые с нуля. Я пошел глубже и оказалось, что и его предшественник забил на хлебные крошки, которые поставлялись зендом, и написал свои с нуля. При этом ни один из уровней хлебных крошек даже не был удален, типо «работает — не трогай». Тоесть крошки генерировались три раза… Я тоже не стал ничего исправлять, так как уже и так собирался менять работу :).



prefrontalCortex 20 сентября 2018 в 18:02

+4

> Я вам напомню, что emacs работает в терминале все-таки :)

Это один из режимов работы. У emacs есть режимы работы в оконных окружениях (X11, Windows).



JTG 20 сентября 2018 в 18:14

+23

> Возьмите любой продукт из 90-х, он написал ужасно, выглядит ужасно, и работает так себе

Продукт из 90-х будет либо шедевром под MS-DOS, либо (за редким исключением) полным шлаком под Windows.

Нет, давайте лучше возьмём из 2000х. Totalcmd, KeePass, PuTTY, WinAMP, позже Foobar, Notepad++, WinRAR/7-Zip, WinHEX… У меня до сих пор стоит Office 2003 (с Compatibility Pack для поддержки OOXML) потому что, мать его, он работает и выполняет мои задачи. Это всё примеры «доделанного» и по-настоящему утилитарного софта, который продолжает хорошо делать то, для чего был написан, а не морфировать значок квадратика в крестик на GPU. Так ли ужасен был QIP 2005? Чем была плоха возможность настройки внешнего вида приложения, изменения размера шрифта, количества сообщений на странице в мессенджере? Что мы получили взамен — возможность выпускать новую версию каждую неделю (зачем?) и постить в чат фрагменты мультиков в формате mp4?

P.S. Ещё такое наблюдение

Раньше:
— Любой школьник мог взять Delphi 7 и с минимальными усилиями сделать windows-приложение, которое ужасно выглядит и медленно работает.
— Средний программист мог взять Delphi 7 и с минимальными усилиями сделать windows-приложение, которое нормально выглядит и нормально работает.
— Хороший программист мог взять Delphi 7, замочиться, и сделать windows-приложение, которое хорошо выглядит и хорошо работает.

Сейчас:
— Любой школьник может взять Electron, неделю смотреть видеоуроки и настраивать тулчейны, потом сделать приложение, которое хорошо выглядит и медленно работает под любой платформой.
— Средний программист может взять Electron и с минимальными усилиями сделать приложение, которое хорошо выглядит и медленно работает под любой платформой.
— Хороший программист может взять electron, замочиться, и сделать приложение, которое хорошо выглядит и медленно работает под любой платформой.
— Компания Microsoft может взять Electron, замочиться, и сделать приложение, которое ужасно выглядит и медленно работает под любой платформой.



Neikist 20 сентября 2018 в 18:17

+5

Ну справедливости ради я знаю единственную программу под электрон которая работает более менее нормально. И она именно от майкрософта. VSCode. Правда я слышал что натива там очень немало.

Alexey2005 20 сентября 2018 в 18:41

+1

Хороший программист и сейчас может создать быстрое приложение на Electron. Для этого в общем надо сделать всего две вещи.
Первое. Отказаться от DOM, с которым связана львиная доля тормозов. Хотя все современные фреймворки создают «буферную» замену DOM, обращаясь к «реальным» объектам только где без этого совсем никак, это всё равно медленно.
Фактически, нужно просто взять один большой Canvas во всё окно и рендерить всё что нужно на нём вручную. Причём процедур рендеринга нужно написать несколько, под разные виды устройств (в зависимости от того, какие операции попадают под хардварное ускорение).
Второй источник тормозов (и заодно зверского потребления памяти) — сборка мусора вместе с автоматическим управлением памятью. Чтобы решить эту проблему, разработчик должен великолепно знать, как осуществляется управление памятью в Electron и, пользуясь этими знаниями, написать на JS собственный менеджер памяти. Как? Ну, почти так, как это делает Emscripten: создать ArrayBuffer и повозиться со всеми этими типизированными массивами, представлениями и блобами. Причём проблемы возникнут ровно те же, что у плюсовиков: утечки и ситуации, когда указатели внутри этих буферов внезапно начинают указывать куда-то не туда.
Конечно же, стандартные фреймворки, которые не способны работать в таких условиях, сразу же идут лесом — вместо них придётся колхозить своё. В том числе грамотно написать обработчики событий, чтоб всё не вставало колом, а это опять же квалификация.
Но ведь кодер, способный на такое, он и стоить будет как плюсовик. А в чём тогда смысл? Electron нужен-то как раз для того, чтобы любого индуса после трёхмесячных ускоренных курсов можно было посадить за код, и созданный продукт даже как-то работал.

picul 20 сентября 2018 в 19:09

0

А можно пожалуйста ссылку на мир, в котором плюсовик стоит дороже, чем расфуфыренный JS-developer? Я вот ни разу такой картины не наблюдал.

Alexey2005 20 сентября 2018 в 20:17

0

www.itjobswatch.co.uk/jobs/uk/c++.do
www.itjobswatch.co.uk/jobs/uk/javascript.do
Сравните Salary trends и соответствующие гистограммы на этих страничках. Даже средненький C++ кодер — это порядка £50к, что несколько дороже среднего JS-кодера (£40к). Топовая часть гистограмм говорит сама за себя.

picul 20 сентября 2018 в 23:22

0

Ну Вы и сократили, там средняя вроде 47.5К для JS и 52.5К для C++ — разница мизерная, как по мне. В топовых — да, по-больше, но что-бы переплюнуть JS много топовых плюсовиков не надо)) Так что в общем случае переплаты минимальны, если вообще присутствуют.
Не холивара ради, но тыкнул по Java — дак они вообще больше плюсовиков получают. ИМХО вот где первые кандидаты на реорганизацию.

Alex_ME 21 сентября 2018 в 03:10

+4

Етижи-пасатижи, куда мир катиться?!
Т.е. мы берём веб с кучей средств рисования, ресайза под размер итп итд и выкидываем нафиг, начиная все контролы и прочее с нуля рисовать, мышку обрабатывать итп, как в старые времена. У и ради чего? Ради электрона ради электрона? Ну так взять окно с опенгл и рисовать.

SirEdvin 20 сентября 2018 в 21:08

+2

> Нет, давайте лучше возьмём из 2000х. Totalcmd, KeePass, PuTTY, WinAMP, позже Foobar, Notepad++, WinRAR/7-Zip, WinHEX

PuTTy — лютый шлак с максимально не очевидным интерфейсом. Людям, которые к нему привыкли может быть и ок, но посмотрите на интерфейс, например, Termius.

По остальным инструментам вы совершаете простую ошибку — они служили исключительно для одной цели и делали ее хорошо. И сейчас такого софта навалом, правда. А вот когда нужен софт, который будет делать много разных штук из коробки, тогда начинаются проблемы.

> У меня до сих пор стоит Office 2003 (с Compatibility Pack для поддержки OOXML) потому что, мать его, он работает и выполняет мои задачи

Скорее всего, вы просто используете его как продвинутый блокнот без какой либо серьезной верстки или макросов, я прав?

> Так ли ужасен был QIP 2005?

Отвратител. К вопросу что мы получили взамен — я получит возможность использование ChatOps, интеграции с мессенджерами вплоть до решение инцедентов прямо из них. Нужно ли это? Еще как.

P.S. Вы и сейчас можете взять делфи, он все еще живой. Что ж не берете то?

sumanai 20 сентября 2018 в 22:20

+1

> А вот когда нужен софт, который будет делать много разных штук из коробки, тогда начинаются проблемы.

Так может такой софт на самом деле и не нужен? Зачем календарю возможность вставки эмодзей и видео в 4к?

SirEdvin 20 сентября 2018 в 22:25

0

Она не нужна. И есть прекрасные календари без нее.

Но вот тот же слак, который тут ругают содержит безумное количество различных полезных вещей, которые люди просто не привыкли использовать и это исключительно их проблемы.

И опять же, електрон они выбрали только по одной причине — потому что пилить еще отдельно десктоп очень дорого.

picul 20 сентября 2018 в 23:27

0

А мне не дорого отваливать деньги за приложение, окно которого открывается не быстрее, чем за 5 секунд?

PsyHaSTe 21 сентября 2018 в 00:59

+5

Не отваливайте

SirEdvin 21 сентября 2018 в 01:03

0

Не открывается — всмысле не запускается или каждый раз вы 5 секунд ждете?
Если каждый раз — у вас что-то не так.
Если при первом запуске — ну, 5 секунд можно и подождать на старте компа, для меня его функционал в целом перекрывает эту мелкую проблему.

picul 21 сентября 2018 в 02:54

0

Имею в виду щелканье по иконке в трее для того, что-бы развернуть окно. Окно то разворачивается, но в течении нескольких секунд рисует всякие овалы вместо контента.
Справедливости ради, возможно на счет 5 секунд я и перегнул, но задержка в 1-3 секунды есть всегда. Зато иногда мы просто зависаем, и что-бы увидеть полученное сообщение приходится убивать процесс и запускать заново.
Комп не топовый, но и не старье, работается вполне нормально.

ValdikSS 21 сентября 2018 в 02:09

+7

> P.S. Вы и сейчас можете взять делфи, он все еще живой. Что ж не берете то?

Вы это серьезно, без сарказма? Я считаю, что лучше Delphi все еще ничего не придумали для разработки графических программ под десктопы.
Во-первых, все компилируется в нативный код с минимальными накладными расходами.
Исполняемый файл получается маленького размера, запускается моментально, работает быстро.
Во-вторых, на Delphi можно писать под Windows, macOS, Android, iOS, и даже веб, используя стандартные компоненты. Для веба Delphi-код компилируется в Javascript-код.
В-третьих, если вам нужно написать какую-нибудь простую программу и быстро, в течение, скажем, 20 минут, Delphi — идеальный вариант. Qt, C#, Mono, Java просто не сравнятся. Такое чувство, что RAD Studio сделана для людей, а все остальное — для свиборгов.

PsyHaSTe 21 сентября 2018 в 02:17

0

Хмм, каким образом C# (который исторически "более лучший" дельфи, скрещенный с плюсами и подглядывающий за джавой) менее удобен, чем дельфи? Не, я могу понять про сложное десктоп-приложение, где под Дельфи наверное тысячи компонентов придумано, но вот как раз простенькая тулза на шарпе напишется в полминута. Удобные *новые* пакеты, linq, IDE явно удобнее RAD Studio (хотя я там не очень много работал, но в студенческие годы все же прилично посидел в них), ну и может еще что-то, но смысла перечислять не вижу, этого уже достаточно, чтобы сделать быстрое и удобное решение.

> Такое чувство, что RAD Studio сделана для людей, а все остальное — для свиборгов.

Расскажите подробнее, что именно вас смущает?

khim 21 сентября 2018 в 02:34

+2

> Хмм, каким образом C# (который исторически «более лучший» дельфи, скрещенный с плюсами и подглядывающий за джавой) менее удобен, чем дельфи?

Дык эта… ответ банален — а любой банальный вопрос должен быть обсосан Джоелом… Этот — не исключение.

> Расскажите подробнее, что именно вас смущает?

Результат?

PsyHaSTe 21 сентября 2018 в 02:36

0

> I just want to link everything I need in a single static EXE that runs without any installation prerequisites. I don't mind if it's a bit bigger.

dotnet publish + ilmerge, будет вам один бинарь, не вопрос. Только смысла всё еще не вижу.

> Результат?

Результат — работающее приложение. И по моему опыту оно тоже получится работающим.

khim 21 сентября 2018 в 03:03

+1

> dotnet publish + ilmerge, будет вам один бинарь, не вопрос.

Вот только проблемы с WPF, с .NET Core и прочим. Да — это работает, но это больше похоже на цирковой номер, чем на что-то продуманное.

> Только смысла всё еще не вижу.

Смысл в том, что заказав приложение… люди хотят получить **приложение**. Бинарник. .exe-файл. Нажми на кнопку — получишь результат. А не гору каких-то непонятных файлов, среди которых неясно — что и запускать-то.

> Результат — работающее приложение. И по моему опыту оно тоже получится работающим.

Если прочитать достаточное количество статей и прочего — да, наверное. Но в Delphi — оно такое получается «из коробки».

PsyHaSTe 21 сентября 2018 в 12:58

+2

> Вот только проблемы с WPF, с .NET Core и прочим.

Это и есть netcore. Проблем там нет.

> Да — это работает, но это больше похоже на цирковой номер, чем на что-то продуманное.

А это и есть цирковой номер. Примерно, как требовать игры не зависеть от C++ Redistrutable.

> Смысл в том, что заказав приложение… люди хотят получить приложение. Бинарник…exe-файл. Нажми на кнопку — получишь результат. А не гору каких-то непонятных файлов, среди которых неясно — что и запускать-то.

Не знаю, открою ли я секрет, что .net предустановлен во всех виндах начиная с XP, и вы точно так же на нем можете распространять "портабельные" приложения (хотя всё еще не вижу больших проблем).

Вы ведь в курсе, что GeForce Experience и Amd панелька это .net-приложения? И они умудряются как-то запускаться с одного install_driver.exe.

khim 21 сентября 2018 в 14:43

+2

> Не знаю, открою ли я секрет, что .net предустановлен во всех виндах начиная с XP

Откроете. Вы из какой-то альтернативной вселенной. Ибо в нашей XP выпустилась с Microsoft Java, а .NET в ней отсутствовал.

> Вы ведь в курсе, что GeForce Experience и Amd панелька это .net-приложения?

В курсе. Это сложно было не заметить: когда вместо быстрой и маленькой приблуды мы получили «прогрессивного монстра на сотню мегабайт» — об этом даже журналисты написали некоторые.



PsyHaSTe 21 сентября 2018 в 15:32

–1

> Откроете. Вы из какой-то альтернативной вселенной. Ибо в нашей XP выпустилась с Microsoft Java, а .NET в ней отсутствовал.

Странный у вас XP

- Windows XP Media Center Edition (Windows XP SP1) includes the .NET Framework 1.0 + SP2 as an OS component
- Windows XP Media Center Edition (Windows XP SP2 and higher) includes the .NET Framework 1.0 + SP3 as an OS component. On Windows XP Media Center Edition, the only way to get the .NET Framework 1.0 SP3 is to install Windows XP SP2 or higher. There is not a standalone 1.0 SP3 installer for this edition of Windows XP.
- Windows XP Tablet PC Edition (Windows XP SP1) includes the .NET Framework 1.0 + SP2 as an OS component
- Windows XP Tablet PC Edition (Windows XP SP2 and higher) includes the .NET Framework 1.0 + SP3 as an OS component. On Windows XP Tablet PC Edition, the only way to get the .NET Framework 1.0 SP3 is to install Windows XP SP2 or higher. There is not a standalone 1.0 SP3 installer for this edition of Windows XP.
- Windows Server 2003 (all x86 editions) includes the .NET Framework 1.1 as an OS component; 64-bit versions of Windows Server 2003 do not include a version of the .NET Framework as an OS component
- Windows Vista (all editions) includes the .NET Framework 2.0 and 3.0 as OS components 3.0 can be added or removed via the Programs and Fatures control panel.
- Windows Vista SP1 (all editions) includes the .NET Framework 2.0 SP1 and 3.0 SP1 as OS components. 3.0 SP1 can be added or removed via the Programs and Features control panel.
- Windows Server 2008 and Windows Server 2008 SP1 (all editions) includes the .NET Framework 2.0 SP1 and 3.0 SP1 as OS components. The .NET Framework 3.0 SP1 is not installed by default and must be added via the Programs and Features control panel though.
- Windows Server 2008 SP2 (all editions) includes the .NET Framework 2.0 SP2 and 3.0 SP2 as OS components. The .NET Framework 3.0 SP2 is not installed by default and must be added via the Programs and Features control panel though.
- Windows Server 2008 R2 (all editions) includes the .NET Framework 3.5.1 as an OS component. This means you will get the .NET Framework 2.0 SP2, 3.0 SP2 and 3.5 SP1 plus a few post 3.5 SP1 bug fixes. 3.0 SP2 and 3.5 SP1 can be added or removed via the Programs and Features control panel.
- Windows 7 (all editions) includes the .NET Framework 3.5.1 as an OS component. This means you will get the .NET Framework 2.0 SP2, 3.0 SP2 and 3.5 SP1 plus a few post 3.5 SP1 bug fixes. 3.0 SP2 and 3.5 SP1 can be added or removed via the Programs and Features control panel.
- Windows 8 (all editions) includes the .NET Framework 4.5 as an OS component, and it is installed by default. It also includes the .NET Framework 3.5 SP1 as an OS component that is not installed by default. The .NET Framework 3.5 SP1 can be added or removed via the Programs and Features control panel.
- Windows 8.1 (all editions) includes the .NET Framework 4.5.1 as an OS component, and it is installed by default. It also includes the .NET Framework 3.5 SP1 as an OS component that is not installed by default. The .NET Framework 3.5 SP1 can be added or removed via the Programs and Features control panel.
- Windows Server 2012 (all editions) includes the .NET Framework 4.5 as an OS component, and it is installed by default except in the Server Core configuration. It also includes the .NET Framework 3.5 SP1 as an OS component that is not installed by default. The .NET Framework 3.5 SP1 can be added or removed via the Server Manager.
- Windows Server 2012 R2 (all editions) includes the .NET Framework 4.5.1 as an OS component, and it is installed by default except in the Server Core configuration. It also includes the .NET Framework 3.5 SP1 as an OS component that is not installed by default. The .NET Framework 3.5 SP1 can be added or removed via the Server Manager.
- Windows 10 (all editions) includes the .NET Framework 4.6 as an OS component, and it is installed by default. It also includes the .NET Framework 3.5 SP1 as an OS component that is not installed by default. The .NET Framework 3.5 SP1 can be added or removed via the Programs and Features control panel.
- Windows 10 November 2015 Update (all editions) includes the .NET Framework 4.6.1 as an OS component, and it is installed by default. It also includes the .NET Framework 3.5 SP1 as an OS component that is not installed by default. The .NET Framework 3.5 SP1 can be added or removed via the Programs and Features control panel.
- Windows 10 Anniversary Update (all editions) includes the .NET Framework 4.6.2 as an OS component, and it is installed by default. It also includes the .NET Framework 3.5 SP1 as an OS component that is not installed by default. The .NET Framework 3.5 SP1 can be added or removed via the Programs and Features control panel.
- Windows Server 2016 (all editions) includes the .NET Framework 4.6.2 as an OS component, and it is installed by default except in the Server Core configuration. It also includes the .NET Framework 3.5 SP1 as an OS component that is not installed by default. The .NET Framework 3.5 SP1 can be added or removed via the Server Manager.
- Windows 10 Creators Update (all editions) includes the .NET Framework 4.7 as an OS component, and it is installed by default. It also includes the .NET Framework 3.5 SP1 as an OS component that is not installed by default. The .NET Framework 3.5 SP1 can be added or removed via the Programs and Features control panel.
- Windows 10 Fall 2017 Creators Update (all editions) includes the .NET Framework 4.7.1 as an OS component, and it is installed by default. It also includes the .NET Framework 3.5 SP1 as an OS component that is not installed by default. The .NET Framework 3.5 SP1 can be added or removed via the Programs and Features control panel.
- Windows 10 April 2018 Update (all editions) includes the .NET Framework 4.7.2 as an OS component, and it is installed by default. It also includes the .NET Framework 3.5 SP1 as an OS component that is not installed by default. The .NET Framework 3.5 SP1 can be added or removed via the Programs and Features control panel.

Хотя один фиг, XP я последний раз видел лет 10 назад, когда поставил себе семерку. Что уж говорить про 2018 год.

> В курсе. Это сложно было не заметить: когда вместо быстрой и маленькой приблуды мы получили «прогрессивного монстра на сотню мегабайт» — об этом даже журналисты написали некоторые.

А я вот не заметил, пока в декомпилятор ради интереса не заглянул.

khim 21 сентября 2018 в 22:05

+2

> Странный у вас XP

Это у вас он странный. Непонятно зачем приводить огромную портянку, если можно пойти в Wikipedia и найти даты:
Windows XP: August 24, 2001
.NET Framework 1.0: February 13, 2002

Было бы странно и удивительно если бы операционка, выпущенная в 2001м году содержала бы компонент, выпущенный в 2002м? Они и не содержала…

А уж сколько лет после этого нельзя было ни рассчитывать на то, что .NET у клиента установлен, ни на то, что он «легко» может скачать 20MB (в те времена ещё флоппи в ходу были, 20MB было большим объёмом).



PsyHaSTe 21 сентября 2018 в 22:12

–2

Разговор был не про август 2001, а про то, что в 2018 году простое десктоп-приложение на дельфи написать проще, чем на шарпе.

khim 23 сентября 2018 в 16:47

0

Конечно приложения на том языке, которому разработчик операционки дает специальные привилегии писать проще. Будь то C# для Windows или Java для Android.

Хотите более-менее объективного сравнения — сравните C# и Delphi приложения под Android или iOS. Есть ощущение, что на Delphi попроще будет.

> Но на тему того, что и кто обсуждал — то фраза Не знаю, открою ли я секрет, что .net предустановлен во всех виндах начиная с XP, и вы точно так же на нем можете распространять «портабельные» приложения была вами произнесена, вас никто за язык не тянул. Дальше, если углубиться в версии .NET, то окажется, что собрать что-нибудь для .NET 1.x современными тулами — очень непросто. а .NET 2.0 — это уже 2006й год, XP тут и не пахнет… что, кстати, очень хорошо объясняет почему поддержка XP для многих разработчиков — большая проблема.



PsyHaSTe 23 сентября 2018 в 16:53

0

> Хотите более-менее объективного сравнения — сравните C# и Delphi приложения под Android или iOS. Есть ощущение, что на Delphi попроще будет.

Да пожалуйста, Xamarin получше будет, как мне кажется. Хотя на дельфи под мобилки я не писал, но прекрасно помню, как с ней возился.

> > Но на тему того, что и кто обсуждал — то фраза Не знаю, открою ли я секрет, что .net предустановлен во всех виндах начиная с XP, и вы точно так же на нем можете распространять «портабельные» приложения была вами произнесена, вас никто за язык не тянул. Дальше, если углубиться в версии .NET, то окажется, что собрать что-нибудь для .NET 1.x современными тулами — очень непросто. а .NET 2.0 — это уже 2006й год, XP тут и не пахнет… что, кстати, очень хорошо объясняет почему поддержка XP для многих разработчиков — большая проблема.

Она была произнесена в контексте 2018 года. Если кто-то вдруг по каким-топ причинам в 2018 году использует XP, то это SP3 со всем фаршем, и именно фреймворками, драйверами и т.п. Сравнивать с XP момента выхода смысла нет, если сейчас они вдруг где-то и используются, то их существование пренебрежимо мало.



0xf0a00 21 сентября 2018 в 10:44

0

> который исторически «более лучший» дельфи

> Удобные новые пакеты, linq, IDE явно удобнее RAD Studio

habr.com/post/423889/#comment_19135891
Это вот этот вот монстр от которого комп можно спасти только полной переустановкой винды лучше радстудио?
Visual Studio 2005 vs Delphi 7 еще может быть, но современные если сравнивать то визуал студио задыхается под собственным весом.



PsyHaSTe 21 сентября 2018 в 13:02

0

> Это вот этот вот монстр от которого комп можно спасти только полной переустановкой винды лучше радстудио?

Минимальная установка VS2017 — 300мегабайт.

Про открытие многомегабайтных файлов отлично рассказывает этот комментарий: https://habr.com/post/423889/#comment_19140333.

То есть попросили в 32-битный процесс загрузить 3гб данных и удивляемся, а че оно так не быстро работает… Ну ок.

Если вы постоянно вместо того, чтобы работать с кодом открываете и закрываете огромные XML-файлы, то может RAD studio и лучше.



0xf0a00 21 сентября 2018 в 15:35

0

> Минимальная установка VS2017 — 300мегабайт.

И какой у него функционал? Подозреваю что околонулевой.

> То есть попросили в 32-битный процесс загрузить 3гб данных и удивляемся, а че оно так не быстро работает… Ну ок.

Если читать комментарии не выборочно, то там же написано что фар не остановило бы ни 32 бита архитектуры, ни ограничение по памяти.

> Если вы постоянно вместо того, чтобы работать с кодом открываете и закрываете огромные XML-файлы, то может RAD studio и лучше.

Я ей постоянно пользуюсь, она отзывчивая и без лишнего хлама. Этого мне достаточно что бы считать ее лучшей ide для разработки на десктоп.



PsyHaSTe 21 сентября 2018 в 15:38

0

> И какой у него функционал? Подозреваю что околонулевой.

Понятия не имею, у меня нет проблем поставить 10гигабайтную IDE со всем, что мне нужно (и сверху какой-нибудь решарпер впилить), потому что моё время дороже, а места на винте — завались. Я же сам конечный пользователь, и мне себя более жалко, чем жесткий диск, который и так наполовину пустой.

> Я ей постоянно пользуюсь, она отзывчивая и без лишнего хлама. Этого мне достаточно что бы считать ее лучшей ide для разработки на десктоп.

Где-то я это уже видел...

> Когда наш гипотетический Блаб-программист смотрит вниз на континуум мощности языков, он знает, что смотрит вниз. Менее мощные, чем Блаб, языки явно менее мощны, так как в них нет некой особенности, к которой привык программист. Но когда он смотрит в другом направлении, вверх, он не осознает, что смотрит вверх. То, что он видит, — это просто "странные" языки. Возможно, он считает их одинаковыми с Блабом по мощности, но со всевозможными сложными штучками. Блаба для нашего программиста вполне достаточно, так как он думает на Блабе.



0xf0a00 21 сентября 2018 в 15:54

0

> Понятия не имею

То есть вы не компетентны, а в спор лезете.

> у меня нет проблем поставить 10 гигабайтную IDE со всем, что мне нужно

Рад за ваш жесткий диск, только то что вы можете ее поставить проблем ее перегруженности и тормозунтости не решает от слова совсем.

> Где-то я это уже видел...

Причем язык к ide мне не совсем понятно. Опять таки поднимается вопрос о вашей квалификации и достаточно ли ее. Тем более в сравнении языков(которые в принципе мало сравнимы). Видимо вы где слышали краем уха что модно засирать Delphi, но разобраться что к чему у вас не хватило то ли усидчивости то ли айкью.

PsyHaSTe 21 сентября 2018 в 15:57

–1

> То есть вы не компетентны, а в спор лезете.

Когда я работал с дельфи, я про рефакторинги и прочее в жизни не слыхивал. На уровне Delphi7 студия в такой конфигурации явно может работать.

> Рад за ваш жесткий диск, только то что вы можете ее поставить проблем ее перегруженности и тормозунтости не решает от слова совсем.

тогда к чему вы привели размер приложения?

> Причем язык к ide мне не совсем понятно.

К тому, что если вы не понимаете, к чему часть функционала, это не значит, что это «хлам».

> Опять таки поднимается вопрос о вашей квалификации и достаточно ли ее.

Ad hominem, отлично

> Тем более в сравнении языков(которые в принципе мало сравнимы).

сравнивали на задаче «сделать простое простое приложение». Я так думаю, любой адекватный язык можно сравнить на этой задаче.

> Видимо вы где слышали краем уха что модно засирать Delphi, но разобраться что к чему у вас не хватило то ли усидчивости то ли айкью.

Просто тупой вброс

novice2001 21 сентября 2018 в 09:39

0

Word не предназначен для вёрстки, для неё существуют совсем другие приложения.
И в целом, расскажите, какие существеннейшие изменения произошли в Word между 2003 и 2013/2016 версиями, что он стал намного тяжелее и медленнее?

Am0ralist 21 сентября 2018 в 12:26

0

Ну так-то и 97 ворд был в чем-то лучше 2003, он, как минимум, у меня не крашился постоянно на файлах с количеством страниц от 500 до тысячи…
Ну т.е. 2003 мог в рандомный момент упасть просто так даже при чтении но минимум раз в час.
А в 97 — его спокойно можно было редактировать активно и все было норм… Эх)

SirEdvin 21 сентября 2018 в 13:12

+1

Набор документа с правильными отсупами, форматированием, заголовками, автоматической нумерацией и прочими прелестями — это верстка.

Если вы не это все не используете, я за вас рад, но Word в таком случае вам просто не нужен, возьмите тот же Sublime.

novice2001 21 сентября 2018 в 13:35

+1

Действительно, идиоты зачем-то пользуются Adobe InDesign (и даже, страшно сказать, покупают его), когда, оказывается Word — это и есть программа для вёрстки… Да, еще какой-то Donald Knuth придумал какой-то TEX, вот ведь смешной старик, наверное просто не знает про Word… Вы его просветите там, а то он уже 40 лет на него потратил.



SirEdvin 21 сентября 2018 в 13:45

0

То, что функционал Word не хватает на что-то серьезное и то, что он проигрывает TEX практически во всем — это другие проблемы.

Функционал то в нем есть, и он обновляется, а то, что он не нужен вам и скорее всего, значительной куче людей — это другая проблема, правда.

novice2001 21 сентября 2018 в 13:53

0

Это вообще не проблемы Word, поскольку он никогда и не позиционировался, как система верстки. Это текстовый процессор.
А что там появилось в Word 2013/2016 вы так и не ответили. Именно потому, что ничего принципиально нового и полезного в т.ч. лично для вас там не появилось, ибо иначе вы легко и непринужденно ответили бы на вопрос вместо описания несуществующих проблем.



Grox 21 сентября 2018 в 21:27

0

В Word 2010 появилась возможность форматировать и отпечатать документ в виде книги. Ранее это делалось с плагинами. В одной из компаний это был частый случай использования.

khim 21 сентября 2018 в 22:09

0

Принято. А теперь вопрос: сколько занимал тот плагин и сколько занимает MS Office 2016? В памяти и на диске? И почему нельзя было просто использовать плагин, вместо того, чтобы превращать офис в монстра?



Grox 21 сентября 2018 в 22:55

0

Так это один из примеров доработанного функционала. Сколько их ещё? Я думаю, что вы можете найти ответы в списках изменений по версиям.

khim 21 сентября 2018 в 23:08

0

Могу. Но все они не дадут ответа на вопрос: зачем. Зачем нужна вся эта архитектурная астронавтика, если все полезные фичи можно сделать аддонами для MS Office 2000 (а то и для MS Office 97)?

Проблема в том, что в развитии офиса (как и почти любого современного софта) произошли две вещи:
1. Добавление кучки мелких фишек, которые реально облегчают людям жизнь
2. Добавление 100500 уровней абстракции, которые только лишь в приводят к потерям ресурсов и замедляют работу

При этом такое ощущение, что всё это сделано для того, чтобы кто-то получил promotion. Ну потому, что за добавление «печати документа в виде книг» вы promotion не получите, а за добавление нового слоя абстракций, на разработку которого ушло 10 человеко-лет и который ни одной проблемы не решил, но зато породил два десятка красивых презентаций — легко.

Всё тот же самый «секрет айсберга» — но уже по отношению к руководителям компаний…



Sychuan 21 сентября 2018 в 14:02

0

> Adobe InDesign (и даже, страшно сказать, покупают его), когда, оказывается Word — это и есть программа для вёрстки… Да, еще какой-то Donald Knuth придумал какой-то TEX, вот ведь смешной старик, наверное просто не знает про Word

Ворд применяется для верстки офисных документов., которые могут быть довольно сложными. Короче все что можно печатать на принтере. ИнДизайн применяется для верстки полиграфической продукции. И отчеты с приказами им никто не верстает.

novice2001 21 сентября 2018 в 14:29

0

Не нужно называть версткой форматирование, вот и всё.

mkshma 21 сентября 2018 в 15:55

+1

Форматирование является подмножеством верстки.



SirEdvin 21 сентября 2018 в 16:45

0

Сейчас бы называть создание автономной формы или диплома форматированием.

Вы из тех, кто нумирует формулы в ворде вручную?



0xd34df00d 22 сентября 2018 в 00:02

0

А чего там диплом создавать, \usepackage нужный и всё.

Посмотрел бы я, кстати, на версионирование диплома в ворде по мере его написания и общения с научруком.



SirEdvin 22 сентября 2018 в 11:36

0

Ахахаха, очень смешно.

На самом деле нужно подключить XeLaTeX, потом еще довольно много подбирать нужные параметры, поменять нумерацию, написать свой формат литературы, потому что в интернете можно найти только российский и то устаревший и все равно надо вставлять костыли и склеивать section и subsection, если между ними ничего нет, так как сделать это из tex очень сложно.

Но это все равно лучше, чем в ворде, да.

> Посмотрел бы я, кстати, на версионирование диплома в ворде по мере его написания и общения с научруком.

Но вообще-то, ворд умеет.



0xd34df00d 22 сентября 2018 в 17:13

0

Зачем так сложно? УМВР, при этом половина пакетов скопирована из статей, которые я перед этим по теме диплома публиковал.

А sty я у одногруппников взял.

За разбитие по главам вам лойс, кстати.



SirEdvin 23 сентября 2018 в 12:38

0

Ну, очевидно, потому что у вас так себе подходят к проверке на соответствование госту.

У вас тут как минимум нет:

1. Шрифта Times New Roman
2. Размер шрифта не 14
3. Не вижу директиву для полуторного интервала, может пропустил
4. Ну и у вас катастрофически неправильно оформлена библиография.

Вполне возможно, что я чего-то не знаю, так как вижу в другой стране, но у нас все эти пункты строго проверяли, поэтому надо было фиксить :)



0xd34df00d 24 сентября 2018 в 00:09

0

А, ну я не уверен, что у нас вообще сам текст читали, это да.

Вот с библиографией для публикации в разных журналах регулярно мучиться приходилось :(

swelf 21 сентября 2018 в 14:41

0

То что он стал лучше, наверно, но соразмерны ли эти улучшения, увеличению размера/расходу памяти/процессора? Чтобы починить правильные отступы надо увеличить размер в 10 раз?



PsyHaSTe 21 сентября 2018 в 15:40

–1

Да, нужно.

swelf 21 сентября 2018 в 19:19

0

Почему нельзя внести эти изменения увеличив потребление ресурсов на 5%, что конкретно заставило офис вырасти на порядок? Количество функций увеличилось на порядок?



PsyHaSTe 21 сентября 2018 в 19:29

0

Потому что правило 80/20. Для добавления 20% нехватающего функционала нужно в четыре раза больше усилий и кода.

swelf 22 сентября 2018 в 00:39

0

ну знание правила парето не дает основания пихать его как аргумент куда угодно бездумно.
Так что вырос объем от чего угодно, но не от новых функций и фич.



PsyHaSTe 22 сентября 2018 в 03:18

+1

Не нужно быть ученым, чтобы понимать, что полноценный продукт содержит на порядок больше кода, чем прототип, который визуально делает то же самое. Это как истории, когда человек показывает JS-прототип без бэкенда, а клиенты уже думают, что на 90% реализовано. А потом не понимают, почему релиза все нет и нет…

Ну например, новая версия может содержать интернационализацию на много языков. Тут возникает проблема с тем, что в кнопки не влезают тексты. Окей, запиливаем модуль масштабирования текста, причем так, чтобы с ним работали даже программы, которые про него не знают. Фух, написали кучу сложного кода, но работает, из коробки и правильно! Отлично. Ан нет, пришли из отдела эргономики, говорят, что у арабов от меюшки слева голова кругом идет, нужно чтобы было справа. «Окей» говорим мы, и начинаем воротить систему автоматического зеркалирования интерфейса (но не всего, а то не дай бог отзеркалировать что-нибудь не то, что надо), опять начинаем тестировать, находим кучу несовместимостей с существующими приложениями вроде игр… Продолжать можно до бесконечности, но именно все эти «свистоперделки» — это то, что хотят пользователи — чтобы из коробки всё работало, как надо, и не важно, сколько лишнего места на винте оно займет.

khim 22 сентября 2018 в 03:41

0

> Ну например, новая версия может содержать интернационализацию на много языков.

Зевает. Да-да, MS Office 2000 так умел: много языков, можно одновременно целую кучу поставить на машину, пакет с десятком языков в Enterprise версии. Про Office 95/97 не скажу, врать не буду, но MS Office 2000 всё вами перечисленное умел.

> Окей, запиливаем модуль масштабирования текста, причем так, чтобы с ним работали даже программы, которые про него не знают.

Ну то есть вместо работающей реализации мы досыпаем говнохода, а потом ещё обматываем говнокодом и присыпаем сверху ещё порцией говнокода.

Всё понятно, кроме одного: что мешало просто взять — и сделать так, как это было задумано (и работало!) 20 лет назад?

> Фух, написали кучу сложного кода, но работает, из коробки и правильно!

Нет. Работает неправильно. Глючит. Кнопки разъезжаются и нам приходится костылять новые костыли поверх старых. И потом обматывать это скотчем и эвристиками. Потому что если разработчик об этом не задумается во время проектирования программы, то сколько костылей не пристраивай — а всё равно что-нибудь, где-нибудь да отвалится.

> Ан нет, пришли из отдела эргономики, говорят, что у арабов от меюшки слева голова кругом идет, нужно чтобы было справа.

И это тот же MS Office 2000 умел, да.

> Продолжать можно до бесконечности, но именно все эти «свистоперделки» это то, что хотят пользователи — чтобы из коробки всё работало, как надо, и не важно, сколько лишнего места на винте оно займет.

Пользователи хотят, чтоб оно работало, а не глючило, на самом-то деле. А все эти потуги сделать так, чтобы работало без того, чтобы на это внимание разработчики программ обращали — ничем хорошим не кончаются. Результат всё равно напоминает руссификацию DOS'овских квестов со всеми этим «Окрт» (вместо «Open»).

Если же не ставить задачу как: «у нас есть куча дерьма, но мы хотим сделать вид, что это такая конфетка», то… всё это не нужно.

Количество людей, работавших как с классической линейкой (MS Office 95-2003), так и с новомодными извращениями (2007-2016) и при этом не оплакивающих «почивший в бозе» «старый мир» — ничтожно мало. Большинство «фанатов» нового интерфейса — это люди, которые старого в глаза не видели… в чистом виде стокгольмский синдром.



PsyHaSTe 23 сентября 2018 в 03:50

0

Одно изменение, которое я точно помню произошло после 2003 — появление риббона. Супер-удобная штука, которая является объективным шагом вперед. Но у вас видимо клин сошелся на 2003.

> Пользователи хотят, чтоб оно работало, а не глючило, на самом-то деле. А все эти потуги сделать так, чтобы работало без того, чтобы на это внимание разработчики программ обращали — ничем хорошим не кончаются. Результат всё равно напоминает руссификацию DOS'овских квестов со всеми этим «Окрт» (вместо «Open»).

У меня ни один ворд никогда не глючил и работал, что 2003, то 2007, что 2010, что 2013, что 2016. Может, вы не правильно его готовите?

> Большинство «фанатов» нового интерфейса — это люди, которые старого в глаза не видели… в чистом виде стокгольмский синдром.

Хех, согласен. только это скорее синдром утенка, что люди не захотели учиться ничему новому, а первый увиденный интерфейс остается самым удобным навсегда. Тот же риббон сюда же «староверы», которые в ворде почти не работают любят вздыхать «ах вы, куда вы дели мои меюшечки. Клятый майкрософт».

khim 23 сентября 2018 в 17:27

0

> Одно изменение, которое я точно помню произошло после 2003 — появление риббона.

Угу. На разработку 4 года ушло, да. И в результате получили нечто, что работает заметно медленнее, требует куда больше ресурсов (в давнном случае места на экране) и не даёт возможности сделать ничего нового. Именно то, о чём статья.

> Но у вас видимо клин сошелся на 2003.

Скорее 2000й. 2003й — это первый, робкий, шаг в сторону подхода «а давайте добавим свистоперделок, которые жрут ресурсы и не дают никаких новых возможностей». Но да, до 2007го и последующих ему далеко.

> У меня ни один ворд никогда не глючил и работал, что 2003, то 2007, что 2010, что 2013, что 2016. Может, вы не правильно его готовите?

Дык они с самого начала более-менее грамотно написаны. Тот факт, что после добавления бессмысленных перделок потребление ресурсов возрасло, но программа не стала глючить — это неплохо, в общем и целом… но вроде бы цель была не в этом, а в том, чтобы заставить правильно работать плохо написанные программы? Этого — не получилось.

> Хех, согласен. только это скорее синдром утенка, что люди не захотели учиться ничему новому, а первый увиденный интерфейс остается самым удобным навсегда.

Извините, но нет. Переходу с Lotus 1-2-3 на Excel этот синдром не помешал, появление тулбара с кнопками в Excel 3.x (да-да, представьте себе, тубары тоже не сразу появились) было принято «на ура». Табы в Excel 5.0 тоже отторжения ни у кого не возвали. И даже разного рода «игры с цветами» с MS Office 97 до MS Office 2003 не особо раздражали.

А вот ribbon — это да, MS Office 2007 насаждали «огнём и мечом»… Если бы MS Office 2003 продавали и поддерживали бы сейчас — он вполне бы пользовался спросом…

> Тот же риббон сюда же «староверы», которые в ворде почти не работают любят вздыхать «ах вы, куда вы дели мои меюшечки. Клятый майкрософт».

Тут вы правы. Я в ворде мало работаю (так как вообще не понимаю зачем и кому он в сегодняшних условиях нужен). А вот LibreCalc использую частенько. Перешёл с Excel когда тот превратился в

тормозилово… Забавно, что до MS Office 2007, наоборот, исторически, OpenOffice жрал больше ресурсов и тормозил… а сегодня — вот так…

PsyHaSTe 23 сентября 2018 в 21:52

+1

> Угу. На разработку 4 года ушло, да. И в результате получили нечто, что работает заметно медленнее, требует куда больше ресурсов (в давнюю случае места на экране) и не даёт возможности сделать ничего нового. Именно то, о чём статья.

Как вы умудряетесь замечать лаги ворда? Как я печатал, так и появлялся символ за 16.6 мс на экране. Я не понимаю, что работает «заметно медленнее». Вот когда я сейчас набираю сообщение, я вижу как запаздывает на 300-400 мс появление набираемого текста, вот уж действительно лагает так лагает. Ворд себе никогда такого не позволял, ни на каком железе.

> А вот ribbon — это да, MS Office 2007 насаждали «огнём и мечом»… Если бы MS Office 2003 продавали и поддерживали бы сейчас — он вполне бы пользовался спросом…

Есть вполне объективные критерии оценки UX, в частности «кликов для выполнения задачи», «частота этой задачи» и так далее. У риббона эти показатели выше, потому что частоиспользуемые операции положены ближе, и за счет общего устройства.

То, что мне кажется и вам кажется — это одно. А я читал анализ UX-дизайнера всех ключевых особенностей. И статистика не на вашей стороне — время совершения частоиспользуемых функций с риббоном снижается вполне объективно.

khim 24 сентября 2018 в 02:22

0

> Как вы умудряетесь замечать лаги ворда? Как я печатал, так и появлялся символ за 16.6 мс на экране.

Очень просто: берёте «печатную машинку» на Atom'е с 2GB памяти и весь этот монстр успешно уходит в своп. А если своп идёт на HDD… совсем плохо будет.

> Ворд себе никогда такого не позволял, ни на каком железе.

Я тоже так думал, пока не увидел его на «компьютерах для бухгалтерии», где в ТЗ было «общая цена — $300 за рабочее место, включая монитор и либо мини-корпус, либо моноблок».

> И статистика не на вашей стороне — время совершения частоиспользуемых функций с риббоном снижается вполне объективно.

Вот только почему-то практика эту теорию не подтверждает. Я видел исследования в крупной компании, где в 2007м году исследовали этот вопрос — без измерения параметров типа «кликов для выполнения задачи», но с замером времени на выполнение типовых задач средним работником.

Сразу после пересадки с MS Office 2003 на MS Office 2007 эффективность работы падала катастрофически, но через две недели скорость возвращалась — однако не до конца, всё равно задачи решались медленнее, чем в MS Office 2003. При переходе на OpenOffice.org катастрофического падения не было, но и возврата назад не было тоже.

Более внимательное наблюдение показало, что происходило это из-за отличий во внутренней логике, которую переход на MS Office 2007 не менял, а переход на OpenOffice.org — менял. Простейший пример: задача добавить в документ нумерацию страниц. В MS Office (любой версии) как «Вставить->Номер страницы», после чего Word спрашивает «а куда, собственно: вниз, вверх, нужно ли на первой странице?». А в MS Office — вставляется поле с номером… и всё. То, что там нужно сначала добавить колонтитул, а уж потом, в него — вставить соотвествующее поле многие не могли вспомнить и через две недели.

В результате перешли-таки на MS Office 2007 — но только и исключительно потому, что лицензий на MS Office 2003 больше купить было нельзя.

PsyHaSTe 24 сентября 2018 в 10:49

0

> Очень просто: берёте «печатную машинку» на Atom'е с 2GB памяти и весь этот монстр успешно уходит в своп. А если своп идёт на HDD… совсем плохо будет.

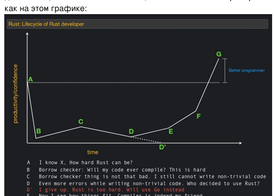
Со свопом что угодно плохо работать будет.
Я буквально пару недель назад купил «печатную машинку» за 20к — очень бюджетно для ноута с IPS монитором скажу вам. И ворд там не лагает.

> Я тоже так думал, пока не увидел его на «компьютерах для бухгалтерии», где в ТЗ было «общая цена — $300 за рабочее место, включая монитор и либо мини-корпус, либо моноблок».

300$ это около 18 тысяч, близко к озвученой цене.

> Вот только почему-то практика эту теорию не подтверждает. Я видел исследования в крупной компании, где в 2007м году исследовали этот вопрос — без измерения параметров типа «кликов для выполнения задачи», но с замером времени на выполнение типовых задач средним работником.
>
> Сразу после пересадки с MS Office 2003 на MS Office 2007 эффективность работы падала катастрофически, но через две недели скорость возвращалась — однако не до конца, всё равно задачи решались медленнее, чем в MS Office 2003. При переходе на OpenOffice.org катастрофического падения не было, но и возврата назад не было тоже.

Видимо, надо было провести замер еще через месяц и год, нет? 2 недели — достаточно, чтобы понять основную логику, но не достаточно, чтобы начать её полноценно использовать. Примерно как на этом графике:



В вашем случае замеры производились в точках A, E, F.

shurricken 23 сентября 2018 в 17:28

0

ну да только риббон)) почем у он всем должен быть удобен??? да еще и в разных версиях в разные места там функции убегали…
вот радости-то помнить для всех версий что и где…
как и предыдущий пост давно пользуюсь «экселем» от опенофиса, а поддерживался бы старый оригинальный «эксель» пользовался бы им по-прежнему…

PsyHaSTe 23 сентября 2018 в 21:52

–1

Выше ответил. Потому что снижается количество кликов для выполнения частовыполняемой операции X. В итоге интерфейс не такой — да, надо переучиваться -да, но он не плохой — нет.

Am0ralist 24 сентября 2018 в 12:13

0

Да, для часто. А вот для редко — усложняется сильно, надо под себя переделывать, плюс пальцем получается даже хуже, чем в классическом меню.

PsyHaSTe 24 сентября 2018 в 12:21

0

Переделать под себя надо, да, но кстати эти настройки можно сохранить вроде и с собой везде таскать.

Про палец я не понял. Если речь про планшеты, то я вообще не вижу, как на них можно подобными вещами заниматься хоть где. Это устройство чтобы покемонов ловить да в интернете серфить.

VolCh 24 сентября 2018 в 17:10

0

А время поиска мышью места куда кликнуть учитывается? Навигация мышью по обычному текстовому меню куда быстрее чем наведение мыши на иконки «риббон» или как его и чтение появившегося тултипа. И даже когда тултипы выедены меню получается двумерное, причём не с регулярной структурой.

PsyHaSTe 24 сентября 2018 в 17:38

0

Скорее всего, был анализ групп «я ничего не знаю про офис», «я знаю базово», «средне», «продвинуто» и «полностью».

Было ли сравнение «человек эксперт в старом офисе, но ничего не знает про риббон» я не знаю, а если и было, скорее всего его

результаты для внутреннего использования в стиле «насколько сильно пользователи будут нас ненавидеть на запуске».



Alexeyslav 24 сентября 2018 в 14:38

0

Я тоже с радостью пользовался бы, но… многий корпоративный софт всё ещё заточен под микрософт офис, там где заявлена поддержка опенофиса — она номинальная и через костыли. Кроме того, по какой-то причине редактор макросов в опенофисе, либреофисе лагает с подсветкой синтаксиса к тому же неотключаемой. На один экран кода лаг незаметен, но когда кода уже два экрана лаг заметен и вызывает полную перерисовку подсветки КАЖДЫЙ введённый с клавиатуры символ. на 300 строках работать уже невозможно — на каждое нажатие кнопки происходит 2-3 секундная перерисовка. Почитал форумы… оказалось так не повезло лишь пользователям с AMD-шными процессорами. Спустя год проблема с перерисовкой не решена, этот баг видимо не в приоритете.

khim 26 сентября 2018 в 22:39

0

Если мне нужно писать что-то сложное, то я просто пишу это на Python. Не самый лучший в мире язык, да, но для него есть куча инструментов и он, по крайней мере для меня, резко удобнее встроенных языков как LibreOffice, так и MS Office.



Alexeyslav 26 сентября 2018 в 23:20

0

Так питон и есть встроенный язык в опенофис. И этим встроенным языкам нет альтернативы, если нужна работа с документом.

ne555 20 сентября 2018 в 20:42

0

Крайне не согласен, обслуживаю в промышленности промышленники на dos/98 и на winxp. Скады на xp валются, для них куча заплаток (прям как в статье написано: скрипты на перезапуск), 98 работает, как ни в чем нк бывало.
От датчика идет провод, все работает, но там, где вместо провода радиоканал-это просто дерьмо, но все вынуждены мириться даже сверху. Могу написать целую статью в поддержку автора и не понаслышке.



SirEdvin 20 сентября 2018 в 21:02

+1

Это круто, а вы разобрались в причинах или просто «ну, на 98 работает, а на winxp — нет, значит новомодные программисты виноваты»?

ne555 21 сентября 2018 в 07:21

0

На win xp sp3 разбирались многие, даже представители скады, в итоге (не разобрались) только заплатка, которая через ~пол года дает сбои.



serbod 21 сентября 2018 в 10:34

0

Просто в 98 мало проверок, прога может лазить в чужую память, пользоваться закрытыми хендлами и дескрипторами, и ей за это ничего не будет. Вот только от этого может рухнуть вся ось целиком (с повреждением файловой системы!), а не отдельное приложение.

ne555 21 сентября 2018 в 16:36

0

За 15 лет ничего такого о чем Вы пишите, потому что это промышленник (арм), там только и должен работать этот софт от лета до следующего лета.

sys_int64 20 сентября 2018 в 21:11

0

Не знаю про 90е, но вот я помню как работал в 3D редакторе — Maya 7 с 1 ГБ оперативки и все летало, на больших сценах у меня начинала тормозить графика, в итоге докуил еще 1 ГБ оперативки и программа стала просто нереально быстрой.

sys_int64 20 сентября 2018 в 21:13

+1

а вот emacs'ом я и до сих пор пользуюсь, второй после IDEA инструмент для написания кода.



Crandel 20 сентября 2018 в 22:30

+1

> Сравниваете современные редакторы кода и древний emacs?

Я на "древнем" емаксе за 2 минуты делаю то, что любители всяких студий от некрософта пыхтят и мучаются мышкой по полчаса. Используя тайловый менеджер без единой анимации, я на порядок быстрее работаю за своих коллег с их гномами или кедами, потому что пока они мышкой наведут на окно, пока анимация, пока окно откроется, я уже сделал все необходимое и вернулся обратно в редактор. Все эти рюшечки только отвлекают от задачи и сильно бесят. Я сейчас работаю в режиме, как только мысль появилась, она сразу автоматом воплотилась в код.



SirEdvin 20 сентября 2018 в 23:31

+3

> Я на "древнем" емаксе за 2 минуты делаю то, что любители всяких студий от некрософта пыхтят и мучаются мышкой по полчаса.

Если сравнивать тот же emacs с продуктами от JetBrains он проигрывает практически во всем, в чем сильна ide.
Если же сравнивать emacs с чем-то не ide, например, sublime — все работает практически так же быстро, если не наставить кривых плагинов.

> Используя тайловый менеджер без единой анимации, я на порядок быстрее работаю за своих коллег с их гномами или кедами, потому что пока они мышкой наведут на окно, пока анимация, пока окно откроется, я уже сделал все необходимое и вернулся обратно в редактор. Все эти рюшечки только отвлекают от задачи и сильно бесят

О класс, и я так делаю. Я использую awesome WM и давайте я перечислю список того, что работает плохо:

1. Lock screen по факту нормальный сделать нельзя или ад с xscreensaver или костыли с light-locker
2. Виджеты — этой вещи практически не существует. Календарь, монтирование дисков, панель доступа в выключению — всего этого нет. Все нужно писать самому на lua или доставать из интернетов, а оно оказывается потом не работает или работает плохо, потому что в одном дистре поменяли формат работы с батарей, во втором поломали логику монтирования так далее. Реально хорошо работают только те программы, которые не привязываются к DE напрямую.
3. Низкоуровневые баги, которые DE фиксят за вас. Например, из-за того, что я переключаю клавиатуру через xkbswitch оно все переодически отваливается и больше не переключает. Печалька.

А так да, довольно классный подход, но с теми же недостатками, что описали выше — авторы DE уже просто поели проблем, а вам это только предстоит и вы скорее всего просто заточите ваш тайловый менеджер под конкретную версию дистра и при обновлении будете страдать.

Crandel 20 сентября 2018 в 23:47

+2

> Я использую awesome WM и давайте я перечислю список того, что работает плохо:

Я использую i3 и уже 2-3 года почти не меняю конфиг, последнее изменение — i3status-rust. Только один скрипт использую, который таймером раз в час запускается и проверяет, если нету процесса i3lock — запускает новый. Делаю на работе перерывы на 5 минут. xxkb у меня показывает текущую раскладку клавиатуры. Украинский и английский настроены в иксах + русские буквы переключаются через правый альт а немецкии умлауты добавляются с помощью композ, назначенной на принтскрин. Проблема за это время была только одна — трей, некоторые приложения не хотят появляться.

> Если сравнивать тот же emacs с продуктами от JetBrains он проигрывает практически во всем, в чем сильна ide.

Я пользуюсь Pycharm только когда надо рефакторить очень много связанных сущностей. Все остальное время он меня бесит своими тормозами и никаким редактором. 95 % моей работы емакс делает очень быстро и качественно. У меня реализуются автокомплит, goto definition, подсветка синтаксиса, реалтайм линтер, автоформат pep8 при сохранении файла и другие полезные в работе функции. Зато magit предоставляет удобнейший плагин для работы с гитом, намного лучше того, что может дать пайчарм или любой другой клиент.

Neikist 20 сентября 2018 в 23:49

0

Вот почему в полноценных DE подобное не запилят? И под виндой? Пробовал тайловые менеджеры, все очень приятно и удобно но не готов я сотни часов своего времени на их настройку и написание своих плагинов тратить… Тем более когда и так разного рода глюки вроде нерабочего японского ввода в части программ.

НЛО прилетело и опубликовало эту надпись здесь

Neikist 21 сентября 2018 в 00:03

0

Тем что есть я пользуюсь, вот только это не дотягивает до тех же awesome и i3 на порядок. И не нужно мне так как хочу именно я. Я просто не понимаю (не возмущаюсь, а именно не понимаю), почему я вынужден либо выбирать комфорт и красоту кед (или винды), или удобное управление окнами но полностью убогое окружение в остальном.

НЛО прилетело и опубликовало эту надпись здесь

sys_int64 21 сентября 2018 в 06:09

+2

> Если же сравнивать emacs с чем-то не ide, например, sublime — все работает практически так же быстро, если не наставить кривых плагинов.

Вот не согласен, саблайму до емакса очень далеко, из емакса вполне себе можно соорудить полноценную IDE, чаще всего емакс может все что может и IDEA от JetBrains, согласен, emacs по функционалу не дотягивает до JetBrains продукции, но не дотягивает он совсем чуть-чуть, для меня emacs — это второй выбор после IDEA, если нет нормально поддержи для языка в IDEA, то я лучше выбиру emacs, для питона есть + всякие приятные плюшки в виде работы с базой данных (в про версии), а вот для го я ничего такого не увидел что может emacs, при этом emacs работает в разы быстрее, стабильнее и полностью настраиваемый, написать расширение для емакса намного проще (пару строк на лиспе), если чего-то не хватает, а вот написать расширение для IDEA уже намного сложнее (я имею опыт написания расширений и под емакс и под IDEA)

swelf 21 сентября 2018 в 02:09

+1

> Используя тайловый менеджер без единой анимации, я на порядок быстрее работаю за своих коллег с их гномами или кедами, потому что пока они мышкой наведут на окно

Сколько ж ты по работе окна переключаешь, что эта операция на порядок рабочие задачи сокращает?

Я вот тоже поставил 27" 4к, подумал, удобно тайлинг, но тайлинг на самом деле немного ограничивает, а еще почти нет тайловых менеджеров с режимом tabbed, разве что i3, xmonad,ion3 на вкидку. Без режима tabbed я совершенно не вижу смысла вобще в тайлинге. Так вот, все мои потребности в тайлинге покрыли 3 key binding'а в fluxbox, которые ставят нужное окно в одну их 3 позиций на экране и добавляют в группу(чтобы были табы над окошками)

Crandel 22 сентября 2018 в 01:37

+1

> Сколько ж ты по работе окна переключаешь, что эта операция на порядок рабочие задачи сокращает?

Я сужу из реального опыта, подхожу к коллеге, он мне пытается что-то показать на гноме, пока мышкой дотянется до значка пайчарма, пока окно переключится с браузера на иде, пока он мышкой потянется к кнопке поиска, введет текст и опять же мышкой нажмет на кнопку поиск, я уже засыпаю. Зато когда они ко мне приходят, они не успевают заметить, что и как именно я поменял, а результат уже виден в браузере. Потом часто медленно показываю шаг за шагом. Хорошо, хоть научил уже переключать окна альт табом.

VolCh 23 сентября 2018 в 08:00

0

Следующий будет enter? или Ctrl+f?

Fractal1 21 сентября 2018 в 08:38

0

> Используя тайловый менеджер без единой анимации, я на порядок быстрее работаю за своих коллег с их гномами или кедами

Мантры, не имеющие никакого отношения к реальности.

Crandel 22 сентября 2018 в 01:39

0

> Мантры, не имеющие никакого отношения к реальности.

Для того, кто это делать не умеет. Скажите еще, что слепой набор — это детская сказка

0xd34df00d 22 сентября 2018 в 00:15

0

А у меня от emacs обратный опыт. Я, правда, spacemacs использовал (ну не умею я в emacs и не хочу тратить месяц на изучение elisp'а и настройку аналогов слоёв), а оно тормозит. Вот, то есть, адово тормозит. Я там пытался код на хаскеле и идрисе ковырять, так оно зависает секунд на 10, если, скажем, попросить сделать case split или вроде того. Простой ввод текста лагает.

В vim всё летает при этом.

Crandel 22 сентября 2018 в 01:46

0

Пользовался вимом 3-4 года назад и он реально тормозил с кучей плагинов для разработки. Может сейчас завезли асинхронность и ситуация стала получше, но все равно никто не отменит ужасный VimL из-за которого настроить под себя ничего реально не возможно. Лисп же намного проще, хотя некоторые неудобства со скобочками присутствуют. Плюс лично мне не подходят режимы, вот никак не получается работать быстро, несмотря на 2-3 года активного пользования вима в качестве основного редактора. Ввод текста у меня меня лагает только на файлах в одну строку либо на гигантских джейсонах, когда он анализирует структуру и подсветку делает. Код набираю со скоростью мысли, даже не задумываясь

0xd34df00d 22 сентября 2018 в 05:27

0

Ну да, в 8 завезли асинхронные плагины, и онлайн-тайпчекинг в idris-vim или произвольные онлайн-проверки через ale хоть плюсов с clang, gcc, cppcheck и clang-tidy одновременно, хоть хаскеля, хоть художественного текста — это просто сказка.

Но вимскрипты я не пишу, я достаточно скучный и обычный, чтобы всё, что мне нужно, делали имеющиеся плагины.

Ну и да, режимы наше всё, редакторы/IDE/etc без vim-стиля редактирования я даже не рассматриваю.

third112 25 сентября 2018 в 17:47

+1

> Возьмите любой продукт из 90-х, он написал ужасно, выглядит ужасно, и работает так себе.

Программы Вирта и сейчас являются образцом. GUI не всегда нужен.

SirEdvin 25 сентября 2018 в 17:48

0

Это какие?

third112 26 сентября 2018 в 07:18

0

См. книги и статьи Вирта. Вместе с другими он заложил основы структурного, модульного и ОО программирования. Его работы по ОС и компиляторам оказали огромное влияние на IT. В Вики большая статья, где множество ссылок, здесь ограничусь цитатой, имеющей непосредственное отношение к данному обсуждению:

> Возможно, одним из наиболее точных выражений принципов, которых придерживается Вирт в разработке всех своих проектов, является фраза Эйнштейна, вынесенная в эпиграф к «Сообщению о языке Оберон»: «Делай просто, насколько возможно, но не проще этого». Во всех его работах прослеживается изначальная ориентированность на реализацию наиболее эффективного решения конкретной инженерной задачи на базе гарантированно работающего, математически обоснованного инструментария. Вирт твёрдо стоит на том, что программирование должно быть нормальной инженерной дисциплиной, гарантирующей достаточный уровень надёжности своих разработок. Достижение же надёжности возможно, по Вирту, только одним способом: максимально возможным упрощением и самих систем, и инструментов, которые используются для их создания. В соответствии с этим принципом языки и системы программирования, разрабатываемые Виртом, всегда были образцом «разумной достаточности», даже своего рода аскетичности — в них предусматривалось только то, без чего нельзя обойтись.

SirEdvin 26 сентября 2018 в 11:23

0

Это все очень классно, но какими «образцами» вы пользуетесь сегодня?

third112 27 сентября 2018 в 16:37

0

Теми, что и вчера, т.к. фундаментальные принципы (стиль и т.д.) мало изменились. Примеры фрагментов моих кодов, скрипты.

SirEdvin 27 сентября 2018 в 16:41

0

Нет, вы сказали, что «программы» являются образцом. Вы имели ввиду, что они круто написаны и можно смотреть какой у них стиль или все-таки о том, что какие-то его программы дожили до наших дней?

Если первое, то это в целом очевидно, ничего особо нового с того времени не придумали, а если именно программы — то тут все печально.

third112 27 сентября 2018 в 17:13

0

Думаю, что и первое и второе. Нпр., я использовал интерпретатор P4 для скриптового языка в редакторе молекулярных графов, потом в боте. Не видел исходных кодов MacOS X, но по внешним впечатлениям многие основные принципы сохранились с 7.0 и более ранних, предполагаю что и коды модифицированы, а не все написаны с нуля.

Подозреваю, что и в «Виндах» остались коды от W-95 и от DOS. Нпр., что command.com и сору с нуля для каждой новой версии не пишут. И ОС Линукс довольно консервативная в основах. Для расчетов в мат.химии (нпр., топологические индексы) использую программы, которые делал в 90х. И т.д.

third112 27 сентября 2018 в 17:19

0

PS Из Вики:

> Разработка cmd.exe была фактически остановлена после выхода Windows 2000. Он всё ещё остаётся частью современных операционных систем Microsoft для персональных компьютеров (в том числе Windows 8, Windows Server 2012 и Windows 10) для обеспечения обратной совместимости. В качестве основной командной оболочки в этих системах рассматривается Windows PowerShell.

sumanai 27 сентября 2018 в 18:35

0

В десятке его внезапно начали обновлять.

third112 27 сентября 2018 в 18:43

0

С нуля?

sumanai 27 сентября 2018 в 19:02

0

Нет. Но сейчас вообще мало что с нуля пишут.

third112 27 сентября 2018 в 20:24

0

Да. И я про то, что мало с нуля пишут. А многое не с нуля уходит корнями в 90е.

SirEdvin 28 сентября 2018 в 10:22

0

И cmd.exe я стараюсь избегать всеми силами избегать, потому что он стршный и даже не умеет в масштабирование.

third112 28 сентября 2018 в 12:37

0

Иногда скрипты бывают удобны.

Alexeyslav 28 сентября 2018 в 13:25

+1

Для скриптов и придуман PowerShell изначально.

third112 28 сентября 2018 в 13:40

0

Изначально были DOS и command.com, далее необходимость совместимости.

0xd34df00d 29 сентября 2018 в 02:26

0

Я чё-т думал, что ООП — это Симула, а модули — это окамл. Вирт вроде ни там, ни там не участвовал.

third112 29 сентября 2018 в 07:26

0

Я сказал «Вместе с другими». Модула и Оберон оказали значительное влияние.

khim 25 сентября 2018 в 18:04

0

> Касательно машин это очень прохладная история, хотя бы потому, что КПД двигателей внутреннего сгорания все-таки ниже 100%.

Зато он очень близок к КПД идеальной тепловой машины. Вот когда вы сможете про ваш софт сказать **во что конкретно** упирается скорость вашей программе **в железе** — тогда и можно будет о чём-то говорить.

SirEdvin 25 сентября 2018 в 23:06

0

железо — это физика, в данном случае.
А идеальная тепловая машина — это эталонная программа.

Моя программа довольно близка к эталонной, потому что та написана точно по такой же схеме и с такими же проблемами. Профит?

Alexeyslav 26 сентября 2018 в 11:21

0

В данном случае, надо рассматривать не некую эталонную программу а ресурсоёмкость алгоритма который реализован в программе. Вот его надо сравнивать с реальной программой, насколько она близка к теоретическому пределу возможному на данном железе и в данных условиях(процессор, память, скорость хранилища данных). Точно так же как теоретический КПД ДВС зависит от атмосферного давления, температуры охлаждающего воздуха и температуры горения топлива. Если к примеру при прочих равных взять топливо которое даёт температуру 10000 градусов, можно получить двигатель в земных условиях с КПД 85%(допустим). Так же есть и теоретическая эффективность программы — полезная работа и накладные расходы(перемещение гигабайтов массивов данных даже в оперативной памяти не бесплатно).

Zoolander 20 сентября 2018 в 16:25

0

Нетрудно заметить, что во многих приведенных примерах, создателем плохого софта является вполне конкретная корпорация, у которой достаточно денег, чтобы делать хороший оптимизированный софт — но нет, она будет открывать и закрывать по 10 плохих сервисов, в надежде повторить удачу, которая случилась у них с их базовым продуктом

Gryphon88 20 сентября 2018 в 16:32

0

Немного не согласен. Сейчас модно давать площадку-маркет и завлекать туда ~~индусов~~ программистов.
Может, нужно разработать стандарт и следить за его выполнением?
Кстати, зачем мне программа размеров в 300 МБ, которая запускается на планшете, телефоне и ПК, если можно, используя один и тот же код и разные ключи компиляции, сделать 3 программы по 50 (пусть даже 100), под соответствующие устройства? Маркет сам поймёт, с какого устройства я зашел.

AlePil 20 сентября 2018 в 16:25

+2

То, что озвучил @tonsky действительно имеет место и отрицать бессмысленно.
Другой вопрос — что делать. Ведь если реально так посмотреть, то не во всем виноваты разработчики, есть еще бизнес, который хочет, что бы продукт в продакшен выкатывался еще вчера и не важно какой ценой. Есть пользователь, который абсолютно все равно, что там внутри ии как оно работает, но готовый платить за «докинуть RAM & HDD/SSD», за более мощную машину, но что бы было красиво, ну и есть он, программист, которому реально проще выдать продукт который в итоге поглотит все возможные ресурсы. Ну а что, юзер если что, докупит там чего-нибудь.

tonsky 21 сентября 2018 в 02:13

0

в рамках текущей экономической модели, действительно, точить пилу некогда, надо пилить. Видимо заниматься улучшениями должны люди вне её, как сейчас это происходит с опен сорсом и людьми в R&D отделах

Oskal174 20 сентября 2018 в 16:27

+6

Нельзя сказать, что я прямо сейчас загорелся идеей и проникся, но согласен, что грузить сайт по 20 секунд в 2к18 это дичь. Мне кажется, что проблема уменьшения производительности в том, что разработчики уходят все дальше и дальше от железа. Сегодня к железу перестали относиться с должным уважением.

ValdikSS 21 сентября 2018 в 02:14

+1

Скачайте и запустите Damn Small Linux. Он старый, софт в нем устаревший и не лучший, но просто понажимайте в интерфейс, позапускайте программы. Все работает **моментально**. Нажимаешь на кнопку — получаешь результат.
Я раз в год его запускаю и просто бесцельно тыкаю, чтобы не забывать, насколько компьютеры могут быть быстрыми, и что мы упускаем.

perfect_genius 21 сентября 2018 в 06:06

0

А Kolibri?

Fracta1L 21 сентября 2018 в 08:46

+1

Ну смотрите, в моём Линуксе с KDE программы запускаются не моментально, но за 0,5-2 секунды, в зависимости от «тяжести» приложения. Разве вот эта вот разница стоит того, чтобы менять комфортное и красивое окружение с фичами на каждый чих на топорный и малофункциональных Damn Small Linux? Компьютер нынче, как и 20 лет назад, 95% времени тратит на ожидание действий пользователя.

mkshma 21 сентября 2018 в 16:06

0

У вас там и SSD наверно, да? И оперативка не гигабайтами, а килограммами уже измеряется? Вот только большинство компов обладает такими скудными характеристиками, что просто запуск ОС уже заставляет их приближать глобальное потепление как минимум в рамках отдельно взятой комнаты.

Fracta1L 21 сентября 2018 в 18:05

0

Да, SSD, только ему уже 5 лет)) Нынче наличие SSD, даже копеечного, и 4-8 Гб оперативки это норма.

khim 21 сентября 2018 в 22:11

+1

> Нынче наличие SSD, даже копеечного, и 4-8 Гб оперативки это норма.

Не знаю у кого это норма. У меня есть знакомый, который в колледже обучается программированию. Он рассказывал что знает 1 (прописью: одного) парня из группы, у которого в компьютере есть SSD.

P.S. И да, он не из США и даже не из Японии. Могу поверить что там — ситуация иная.

springimport 24 сентября 2018 в 17:44

0

Да ладно. Топовые самсунговские ссд стоят сотню баксов. Не тысячу, сотню.
Какие-нибудь 128 можно вообще за полтинник найти.

В общем, было бы желание.

PsyHaSTe 25 сентября 2018 в 14:35

0

У одного парня из группы нашлась лишняя тысяча рублей на диск? Я конечно извиняюсь, но даже в студенческие годы со стипендией в 1500р/месяц я мог себе это позволить. Собственно я и купил первый SSD со стипендии, еще на 2 курсе, причем тогда он стоил не тысячу рублей а 60, а 7 при долларе по 30. Как-то за полгода скопил, и с тех пор не представляю, как можно вообще жить без него. Конечно, когда его покупаешь особо разницу видешь. Ну, сначала удивляешься, что винда устанавливается за 3.5 минуты вместо получаса, но все же. А вот когда пересаживаешься на комп с HDD разница становится заметна до боли.

khim 25 сентября 2018 в 17:47

+1

Проблема в том, что у большинства компьютеры — это что-то 5-7 лет от роду (причём и пять лет назад бывшее не топом), доставшееся от родственников, и воткнув туда SSD вы особого ускорения не увидите. А покупать целый новый комп — это уже другие деньги.

А 1500р/месяц (или, скорее, 1000р в месяц) уйдут скорее на телефон, а не SSD.

Вопрос ведь не в том «может ли студент позволить себе SSD, если напряжётся», а «является ли SSD нормой». Нет — не является. И не будет являться пока SSD не будут стоять в «офисных машинках на списание». Думаю лет через 10 — да, это таки случится. Но не сегодня.

Да и не нужен для Turbo C++ 3.1, на котором их обучают программировать (и нет, я не издеваюсь, вы думаете для кого подобные гайды пишут?) никакой SSD…

PsyHaSTe 25 сентября 2018 в 18:17

0

> Проблема в том, что у большинства компьютеры — это что-то 5-7 лет от роду (причём и пять лет назад бывшее не топом), доставшееся от родственников, и воткнув туда SSD вы особого ускорения не увидите.

Еще как увидите.

Ну а остальное — выводы, следующие из неверной посылки.
И речь не про 1500р/месяц, а про единоразовую трату.

Кстати, я так и не понял, как мы перешли от хайенд машин разработчиков до офисных машинок за 300$. Я очень надеюсь, что в продакшне никто не пишет на Turbo C++ 3.1 и без SSD.

khim 25 сентября 2018 в 18:44

+1

> Кстати, я так и не понял, как мы перешли от хайенд машин разработчиков до офисных машинок за 300$.

Вы хотите сказать, что люди на офисных машинка за $300 софтом не пользуются? Или что все должны отвалить за свою пишущую машинку по $20000-$30000, потому что кому-то лень свои программы оптимизировать?

Мы, я напомню, обсуждали вот это

> Нынче наличие SSD, даже копеечного, и 4-8 Гб оперативки это норма.

Если бы там было уточнение… скажем «наличие SSD и 32GB памяти для разработчика браузере — это норма», то тут можно было бы и согласиться. А если обсуждается «софт вообще»… так разумно его ожидать заточенным как раз под «офисные машинки за $300»…

> Я очень надеюсь, что в продакшне никто не пишет на Turbo C++ 3.1 и без SSD.

Это скорее про «ужасный софт из 90х, которых страшен и которым неудобно пользоваться». По-большому счёту к интерфейсу Turbo C++ 3.1 у меня единственная (но катастрофическая) претензия: на экране с разрешением 2560 x 1600 хотелось бы видеть чуть больше текста, чем Turbo C++ 3.1 умеет. Вот к компилятору — да, претензий побольше.

PsyHaSTe 25 сентября 2018 в 19:59

0

> Вы хотите сказать, что люди на офисных машинка за $300 софтом не пользуются? Или что все должны отвалить за свою пишущую машинку по $20000-$30000, потому что кому-то лень свои программы оптимизировать?

Вы видимо читаете через строчку, если не через коммент.

Корневое предложение было «давайте разрабам дадим машинки по 300$ тогда они будут оптимизировать лучше».

> Если бы там было уточнение… скажем «наличие SSD и 32GB памяти для разработчика браузере — это норма», то тут можно было бы и согласиться. А если обсуждается «софт вообще»… так разумно его ожидать заточенным как раз под «офисные машинки за $300»…

Об этом речь и шла.

khim 25 сентября 2018 в 21:06

0

> Вы видимо читаете через строчку, если не через коммент.

А вы, похоже, вообще не читаете.

> Корневое предложение было «давайте разрабам дадим машинки по 300$ тогда они будут оптимизировать лучше».

Это вы вообще с другой статьёй перепутали. Или другой веткой, возможно.

В этой о машинах для разработчиков начали говорить вы — причём сразу с претензиями «а чагой-то мы с темы съехали».

> Если бы там было уточнение… скажем «наличие SSD и 32GB памяти для разработчика браузере — это норма», то тут можно было бы и согласиться. А если обсуждается «софт вообще»… так разумно его ожидать заточенным как раз под «офисные машинки за $300»…
>
> Об этом речь и шла.

Если речь шла об этом, то достаточно взглянуть на статистику, чтобы понять — да, SSD перестали быть экзотикой, но если у нас **в среднем по больнице** компьютеры с HDD продаются чаще, чем компьютеры с SDD, то понятно, что среди «пишущих машинок за $300» — и подавно.

И не нужно тут обсуждать «хорошо это или плохо» — разработчикам программ просто нужно это учитывать.

PsyHaSTe 25 сентября 2018 в 21:59

0

> Это вы вообще с другой статьёй перепутали. Или другой веткой, возможно.

Да, перепутал с другой веткой. Прошу прощения.

Домашний пользователь SSD действительно скорее всего не имеет.

VolCh 26 сентября 2018 в 10:32

0

Как раз на сатрых компах с небольшим объёмом оперативки прирост от SSD очень заметен на повседневных домашних и деловых задачах.

Alexeyslav 26 сентября 2018 в 11:50

0

Пробовал ставить SSD на довольно старенький комп 4Гб WinXP так он просто дико тормозил. Проблема была именно в интерфейсе с SSD — какие-то дикие тормоза на ровном месте. Поставил уже Win7 x64 и всё без проблем залетало. А этой системе 4Гб в принципе мало очень, голая операционка работает а как что запускаешь более-менее серьёзное так всё — недостаток памяти, свопинг и толку от SSD ровно ноль.

PsyHaSTe 26 сентября 2018 в 17:45

0

Эффект от установки ссд за 2к заметнее, чем от процессора на 400$ дороже

Alexeyslav 26 сентября 2018 в 23:24

0

Нифига. Тормозит всё так же безбожно, только теперь единственное что работа с большим количеством мелких файлов ускорилась, но с RAM-диском не сравнится всёравно.

PsyHaSTe 27 сентября 2018 в 21:58

0

Ну вот у меня опыт противоположенный. Старый комп с Prescott пнём получил второе дыхание, как и несколько компов на старых атлонах.

0xd34df00d 25 сентября 2018 в 20:30

0

Завидую. Стипуха в 2200 и, чуть позже, аналогичный полупассивный доход едва покрывали потребности в еде, интернете и поездках раз в месяц в город.

PsyHaSTe 25 сентября 2018 в 22:01

0

Ну, я часто ел у знакомых, а так бюджет на день был 100р — 3 булочки с сахаром по 17 рублей и поллитровая пачка сока за 40. Но понемногу деньги копились, особенно если экономить почти половину суммы, съедая всё всухомятку.

Но, слава богу, то время мы пережили, а теперь можно немного расслабиться и перестать считать, насколько завтра мне придется меньше есть, если сейчас я возьму еще вон тот вкусный пирожок :)

slg 20 сентября 2018 в 16:27

+1

Продукты выходят быстрее, снимая сливки. Покупатели в постоянной фрустрации, заставляющей покупать новинки. Экономика растет, только вы чем-то не довольны.

staticmain 20 сентября 2018 в 16:27

+1

> Так что Slack — это пример очень ресурсоёмкого приложения.

Странные у вас скриншоты slack. Это вообще без подключенных команд? У нас 200 человек + 25+-10 рабочих комнат — веньше 1.5 ГБ этот «мессенджер» не отжирает в принципе.

Massacre 20 сентября 2018 в 16:48

+13

Это (и Skype 8.x) является отличным примером, почему нельзя делать десктопный софт на JS.

Neikist 20 сентября 2018 в 16:54

+10

Как и мобильный.

Alexey2005 20 сентября 2018 в 17:25

+7

На самом деле можно. Только создать шустрое, отзывчивое десктопное приложение на JS ничуть не проще и даже не быстрее, чем на C++.
Качественный профессиональный код писать на JS дорого — ничуть не дешевле, чем на тех же плюсах. Зато вот дерьмовый код на JS из серии «лишь бы работало» писать гораздо, гораздо дешевле и проще. А значит, появляется возможность сэкономить за счёт снижения качества. С плюсами экономить на качестве удаётся гораздо хуже — даже самый дерьмовый код всё равно довольно дорог.

looogle 1 октября 2018 в 11:10

0

Ну а Discord тогда является примером того, почему на самом деле это делать можно.

Massacre 1 октября 2018 в 13:14

0

Нет, от 200мб в памяти для мессенджера, всё же, многовато.

asmaster 21 сентября 2018 в 01:17

+1

Я почти завершил разработку 300 КБ Slack (и не только) клиента

volt.ws

SirEdvin 21 сентября 2018 в 01:19

0

Мм… как у вас с кнопками и событиями из слака?

Massacre 21 сентября 2018 в 03:20

0

Интересная идея… Добавил в закладки. Если будет поддерживать всё, что имеется в Telegram Desktop (и делать это без утечек памяти) — станет хитом. Ещё можно подтянуть оставшиеся протоколы из Миранды (особенно, ICQ), и она тоже станет не нужна :)

Zloy_tarakan 20 сентября 2018 в 16:27

+3

Эй, ребят, а может ну его, лучше в космос сгоняем.

Gryphon88 20 сентября 2018 в 16:28

-2

Это нытьё я считаю обоснованным, жаль, что никто, включая автора, не будет разгребать сложившуюся ситуацию. Подскажите, пожалуйста, какие-нибудь расширения для браузера, чтобы лишнее просто не грузилось (а не скрывалось), в идеале, препроцессировалось под текущую комплектацию (есть у меня смутное ощущение, что если написать очень косо, то при каждой прокрутке экрана будут пересчитываться относительные размеры).

Massacre 21 сентября 2018 в 03:30

0

uBlock Origin? Там можно достаточно гибко настроить, что именно не должно грузиться. Правда, «лишнее» придётся прописывать вручную, если это не распространённая реклама…

miga 20 сентября 2018 в 16:35

+4

Тонский опять ноет.

На самом-то деле, наш софт и железо работают *поразительно* хорошо, если просто призадуматься, насколько глубока бездна абстракций, которую несколько поколений инженеров тщательно выстраивали. Взять, например, тривиальный пример того, что происходит, когда пользователь набирает «google.com» в адресной строчке и нажимает enter — весьма поверхностное описание занимает девять страниц, и это еще не опускаясь ниже уровня драйвера клавиатуры (ниже еще одна бездна аппаратных абстракций).

Не то чтобы я был доволен индустрией и ее нынешним (хаха, нет), но все же мне надо понимать, почему получилось именно так и почему иначе и быть не могло.

И еще заодно можно, например, взять машинку на DO, поставить туда докер с вордпрессом, зайти со своего браузера на дефолтную заглавную страничку и испытать экзистенциальный ужас от количества и сложности усилий и объема знаний, которые были приложены многими тысячами не самых глупых людей чтоб вы могли совершить эти, в общем-то, весьма тривиальные действия. Ставлю на то, что проекты атомной бомбы и полета на Луну вместе взятые были проще и дешевле.

InterceptorTSK 20 сентября 2018 в 22:34

–2

Ага. Ток тормозит безбожно.

Я мозиллу открываю, набираю в строке поиска что-то, а оно подвисает. Вы объясните, какими такими технологиями можно сообразить НАСТОЛЬКО ЛЮТОЕ ГАВНО???

У меня на мобильной джаве собственная реализация такого же поиска да еще и с собственным т9 вообще не тормозила. Это НЕВОЗМОЖНО сделать так, что бы такая примитивщина тормозила. Вы что правда не понимаете, что это примитивнее ну просто некуда?
И на мобильной джаве я реализовал даж клеартайп, и все писалось в моем текстбоксе, искалось по т9 и вообще не тормозило никак.
И это на устройстве с 20мб оперативы (если повезет), и процессором в несколько мегагерц (или десяток мегагерц, если повезет).

Вы объяснитесь хоть как-то. Вы с луны свалились совсем что ли, и головой ударились об нпм пресильно?

ImKremen 20 сентября 2018 в 22:36

+1

Не просто проще, а на порядки проще.
Есть у меня немолодой коллега (на данный момент фронтенд), программировал он на всей цепочке языков начиная с перфокарт. Так вот он приводил пример диалога с неким инженером который разрабатывает нетривиальный детали (если не ошибаюсь ракетный инженер), суть вопроса звучала примерно так — "зачем вам целый штат тестировщиков, вы что не можете сразу делать без ошибок?". Ответ прост, система на порядок сложне. Даже отдельный фичи проекта со всеми приложеными требованиями могут не помещается в голову разработчика.

khim 21 сентября 2018 в 00:50

+1

> Ответ прост, система на порядок сложне.

А **почему** она на порядок сложнее? Мы тут как-то уже обсуждали вопрос: вот что такое новое и ультраполезное появилось в MS Office после MS Office 2000… Единственное, что удалось вспомнить — это OOXML, и то «Compatibility Pack» есть и работает в Office 2000.

Так почему MS Office 2016 занимает в 10 раз больше, чем MS Office 2000? За что мы платим этим «увеличением сложности»?

> Даже отдельный фичи проекта со всеми приложеными требованиями могут не помещается в голову разработчика.

А они их туда пробовали поместить? Мне кажется вся система «пошла в разнос» где-то как раз на границе веков, когда вместо того, чтобы бить задачу на подзадачи, которые человек мог бы понять её стали шинковать на мелкую лапшу, работу которой не понимает вообще никто, а в качестве суррогата понимания — её обматывают толстым-толстым слоем тестов.

Всё равно как если бы машины собирали не из отдельных деталей, подогнанных друг к другу, а из кучи песка, обмотанного скотчем и тряпками…

SirEdvin 21 сентября 2018 в 00:57

+1

> А почему она на порядок сложнее? Мы тут как-то уже обсуждали вопрос: вот что такое новое и ультраполезное появилось в MS Office после MS Office 2000… Единственное, что удалось вспомнить — это OOXML, и то «Compatibility Pack» есть и работает в Office 2000.

Ну почитайте. Вы, конечно, можете сказать, что это все чепуха и не нужно, но вы же в ворде не верстаете.

khim 21 сентября 2018 в 07:21

0

> Вы, конечно, можете сказать, что это все чепуха и не нужно, но вы же в ворде не верстаете.

А что это меняет? Понимаете — если фича вам **нужна** и **полезна**, то вы про неё быстро вспомните. Вот, например из подобного же списка:
Source files (.cpp) belonging to CMake projects can be built individually via the CMake menu or by right-clicking on a source file in the Solution Explorer.
Я могу легко объяснить — почему и кому она может быть полезной (объяснить почему для неё нужно было создать такого монстра как Visual Studio 2017 я не могу — но это другая история).

А если всё, что вы можете сделать — дать ссылку на портянку текста без указания в ней конкретного места… то это значит, что, по большому счёту, ничего **реально** важного там нет… было бы — вы бы запомнили.

Am0ralist 21 сентября 2018 в 16:24

0

> А что это меняет? Понимаете — если фича вам нужна и полезна, то вы про неё быстро вспомните.

Неа, если ты уже кучу лет сидишь на 2010, то вспомнить, что было не так с 2003 уже сложно. Ты уже все фичи воспринимаешь, как нужные и естественные…

khim 21 сентября 2018 в 22:15

0

> Неа, если ты уже кучу лет сидишь на 2010, то вспомнить, что было не так с 2003 уже сложно.

То есть вы хотите сказать что тут все работали с MS Office 2016 и ничего старше не видели?

> Ты уже все фичи воспринимаешь, как нужные и естественные…

Для этого на Хабре есть комментарии. Назовите фичу — а люди, работавшие с предыдущими версиями скажут когда она появилась. Пока видел фичи из MS Office 2000. Ничего более позднего из числа «реально полезных фич» названо не было.

Am0ralist 21 сентября 2018 в 23:30

0

Наоборот, ты запускаешь старую версию и понимаешь, что некоторые вещи не работают и ты чертыхаешься. Это можно запомнить, если ты постоянно работаешь во всех версиях (97, 2003, 2007, 2010 и 2016), а не когда ты прошел их все за лет 16, причем не на постоянке используя.

khim 22 сентября 2018 в 03:44

0

> Наоборот, ты запускаешь старую версию и понимаешь, что некоторые вещи не работают и ты чертыхаешься.

Ну так запустите и «ужаснитесь». Не знаю про MS Office 97, но все версии начиная с MS Office 2000 под Windows 10 встают и нормально работают, можно впрямую сравнить.

Am0ralist 22 сентября 2018 в 10:30

0

Ну так, периодами сталкиваюсь с экселем 2003 и новыми.
10 намного удобнее, в мелочах, которые только по факту понимаешь, когда начинаешь делать по одному, а его там нет.
Ну или с 40к строк работаешь.
С 16 разницу не заметил особую, но последний мне не очень зашел. Хотя менюшка верхняя — до сих пор выморáживает.

Это как калк в OOo и Либре сравнивать. Вроде функции одни и разница незначительная, но в OOo по мне куча всего по мелочам хуже сделано.

khim 23 сентября 2018 в 17:45

0

> Вроде функции одни и разница незначительная, но в OOo по мне куча всего по мелочам хуже сделано.

Да, но и потребление ресурсов у OOo и LibreOffice сравнимо (причём не факт, что у LibreOffice больше). В этом случае «куча всего по мелочи» — нормальное оправдание.

А когда котребление ресурсов вырастает **в разы** — хочется чего-то большего.

CodeRush 21 сентября 2018 в 00:59

+1

Потому что из песка, скотча и тряпок ее делать дешевле в соответствии с теми процессами, которые идут внутри компании, эту систему производящую.
Чтобы бить задачи на подзадачи, нужно ясное видение «что именно строим», и план детальный к нему. По нынешней же моде это все называется жутким словом «водопад» и преподносится дорогими консультантами как жуткое болото, страшный суд, ад и погибель (о том, что так делается любой инженерный проект от моста через реку до ракеты, консультанты умалчивают почти всегда). Теперь у нас в моде эджайл, скрам, канбан и прочее все, призванное ПО не разрабатывать, а выращивать, лепя MVP из песка, тряпок и скотча, чтобы релизить быстрее, потому что иначе деньги кончатся, а инвесторы не дадут новые.

PsyHaSTe 21 сентября 2018 в 01:07

+1

Agile и DDD это про то, чтобы получить на выходе продукт, который всем нужен, а не тот, который согласовали когда-то на заре. И получить в приемлемые сроки. Я поработал по обеим методологиям, по водопаду сделанную софтину мне через 3 года жалко было даже запускать. И спустя эти 3 года от неё отказался и заказчик, и соответственно мы. Потому что спеку мы всю удовлетворили, все требования реализовали, но пользоваться этим реальные пользователи не захотели.

Лучше узнать это до того, как отдал сотню миллионов, полагаю. А еще лучше получить рабочий

и удобный продукт. Agile говорит только про «имейте частый фитбек от заказчика», он ничего не говорит про «фигак фигак инвестиции». Если вы не согласны, то получается ваша позиция «надо зафиксировать ТЗ, а потом идти по нему, даже если очевидно, что оно перестало всех устраивать»? Ну ладно…

CodeRush 21 сентября 2018 в 01:19

+1

Именно в этом и разница между инженерной задачей и задачей получить на выходе продукт, который кому-то нужен. И именно поэтому софт для спутника или марсохода пишется по спеке и исключительно по ней, а писать так приложения для бизнеса, который сам не знает, чего хочет — выбросить сотню миллионов на ветер, и остаться позади конкурентов. Разные цели — разные методы, и я не ругаю ни тот, ни другой, просто отвечаю на вопрос «почему результат вот такой» — он вот таким получается, получать его иным — дорого.



PsyHaSTe 21 сентября 2018 в 02:11

+2

Я к тому, что "жуткое болото, страшный суд, ад и погибель" — вполне себе имеет место быть в реалиях бизнеса.

У меня сейчас небольшой проект на год, и даже тут я ощущаю пользу Agile, когда я сделал первый MVP-прототип, показал, мне сказали "не, ты все не так понял", и мы сели и за пару недель исправили тонну недочетов, которые впоследствии потребовали бы пересмотра всей архитектуры.

Лично для себя минусов я не вижу. Бывает, что иногда приходится что-то переделывать, чтобы удовлетворить новые требования, но я стараюсь находить в этом обучающий момент, что я плохо собрал требования/не задал нужных вопросов/… А вот с водопадом я покушал проблем, потому что опять же, тремор требований у бизнеса слишком высокочастотный, чтобы его таким образом формализовывать. Рад за мосты и инженерные проекты, но бизнес завязан на людей, которые тоже меняют мнение. Вчера оформили подряд на новый крутой кнопочный телефон на тонну денег, а на следующий день эппл презентует первый айфон и всем срочно нужен телефон с сенсорным экраном… А спека-то уже всё…

Ну, это чисто моё мнение. Может, есть проекты, где люди тащатся от того, как год за годом однажды начерченный план претворяется в жизнь и становится продуктом, но мне они не попадались, только те, который делали вид, что они такие, но заканчивалось это как я написал выше.



CodeRush 21 сентября 2018 в 02:20

+1

Так я и не спорю с этим, в реалиях бизнеса все так и есть. Проблема водопада именно в том, что он несовместим с качанием требований вслед за линией партии, но это не делает саму методологию плохой, а лишь неподходящей к этим реалиям. Т.е. если цель — решить задачу бизнеса, то решать ее надо как получается, и не жаловаться на тормоза и неоптимальность решения. А хочешь оптимально — доставай строгие требования, математику, теорию систем, инженерию, все то, что мы как человечество наработали для решения проблем, влезающих в головы архитекторов.

SirEdvin 21 сентября 2018 в 01:08

0

Наверное, вы никогда не работали по "водопаду", да?)

Если вы не можете справится с аджайлом, где на каждом этапе должны планировать маленькие кусочки системы, думаете у вас получится спланировал сразу все? Спойлер — не получится, просто вы будете каждый раз переделывать спеку, если будете, конечно.

> о том, что так делается любой инженерный проект от моста через реку до ракеты, консультанты умалчивают почти всегда

Помимо того, что ПО, как бы это глупо не казалось, безумно сложнее в проектировании, чем мост или ракета, любой инженерный проект страдает от тех же проблем, что и ПО при водопаде, но в силу условной простоты проекта это костылят на местах.



CodeRush 21 сентября 2018 в 01:25

0

Работал и так, и так, и справляться приходилось и с водопадом, и с аджайлом.

Если у вас задача не влезает по сложности в дизайн — это показатель того, что решать ее рано. Ах, надо уже вчера? Ну тогда никто не гарантирует никакого качества, кроме того, что получится. Именно так получается нынешнее ПО, нынешние процессоры и все остальное. Нужно лучше — планируйте, контролируйте, используйте подходящие инструменты, контролируйте, хорошо платите, контролируйте, и не забывайте контролировать, это обязательно.

Пока это все не станет выгодно (т.е. ничем другим от конкурентов отличиться станет невозможно, кроме размера и/или скорости кода), для масс это не будет сделано никогда, потому что это адски дорого, адски долго, а на выходе получаются только примитивные вещи, влезающие в дизайн по ТЗ.



ImKremen 21 сентября 2018 в 08:13

0

Ну возьмите частную задачу, например построение UI и его layout с учётом всего зоопарка размеров и dpi современных экранов.



Vanellope 21 сентября 2018 в 15:06

0

Когда то была на хабре статья про миллионы людей, занятых бессмысленной работой, перекладыванием бумажек, только ради работы. Так вот АйТи стремится примерно к такому же. Программисти пакет уже написан десяток лет назад, обладает всей необходимой пользователям функционалностью, казалось бы, отшлифовывай, патчь баги, выполняй пожелания пользователей в плане разных фич. Но надо же показать продуктивность деятельности, надо релизить новые и новые версии каждый год, добавлять всякие маджонги и поэтесс, чтобы не отстать от конкурентов, заставлять пользователей пересаживаться на новый продукт, т.к. теряется совместимость со старыми версиями, старые продукты перестают поддерживаться, не работают на древних ОС, типа XP и прочее, и прочее, и прочее.
Так как все программы уже давно написаны, революционной новизны ждать не приходится, получается именно работа ради работы.

springimport 20 сентября 2018 в 16:42

-1

Мне тоже не нравится структура папок в линуксе. Я бы даже сказал что она ужасная, она стала слишком низкоуровневая на сегодня. И лучшим подтверждением этого есть docker.
Идеальным решением видятся контейнеры по типу мобильных приложений: соседние контейнеры могли бы требовать любые библиотеки без конфликта версий, поддерживать многоверсионность и разные источники для обновления, иметь разделение данных на внутренние и пользовательские, использовать интерфейсы других контейнеров.

Чтобы в конце-концов можно было запускать контейнеры десятилетней давности на новейших системах и перенос данных с любого компьютера на другой без каких-либо препятствий. А не так как сейчас что после установки вообще не ясно что произошло.

staticmain 20 сентября 2018 в 16:51

+6

И у каждой программы тогда будет по копии системных библиотек. Отлично. Теперь каждая программа будет весить 1-5 ГБ, даже обычный калькулятор. Вы изобрели flatpack. В средней системе установлено 50-80 программ, которые доступны пользователю и еще около 100 системных утилит. И каждая с вашим решением будет весить больше гига. Ну да, у всех же стоят 5 ТБ SSD последнего поколения и google fiber чтобы всю эту ораву обновлять.

springimport 20 сентября 2018 в 16:59

0

Стоп. А где я говорил что реализация будет виндовой? Никаких копий, только ссылки. А при необходимости добавляется дополнительный контейнер с кастомным пакетом который может, опять же, использоваться другими пакетами. Соответственно, в памяти будет условный набор из 100 контейнеров-библиотек на которые ссылаются все остальные программы.

Нет, серьезно. Что угодно может быть лучше чем запихивание отдельной ОС для каждой программы.

staticmain 20 сентября 2018 в 17:15

0

> Соответственно, в памяти будет условный набор из 100 контейнеров-библиотек

Вместо одного, ага. Ваше потребление памяти выросло в 100 раз. Сейчас оно худо-бедно требует 3-4 ГБ для системных\повседневных приложений (специфичные типа blender/krita/uengine я не считаю), будет требовать 300-400 ГБ оперативки. Шикарный план.

> Что угодно может быть лучше чем запихивание отдельной ОС для каждой программы.

А создатели flatpack/snap так не считают. И это сейчас впиливается в новые линуксы по-умолчанию.

springimport 20 сентября 2018 в 17:23

0

Чего-то я не пойму, а что, сейчас разве не так? Разве хром не загружает в память, например, zip library 1.0, в то время как проводник загружает zip 1.25?
Да, будет не очень если каждое приложение захочет zip 1.0, 1.01, 1.02, но с этим же борются, например, сейчас: ставят не 1.0, а условно ^1.*

staticmain 20 сентября 2018 в 17:25

0

Не так, если chrome у вас из системного репозитория, в котором стоит только одна версия zip. Если вы какой-то пакет ставите в обход репозиториев, то вы ССЗБ.

polearnik 21 сентября 2018 в 11:47

0

это вы сейчас про npm и node_modules?

PsyHaSTe 20 сентября 2018 в 22:21

0

По-моему докер со своими слоями эту проблему нормально решает. Каждая программа будет весить ровно столько, сколько ей нужно. Единственное чего не будет, что нельзя будет использовать либу версии 1.2 с программой, скомпилиной под 1.1 потому что «ну там ведь небольшие изменения, да и минорная версия поменялась, наверное ок».

SirEdvin 20 сентября 2018 в 16:51

+1

Не получится, так как бинарная совместимость таки ломается иногда между версиями.

springimport 20 сентября 2018 в 17:11

0

Таких подробностей не знаю… Есть подозрение что все же это можно решить в рамках одной платформы.

Akon32 20 сентября 2018 в 17:04

0

Насколько я понимаю, структуру каталогов в Linux можно поменять, сделав свой дистрибутив.

И есть такая штука, как Qubes OS. Это ОС с песочницами для отдельных приложений на базе xen и linux. А в каком-то дистрибутиве (не помню точно) вообще конфигурацию ОС в git хранят.

ValdikSS 21 сентября 2018 в 02:27

+1

Flatpak, Snap, в какой-то степени AppImage, ZeroInstall.

engine9 20 сентября 2018 в 16:52

+11

Enjoy your capitalism!

В какую помойку превращают маркетологи нашу планету в такую превратили и IT.

Я всю трудовую жизнь так или иначе связан с компьютерами, разработкой GUI и компьютерной графикой, и эта тенденция начала напрягать уже давно. Открыл для себя «островок умиротворения» в мире линукса, который год от года все больше радует инструментарием. Вот сейчас в отпуске работаю на машине с процессором 900 мГц, и гигом оперативы на «минте» — система летает, в памяти занимает 230 мБ! Поставился весь софт и дрова из репов, никаких танцев с бубном. Интерфейс вылизан, отзывчивость отличная.

На большом брате создаю 3Д в блендере и каждый раз искренне благодарен его создателям, **сто мегабайт** мощнейший 3Д пакет, в котором можно моделировать, монтировать виде и аудио, композить, даже создавать 2Д анимацию!

Я очень рад, что максимально дистанцировался от всей этой монструозной проприетарной параши напичканной бэкдорами и жучками, создатели которой пишут код за зарплату, а не за идею ~~и потом уже зарплату~~. Ну и в современный веб захожу в костюме химзащиты, позволяющем пробраться через увесистые гроздья яваскрипта, облепившие интернеты, и прочие свистопердельные миазмы, которые отдаляют от контента.

Akon32 20 сентября 2018 в 17:09

+3

> сто мегабайт мощнейший 3Д пакет

Может, я чего-то не понимаю, но у меня свежий Blender под windows весит 516МБ. Это не считая драйверов видеокарты и прочего, что позволяет ему работать.

Вы точно учитываете при подсчёте все библиотеки?

engine9 20 сентября 2018 в 17:24

+3

Я сужу по установщику с оффсайта: www.blender.org/download 137 мб (под пингвина)

bogolt 22 сентября 2018 в 13:33

0

Современные установщики частенько качают реальный софт из сети.

sumanai 22 сентября 2018 в 19:25

0

Обычно такие качальщики весят от силы мегабайт.

terra-slav 20 сентября 2018 в 18:14

+1

В Arch Linux:
[user]# pacman -S blender
Пакеты (1) blender-17:2.79.b.git3.32432d91-6

Будет установлено: 246,13 MiB
:: Приступить к установке? [Y/n]

Alex_ME 21 сентября 2018 в 03:17

0

У меня установленный Blender 2.79b занимает около 300 мб.

ValdikSS 21 сентября 2018 в 02:28

+1

> Ну и в современный веб захожу в костюме химзащиты, позволяющем пробраться через увесистые гроздья яваскрипта, облепившие интернеты, и прочие свистопердельные миазмы, которые отдаляют от контента.

Попробуйте Weboob (Web out of browsers), но придется написать кучку плагинов.

namikiri 20 сентября 2018 в 16:55

+14

Забавно читать перевод с английского на русский статьи, написанной русским автором.

vasimv 20 сентября 2018 в 16:58

0

«But I predict that within 100 years, computers will be twice as powerful, 10,000 times larger, and so expensive only the five richest kings in Europe will own them.» (c) Simpsons. Промахнулись чуток, не про компьютеры это надо было говорить, а про программы :)

VezhLos 20 сентября 2018 в 17:10

0

Полностью согласен с автором статьи, однако описанное положение вещей — это плата за низкий порог вхождения в отрасль и возможность для бизнеса быстро выкатывать требуемые функции. Хотя, конечно, с инженерной точки зрения текущее положение дел в IT — это «Адъ и Израйль». Мыши колются, плачут, но продолжают жрать кактус, ибо выхода нет.

Areso 20 сентября 2018 в 21:04

0

Не согласен. Программировать на Pascal (в среде Delphi 5) было ничуть не сложнее. Даже проще, я бы сказал. У нас Basic'у (QBasic 45R) учили пятиклассников.

VezhLos 20 сентября 2018 в 22:35

+1

Хелловорлды, может, и не сложнее, а вот коммерчески ценный мало-мальски сложный софт на каком-нибудь Электроне «начинающему перспективному программисту» накидать гораздо проще, чем на нормальном языке. Но и тормозить будет знатно, конечно, и ресурсы жрать, как не в себя… Такая вот борьба противоположностей.

P.S. Во всей этой хипстерской вебо-мобильной фреймворкщине и яваскриптщине вымораживают именно жуткие тормоза и потребление ресурсов. А чтобы написать нормальный десктопный аналог на C++ или Delphi, ну или на крайняк Java/C#, нужна нехилая квалификация и время.

khim 21 сентября 2018 в 07:28

+1

> А чтобы написать нормальный десктопный аналог на C++ или Delphi, ну или на крайняк Java/C#, нужна нехилая квалификация и время.

Насчёт времени я не уверен. Не вижу я, чтобы фичи появлялись быстрее, чем во времена, когда всё делалось на Visual Basic'е (**не** Visual Fred) или Visual C++ 6.0. Вот не вижу и всё.

Только разработка идёт по другому: если раньше вы делали, условно, 10 формочек за 10 дней — и релизились, то теперь вы делаете их за один, а ещё 9 дней тратится на то, чтобы полученный монстр хоть тушкой хоть чучелом заставить работать и убрать хотя бы часть глюков.

Mabusius 21 сентября 2018 в 12:09

0

Аналогично. В школе нас учили на Турбо Паскале. Даже троешники чето там могли. А в ВУЗе уже пошел Delphi 6 с обязательным ООП даже для «Hello World». Большинство сидело и не вдупляло, что вообще происходит.

mayorovp 21 сентября 2018 в 13:35

0

Обязательный ООП в Delphi только в графических программах. И он, на самом деле, не усложняет, а упрощает написание программы. Без ООП этот самый «Hello World» испугает даже отличников…

Mabusius 21 сентября 2018 в 13:54

0

Я то согласен, я к тому времени пару игр на Турбо Паскале написал и про то как работают объекты был в курсе. А вот простым людям намного легче понять код типа $a := $a + 1;, чем $form1.label1.text := $form1.label1.text + 1;.

mayorovp 21 сентября 2018 в 13:57

+1

Не видел ни одного человека, который бы испытывал трудности с оператором «точка»…

Mabusius 21 сентября 2018 в 14:57

0

Ну там возможно еще была проблема в преподавании. Никто не объяснял, что это за точка :). ООП вообще никак не объяснялось довольно долго, но использовалось.

VolCh 23 сентября 2018 в 08:22

0

В структуры +record вроде в Паскале как поднимались?

lokiB 20 сентября 2018 в 17:15

+1

Неужели программисты начали замечать, что закон Мура больше не работает?

dimka11 20 сентября 2018 в 21:07

0

Все работает. Количество ядер в процессорах растёт. Техпроцесс становится меньше.
То, что пока не все приложения могут в многопоточность — вопрос времени.

PsyHaSTe 20 сентября 2018 в 22:23

+2

> То, что пока не все приложения могут в многопоточность — вопрос времени.

Этой мантре уже лет 20, но воз всё там же. Понемногу где-то наращивают, но воз и ныне там. Только софт от программистов для программистов, типа гитов, архиваторов, идешек и прочих умеет грузить все ядра, а остальной радуется, если смог сделать отдельный тред для трудоемких вычислений.

Fracta1L 21 сентября 2018 в 08:51

+1

Видеокодеки многопоточны, браузеры многопоточны, игры многопоточны (хоть и не сильно — как правило, до 6-8 потоков). Где там воз, говорите?

sumanai 21 сентября 2018 в 10:48

+1

> браузеры многопоточны

Яваскрипт на странице однопоточен.

EvilFox 21 сентября 2018 в 16:45

0

Но, есть же веб-воркеры.

sumanai 22 сентября 2018 в 10:45

0

Работа с DOM всё равно однопоточна, а без неё ничего на странице не изменить. В итоге изменение страницы при помощи Javascript возможно только в один поток, и это самая медленная часть в современных страницах.

EvilFox 22 сентября 2018 в 15:06

0

И что? Большая часть тормозов из-за того что DOM используют как хранилище данных. При не умелом использовании DOM многопоточность вас не спасёт.

lRainman 22 сентября 2018 в 12:45

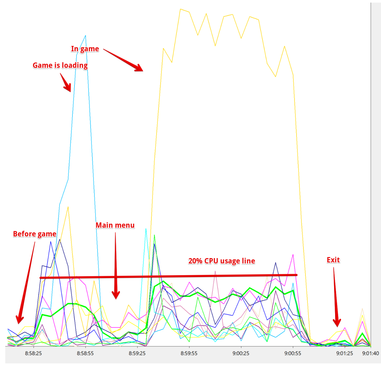
0

Но браузер то нет, более того они уже поголовно многопроцессорные.

PsyHaSTe 21 сентября 2018 в 13:06

0

Браузеры однопоточны, игры тоже: ну чтобы далеко не ходить, вот пример из Stellaris: 80% работы делает одно ядро



Так что воз и правда там.

Fracta1L 21 сентября 2018 в 13:17

0

> Браузеры однопоточны

Почти в каждом релизе Firefox пишут, что распараллелено то, распараллелено сё, где ж они однопоточны?

> вот пример из Stellaris: 80% работы делает одно ядро

А когда я играю в Cities Skylines, у меня стабильно нагружены на 100% 4 потока из 8. В новых AAA-играх типа GTA V многопоточность ещё более агрессивная.

EvilFox 21 сентября 2018 в 16:49

0

> Браузеры однопоточны

Только отчасти. Есть же веб-воркеры.

> игры тоже

Сильно от игры/движка зависит. Современные движки уже многопоточны. Даже игровую логику можно сделать многопоточной, например в юнити есть C# Job System.



PsyHaSTe 21 сентября 2018 в 18:30

0

Я в основном играю в стратегии, так вот европа/стелларис — современные игры 2013 и 2016 годов. И с ними та история, как я привел выше.

Из реально тяжелых игр с хорошей оптимизиацией могу вспомнить только новый дум. Во всех остальных игра сидит на 1 ядре, но ей хватает. Может я просто редко что-то новое устанавливаю, и не вижу, что что-то меняется. С моей колокольни grand strategy всё так же плохо с префомансом, настолько, что играть лейт в стелларисе на гигантской карте просто невозможно.

EvilFox 21 сентября 2018 в 18:32

0

> современные игры

Увы не всегда имеют современные движки. Но не стоит на них замыкаться выдавая это за всю картину в индустрии.

Druu 22 сентября 2018 в 15:28

0

Дело не в движке. Просто не все задачи параллелятся.

EvilFox 22 сентября 2018 в 19:03

0

В движке в том числе. В хороших движках многие подсистемы выносятся в отдельные потоки, и игровую логику при желании можно раскинуть по ядрам (ясное дело что не всё параллелится, но хватает задач где можно раскидать нагрузку).

Druu 23 сентября 2018 в 15:01

0

> и игровую логику при желании можно раскинуть по ядрам (ясное дело что не всё параллелится, но хватает задач где можно раскидать нагрузку)

Ну вот если вы в какую-то стратегию играете, то нельзя, потому что каждый юнит и каждое действие должно быть отработано последовательно.
Можно, конечно, пытаться разделить все по так скажем "причинной связности", но в итоге вы получите что разделение такое будет работать только в начале игры (когда и так тормозить нечему) а ближе к концу (где как раз и начинаются тормоза из-за просчета большого количества объектов) все будет связно.



0xd34df00d 24 сентября 2018 в 07:08

0

Как же ж физики задачи многих тел всякие параллелят? :)

Druu 24 сентября 2018 в 09:27

0

> Как же ж физики задачи многих тел всякие параллелят? :)

Именно вышеописанным способом (разделяя по причинной связности) и параллелят :)
Только в физике это легко удается, потому что происходит куча процессов, связями между которых можно пренебречь. А в игре — нельзя, потому что игровое моделирование обязано быть идеально точным. Пренебрегать нечем.

IRainman 22 сентября 2018 в 12:47

0

Дополню @Fracta1L, не только многопоточны, но ещё и на GPGPU.
P.S. пул потоков доступен каждому и много где из коробки, разные asyncи и прочее тоже.

nicholas_k 20 сентября 2018 в 17:22

0

Компьютерные программы — это не вещь в себе и не самоценность. Ценность программы определяет пользователь.

У каждой программы есть две стороны — ресурсы, которые она потребляет для написания и функционирования и польза(выраженная деньгами?), которую она приносит.

Раз уж сложилась такая картина, как описывает автор, значит ресурсы, потребляемые современными программами соответствуют приносимой пользе.

VolCh 23 сентября 2018 в 08:32

0

Пользователь определяет ценность только в идеальном мире где он может за одни и те же деньги установить любую версию любой программы и они не будут конфликтовать. Но если пользователю нужно запустить программу, которая работает только на Windows 10, то как бы ему не хотелось запустить её на XP, то будет вынужден показать всему миру, что Windows 10 полезная штука, как бы не плевался от неё. А даже если программа под XP работает, то попробуй эту XP легально установить в 2018-м году.

khim 23 сентября 2018 в 17:47

0

Если нужна старая программа, то как раз проблем с XP не будет. У меня есть знакомый, который на XP перешёл в 2010м году. С Windows 98. В прошлом году он начал задумываться об обновлении…

VolCh 24 сентября 2018 в 17:15

0

Ссылочку кините, где сейчас можно купить XP? или я пропустил и MS старые версии раздаёт бесплатно?



PsyHaSTe 24 сентября 2018 в 17:34

0

Можно попробовать многострадальный реактос. Как мне кажется, сейчас он должен быть на уровне первых сервис паков XP или лучше.

sumanai 24 сентября 2018 в 18:08

+1

Вам, увы, кажется.

sumanai 24 сентября 2018 в 18:08

0

Даунгрейд.

nicholas_k 23 сентября 2018 в 20:24

+1

Ценность не равняется цене.

Пользователь определяет ценность программы в реальном мире и делает это путем выбора — пользоваться или не пользоваться.

Раз пользуется, значит польза перевешивает недостатки. Иначе — не пользуется.

PashaNedved 24 сентября 2018 в 12:27

+1

А я курильщик, недостатки курения значительно перевешивают достоинства. Но я продолжаю курить.
Да и в общем, перспектива определять ценность по принципу: хочешь ешь, не хочешь — не ешь, выглядит не привлекательно.

nicholas_k 24 сентября 2018 в 13:52

0

Пример с курением не очень подходящий, так как свобода воли тут под вопросом.

Но в любом случае интересно, какие системы определения ценности, кроме субъективной свободы воли более перспективны?

VolCh 24 сентября 2018 в 17:18

0

Ценность для других субъектов, от которых наш субъект зависит и вынужден пользоваться согласно их воли вплоть до угрозы лишения свободы, а то и жизни.

nicholas_k 24 сентября 2018 в 18:59

0

Я полагаю, что если использование плохой программы позволяет человеку сохранить его жизнь, то это очень ценная программа для него.

PashaNedved 24 сентября 2018 в 17:23

0

Я без понятия какие, но только не заведомо проигрышную для меня…
Вы предлагаете делать выбор пользоваться или не пользоваться, а я вижу еще варианты — платить и пользоваться и не платить и пользоваться. Могу предложить еще десяток гипотетических вариантов выбора.

nicholas_k 24 сентября 2018 в 18:57

0

«Платить или не платить» — относится к ресурсам, которые программа потребляет для функционирования.

Очевидно, что ценность программы, за которую не надо платить будет выше и пользователь несомненно выберет такой вариант.

Но это не отменяет дихотомии.

VolCh 24 сентября 2018 в 17:16

0

А если это не польза программы перевешивает, а приказ руководства? Может без всяких программ он вообще в два раза больше бы делал, но начальство это не понимает.

nicholas_k 24 сентября 2018 в 19:04

+1

Если человек использует программу по приказу начальства под страхом увольнения, то к пользе программы надо добавлять ту пользу, которую приносит наличие у него работы.

Ведь работу тоже можно выбирать, а значит плюшки, которые дает работа, перевешивают недостатки программы, которой приходится пользоваться.

VolCh 24 сентября 2018 в 19:40

0

Перевешивают плюшки работы, а не программы. У программы может не быть никакой пользы, ну например CRM система, в которую все вбивают данные по приказы и всё. Никто эти данные не читает даже.

nicholas_k 24 сентября 2018 в 20:47

0

В данном случае программа — это и средство получения плюшек от работы в том числе.

VolCh 25 сентября 2018 в 09:53

−1

Пользу в данном случае приносит не программа, а исполнение блажи начальства. Захочет оно, что бы косынку раскладывали — будут косынку раскладывать, от программы ничего не зависит.



serbod 20 сентября 2018 в 17:48

0

Если дать разработчику не топовое железо и ежедневные апдейты, а Athlon без SSE и WinXP без сервис-паков (без поддержки EncodePointer/DecodePointer) и гигабайт памяти, то требования к готовому продукту резко снизятся. По себе знаю. =)



PsyHaSTe 21 сентября 2018 в 01:11

0

Да, но сроки увеличатся в несколько раз, частично из-за замедления общего цикла «написание-отладка-тестирование», частично из-за траты времени на собственно оптимизацию. Хотя мне кажется, что разработчик который только что мучительно смог все же запустить идеешку и прогнать тесты не будет сильно мотивирован еще и оптимизировать.



serbod 21 сентября 2018 в 10:14

+1

На самом деле нет, вместо оптимизации будет более тщательный выбор инструментов, технологий и компонентов, поиск узких мест в профайлере, стресс-тесты. Впрочем, онлайн-разработчикам это нафиг не нужно. Это проблема старого геймдева (картриджи, диски) и хардкорной инженерии (сильно ограничены ресурсы, связь, физический доступ).



PsyHaSTe 21 сентября 2018 в 13:08

0

Выбор инструментов — это когда вместо идеешки мы работаем в виме? Ну так это как раз и выльется в увеличение сроков реализации, человек при прочих равных выбирает IDE не просто так.



serbod 21 сентября 2018 в 13:49

0

Даже не буду убеждать, что в CLI можно работать быстрее и продуктивнее, чем в GUI, все равно никто не поверит. Просто приведу в пример те же самые IDE, только 2000-2005 годов.



PsyHaSTe 21 сентября 2018 в 15:44

0

Это старый холивар, и я действительно не поверю. По крайней мере для меня это не так, и если бы меня заставили работать в таких условиях, я скорее всего не смог бы, и вы бы наблюдали как раз те эффекты, описанные выше. В итоге вместо продуктивного взаимомодействия полагаю в течение полугода мы бы разошлись, крайней недовольные друг другом.

0xd34df00d 22 сентября 2018 в 00:51

0

Можно. Только не надо отождествлять IDE и GUI, особенно если там поддержка вим-кейбиндингов есть.

yefrem 20 сентября 2018 в 17:52

+2

Некорректно сравнивать штучные нишевые продукты прекрасного прошлого с унылыми массовыми из грустного настоящего. Сегодня любая булочная может иметь онлайн-магазин, а приложение для мобильного можно склепать на коленке и поделиться с тысячами юзеров не вставая с дивана. И это хорошо.

yarric 20 сентября 2018 в 23:47

+2

Тут уже раз пять упомянули Delphi, который был существенно проще в установке и использовании современных зоопарков из новомодный фреймворк+новомодный редактор+новомодный компилятор и т.д.



0xf0a00 21 сентября 2018 в 09:46

+2

В последнее время он тоже что то разожрался, но возможно из-за поддержки практических всех платформ. Впрочем пользоваться им все так же легко и приятно, и ide построена так что программисту с любой квалификацией будет одинаково удобно.



PsyHaSTe 21 сентября 2018 в 13:09

0

А на чем кроме дельфи вы писали? Ну, просто любопытно, с чем сравниваете. Имеется ввиду что-то больше, чем лабу, какого-нибудь бота или что-то подобное..



0xf0a00 21 сентября 2018 в 15:41

+1

На визуал студио и эклипсе.



PsyHaSTe 21 сентября 2018 в 15:45

+1

Я про языки программирования. Джава там. Котлин. Плюсы. Шарп. Питон. Руби. Ну, что-то мейнстримное или может быть хаскель какой-нибудь?



0xf0a00 21 сентября 2018 в 16:07

+1

И вот опять, в сравнении инструментов вы пытаетесь сравнивать языки. Учитывая ваши комментарии выше, видимо устраивать срачи ваша цель. Вопрос компетентности все так же остается открытым, как бы вы там не изгалялись в словоблудии. Адиос :)

PsyHaSTe 21 сентября 2018 в 18:31

0

Ну понятно. Программист на фортрнне может писать на фортране на любом языке программирования.

yshurik 20 сентября 2018 в 18:03

+1

А что я как-то "вернулся" ;) Моя inferno os загружалась за одну секунду на raspberry pi. Включаешь и пока экран загорается система уже загрузилась в гуи и готова. Класс ;) Только софта нет ;))))

Gryphon88 20 сентября 2018 в 18:17

+1

Примерно так же я пробовал КолибриОС: менее 5 секунд загрузки и определилось всё оборудование. Дистрибутив — менее 2 МБ на всё. В первый раз за 10 лет заплакал.
Семёрка грузится почти 3 минуты, и с драйверами я повоевал знатно…

springimport 20 сентября 2018 в 18:57

0

Если не считать долгую загрузку bios — это отдельный разговор, то win 10 на m2 ssd с 8700k грузится за 10 сек. Новая и почищенная загрузится за 5 сек.

Gryphon88 20 сентября 2018 в 21:05

0

Всё-таки не сдержусь: зачем по жизни нужен SSD? Только если под своп. Ну и загрузка не только биоса зависит от количества периферии.

springimport 20 сентября 2018 в 21:18

+2

Приятно когда все быстро работает.

minusnaminus 20 сентября 2018 в 22:01

+1

не путайте. Хорошо, когда быстро включается и долго работает, спросите любую женщину…
Да и любой айтишник тоже обрадуется

PsyHaSTe 20 сентября 2018 в 22:26

0

Чтобы всё быстро работало. Загрузка ОС — это ведь не все задачи. Например, на харде Europa universalis у меня грузилась несколько минут, с SSD — секунд 15. Разница существенная. Пример еще забавнее становится, когда оказывается, что игра немного странно сделана, и для загрузки сохранения она практически заново перезагружается. Кстати, а тут уже помогает префетч, и вместо 15 секунд грузится уже 10. А это один из тех «лишних» кусочков, которые по мнению автора всё усложнили «по сравнению с Windows 95».

khim 21 сентября 2018 в 01:04

+1

> А это один из тех «лишних» кусочков, которые по мнению автора всё усложнили «по сравнению с Windows 95».

Именно. Вначале мы всё делаем всё через… одно место, а потом начинаем со всех сторон пристраивать подпорки. И радуемся тому, что процесс, который, по хорошему, должен занимать меньше секунды занимает не пять минут, а всего лишь 10 секунд.

Почему загрузка должна занимать 10 секунд? Когда у вас уже все текстуры не просто в памяти, но в GPU и всё уже «на мази»? Потому что у кого-то руки неизвестно откуда растут? А выпрямлять не пробовали — вместо того, чтобы прикручивать 10 уровней костылей?

Статья, собственно, об этом… но она не доходит до сути — ответ-то прост: выпрямлять ничего не пробовали и пробовать не будем, там как это не прискорбно, денег.

CodeRush 21 сентября 2018 в 01:10

0

Истинно так, выпрямлять — невыгодно, т.е. делаться будет либо энтузиастами, либо только тогда, когда ничего другого уже не поможет, и останется только выпрямлять. Думаю, мы вполне успеем дожить до этого момента, если никаких существенных прорывов в скорости вычислений и объемах памяти не произойдет в ближайшие лет 30.

PsyHaSTe 21 сентября 2018 в 01:17

+1

> Почему загрузка должна занимать 10 секунд?

Потому что IO. Потому что в игре десятки тысяч файлов ресурсов (я лично видел полосу загрузки до 5600 ресурсов, мб их больше), ну и так далее.

> Когда у вас уже все текстуры не просто в памяти, но в GPU и всё уже «на мази»?

А как оно туда попало? Игру закрыли, всё выгрузилось. Мы же уже выяснили, что префетчи это все костыли через одно место?..

> Статья, собственно, об этом… но она не доходит до сути — ответ-то прост: выпрямлять ничего не пробовали и пробовать не будем, там как это не прискорбно, денег.

Деньги — мерило затрат человечества, если говорить абстрактно. Тратить больше денег — значит разбазаривать чьё-то время, который это будет оплачивать. Так делать нехорошо.

Ну и это beside the point. Почему SSD? Потому что ЛЮБАЯ операция с IO работает в 2..100 раз быстрее. И это очень сильно ощущается в любых операциях, даже если комп как у меня не выключается месяцами, и всё сидит в кэшах.

khim 21 сентября 2018 в 04:38

0

> А как оно туда попало? Игру закрыли, всё выгрузилось.

А почему игру закрыли? Я говорю про загрузку изнутри игры. Почему этот-то вариант должен тормозить?

Druu 21 сентября 2018 в 12:35

+1

Так разработчики реализовали.

khim 21 сентября 2018 в 22:18

0

Ну дык о том и речь: разрабочики всё реализовали через одно место, из «гавна и палок», а теперь эту кучу… добра обматывают толстым-претолстым слоем скотча, чтобы оно не так сильно пахло.

Druu 22 сентября 2018 в 05:20

+3

> Ну дык о том и речь: разрабочики всё реализовали через одно место, из «гавна и палок», а теперь эту кучу… добра обматывают толстым-претолстым слоем скотча, чтобы оно не так сильно пахло.

Если мне бы предоставили выбор — либо получить игру с дополнительными реализованными механиками, либо чтобы она грузилась 1 секунду вместо 10, то я бы выбрал ту, что грузится 10 секунд, но при этом есть дополнительные механики.
И это хорошо, что разработчики EU понимают вот эти мои желания.

У них если что движок в СК и EY по сути один и тот же а вся игра — это тонны скриптухи. Естественно, оно тормозит, но мне пофиг — потому что такой подход позволяет парадоксам регулярно выкатывать обновы, добавляющие целые игровые механики, и в итоге предоставлять мне уникальный продукт, который не имеет сколько-нибудь близких аналогов. И за который я готов платить деньги.

Очень удачно, что в парадоксах работают настоящие инженеры, которые решают задачу пользователя, а не рассуждают о том, что их ПО работает со скоростью 1% от потенциально возможной.

Gryphon88 21 сентября 2018 в 16:18

0

Тут, имхо, тоже вопрос «в руках». Про запуск вопросов нет: нам надо считать с диска и распихать в память. А что потом, где можно налажать:
— Всё нужное уже в памяти, но что-то постоянно не догружается, лезем за ним
— В память не влезает, поэтому постоянно сбрасываем на диск
— Косо расположили ресурсы: нелокальность файлов на диске, недостаточная гранулярность, не сделан префетч.
В моём идеальном мире за такие случаи должна отвечать ОС и ФС, но пока что, если мне требуется обработать несколько тысяч 5МБ файлов, я открываю O&O Defrag, чтобы убедиться, что файлы расположены в одном или соседних кластерах (чтение/запись одного куска значительно дешевле чтения многих мелких), и снизить вероятность того, что я упрусь в дисковое IO.

BD9 22 сентября 2018 в 04:37

0

До сих пор вспоминаю Transport Tycoon.
Под MSDOS, с множеством окон, создаваемых и перемещаемых пользователем, и в каждом окне наблюдаешь передвижение своего авотбуса/машины/самолёта…
На 2 дискеты влезало.

springimport 24 сентября 2018 в 17:50

0

Вы правы.
Но мне важнее правоты то что тяжелый проект на 50к файлов в phpstorm открывается на 8700k 5.0 16gb 970 pro m2 за 10 секунд, индексируется за 30. А на hdd это можно умножать в 5 раз.

atamanenko 20 сентября 2018 в 21:27

0

6600k, дешманский SATA3 Kingston, винде больше года. Около пяти секунд от нажатия на кнопку до экрана входа

springimport 20 сентября 2018 в 21:46

0

Отлично. До рабочего стола или экрана входа?

khim 21 сентября 2018 в 07:33

0

От экрана входа до рабочего стола на SSD под Windows 10 где-то секунда. Это не Windows XP на BigFoot'е где от момента появления рабочего стола до момента, когда компьютер соизволит начать реагировать на мышь и клавиатуру секунд 20-30 проходило…

da-nie 20 сентября 2018 в 20:30

0

Попробуйте ещё qnx 4 с демонстрационной дискеты. :) Вам понравится, гарантирую.

Gryphon88 20 сентября 2018 в 21:00

0

Пробовал, радовался, но там с прикладными программами ещё грустнее.

ValdikSS 21 сентября 2018 в 02:30

0

Попробуйте Damn Small Linux.

maikus 20 сентября 2018 в 18:04

+1

Наверное, никто сейчас не знает, что за код на самом деле день за днём выполняется на миллиардах устройств во всём мире. Кто даст гарантию, что там не на 80% Чужой код, и лет через -дцать, когда 5G или 6G случится везде, а устройства станут в n-раз мощнее, нас не ждёт здакая Массачусотская машина из смартфонов и ферм, с глазами и ушами в каждом закоулке, и которую нельзя будет обесточить и обнести колючей проволокой.

Scf 20 сентября 2018 в 18:05

+1

В большинстве инженерных областей изготовление изделия (или стоимость отказа изделия) значительно дороже его проектирования. В таких условиях имеет смысл максимально оптимизировать производство автомобилей, т.к. каждая сэкономленная гайка умножается на кол-во произведенных авто. Аналогично с архитектурой — построить дом куда дороже, чем спроектировать.

В IT все наоборот — разработка "по аджайлу" относительно дешевая, а вот проектирование и верификация — очень дороги. Как в пословице про дурака, реку и камень — один заходил, десятеро протестировать не могут.

Об оптимизации беспокоятся только те, кто сам платит за железо: прежде всего производители устройств на микроконтроллерах и владельцы облаков из сотен и тысяч машин.

Jeka178RUS 20 сентября 2018 в 18:12

+7

Автор очень точно прошелся по индустрии сайтов и мобильных приложений, вот там как раз самый ад и твориться. Потому что надо всем, быстро и дешево вот и результат. В то же время автор правильно заметил, что игры в разы быстрее работают. Вот тут и работают «правильные» программисты, правда я с опаской смотрю на Unity и подобные вещи и ожидаю, что скоро апокалипсис придет и в эту индустрию.

Emulyator 20 сентября 2018 в 23:04

+1

Достаточно заглянуть в гуглплей, чтобы убедиться, что апокалипсис уже давно там.)
100500 новых игр в день, большинство из которых по просмотрам и монетизации не намного превзойдут фотку кота тети Зины в одноклассниках. А туторы на ютубе по созданию игры на юнити намекают, что время разработки этих игр остро конкурирует со временем создания фотки вышеупомянутого кота.
Все меньше фирм готовы вкладываться в серьезный игровой продукт, который может не выстрелить.

prefrontalCortex 20 сентября 2018 в 18:19

0

Мне кажется, что такое положение дел в целом — это обратная сторона нынче популярного в узких кругах мнения, что, мол, хороший инженер должен думать в первую очередь о бизнесе, о его задачах и его нуждах. Но позвольте, — хочется мне возразить таким людям, — в стране, которая когда-то славилась одной из самых мощных инженерных школ в мире (это я про СССР, если что), за занятие бизнесом сажали за решётку, как у вас такие факты в голове укладываются? Инженер, блин, должен думать об инженерных системах, о бизнесе пусть голова болит у бизнесменов.

Losted 20 сентября 2018 в 20:41

+2

Софт и инженеры не существуют ж в вакууме. Любое решение должно соблюдать баланс между затратами и доходами. Условно идеальный софт, который не сделан вовремя и при этом еще дороже ожидаемой выгоды будет ужасным техническим решением.

PsyHaSTe 20 сентября 2018 в 22:33

+2

Инженер получает ЗП за решение задач, а не за то, что он делает крутые штуки. Всем нравится их делать, только никто почему-то ответственности не несет. А то легко сказать «давайте заморозим разработку на 3 месяца, отрефакторим Х», за это время конкуренты обошли, новый раунд инвестиций не дали, компания закрылась… Но с пустым кошельком остался только владелец бизнеса, а вы через пару недель пойдете к тем же конкурентам, которые остались на плаву. Немного нечестно получается, не находите?

Выше я уже написал. Ваша задача — дать квалифицированную экспертизу со всеми возможными рисками, возможно дать рекомендацию, но не принимать решения, если только это не ваша прямая обязанность по трудовому договору. Но тогда вы должны быть CTO (если не CEO), и мы наверное не про это говорим.

В конце концов, всегда можете показать, как надо сделать. Вот берете, делаете на гитхабе репозиторий такого же проекта, как у вас на работе (в той же сфере). Нанимаете людей, начинаете работать. Безо всяких нахлебников, которые ничего не понимают в продуктах. Только инвестиций брать нельзя, т.к. инвесторы не будут сидеть и ждать, пока их деньги вылетят в трубу. И продажников наверное придется нанять, иначе самый крутой продукт так и останется неузнанным и не скачаным, потому что про него просто никто не слышал. И много других но, с которыми постоянно взаимодействуют люди из бизнеса, но от которых мы с вами защищены джирой. Хотите принимать решения — берите риски на себя, а то принимать решения за которые потом будет расплачиваться владелец компании довольно подло.

VolCh 23 сентября 2018 в 08:42

0

Просто бизнес был государственным и государство сажало конкурентов, вознаграждая тех, кто хорошо решал его бизнес-задачи.

bosom 20 сентября 2018 в 18:29

+7

За наличие букв Ё в статье, автору статьи (переводчику), огромный респект и уважение!
Мало, очень мало, людей не только код стали писать плохо, но и русский язык не знают!

novice2001 21 сентября 2018 в 10:10

–3

Вы — один из тех людей, кто русский язык не знает…
Ваше второе предложение «Мало, очень мало, людей не только код стали писать плохо, но и русский язык не знают!» — это шедевр в духе киевского мэра.

hellowitty 21 сентября 2018 в 13:40

0

Респект, ё!

Corosan 21 сентября 2018 в 16:48

+2

Да ладно с этой Ё, тут ещё прокатывает вариант в стиле "да у меня клавиатура из 56 букв, у меня нет такой клавиши". Но как же сильно выматывает неправильное использование мягкого знака после "т".

Уважаемые читатели и все-все-все. Пожалуйста. Запомните, те, кто ещё не в курсе после школы. Нужно *ли* ставить "ь" в глаголах после "т", проверяется вопросами "что делают/ет?", "что делать?". Если в вопросе есть "ь", то нужно ставить. Пример:

- "Многие люди не напрягаются в использовании мягкого знака" (Что делают? ЮТСЯ — без "ь".)
- "А пора бы уже напрягаться" (Что делать? АТЬСЯ — с "ь".)

hfinn 21 сентября 2018 в 17:59

0

> Тема сложная. Раньше для объяснения уже высшие силы привлекали.

rulebook 1 октября 2018 в 11:09

0

То есть, если человек напишет «матерый» или «потемки», то он уже русский язык не знает? И еще давайте учитывать, что у большинства людей на клавиатуре ё находится отдельно от остальных букв, так что сидеть и специально печатать «ещё» вместо «еще» попахивает педантством.

sumanai 1 октября 2018 в 14:24

+1

> И еще давайте учитывать, что у большинства людей на клавиатуре ё находится отдельно от остальных букв

Не так уж и далеко, не в соседнюю комнату сходить.

Emulyator 20 сентября 2018 в 18:48

+3

А какой ситуации можно ожидать, если ПО возникает, меняется, умирает по законом очень похожим на эволюционные? Да, кто-то скажет, что нет в мире ПО нормальной конкуренции и настоящего выживания наиболее приспособленных, что бабло и маркетинг исказили законы и побеждают не самые быстрые, компактные, удобные, надежные, а блестящие, мигающие, часто обновляющие внешний вид, разрекламированные.
Печаль в том, что эволюционный подход хоть и прекрасен, но он вовсе не обязан выдавать эффективное во всех отношениях решение, а главное условия, к которым вынуждены приспосабливаться программы (их разработчики) сильно изменились. Изменились не столько из-за жадных бизнесменов, хитрых маркетологов и глупых дизайнеров, а по причине широкого распространения компьютеров, мобил и прочих умных устройств среди населения. ПО, превратилось в пакет с соком на полке супермаркета и на него стали действовать отработанные десятилетиями методы эффективной продажи (заодно придумали несколько новых специфичных для софта). Да, маркетологи виноваты, но они вторичны. Первичен человек, который, как и 100 лет назад, любит «новое», привлекающее внимание, с функцией «потешить ЧСВ» и т.п.
Так что ничего существенного мы тут не изменим, разве что всем возмущенным скинуться и оплатить масштабную рекламную компанию с тезисами, продвигающими эффективность, функциональность, реальную надежность, олдскульность и т.п. Но это все временно, и против любви к «новому» мало что поможет, она «прошита» в людях аппаратно.)



BD9 22 сентября 2018 в 04:48

0

Живая природа тоже любит выкрутасы — посмотрите на павлинов. Или на возвратный гортанный нерв.



Wesha 22 сентября 2018 в 07:37

0

Я просто оставлю это здесь.

Emulyator 22 сентября 2018 в 17:01

0

Это то и настораживает. Когда наблюдаешь «выкрутасы» в отдельных программных продуктах, поневоле задумываешься об отсутствии разумного создателя.)



gatoazul 20 сентября 2018 в 18:52

0

Ничего, скорость процессоров растет все медленнее и в конце концов упрется в потолок. Вот тогда и начнется волна оптимизации софта.



maydjin 20 сентября 2018 в 19:01

0

Дык иначе надо мощности останавливать и огромные штаты людей выгонять на улицу. А зачем останавливать, когда можно не останавливать а выхлоп продавать и обогащаться?

На самом деле, есть определённый круг людей, который тусуется на железе 10+лет давности или на low end актуальном железе, что бы хотя бы для себя снизить такие неэффективности.

Ещё, есть системная разработка, где нервы автора статьи наверное могли бы слегка поуспокоиться :)

InterceptorTSK 20 сентября 2018 в 19:03

−1

Когда-то программирование было точной наукой. И никак иначе. 2+2 всегда было равно четырем. Нынешнее программирование совсем убого. Если 2+2 выдает 5 — то это считается не таким уж и плохим результатом. Если 2+2 выдает 4.5 — то это вообще хорошо! Если 2+2 выдает 4.1 — то вы бог программирования!
Причем нынешние дегенераты реально думают так, и живут именно в таких понятиях. Они же не могут понять простой вещи, что все ошибочное — ОНО ОШИБОЧНО, и не важно насколько близка ошибка к точному результату. Нет же. Вместо того что бы избегать ошибок вовсе, дегенераты придумали градации ошибок, и термин ошибка сменили на другой. И все сразу стало в шоколаде) Ошибки заменились на ворох каких-то никчомных дегенерацких терминов. И это спасает дегенератов от выпила из системы. Ну а кому охота идти с метлой к помойке?
И да, основная проблема только в том, что программирование стало доступно для широких масс, читай — для быдлы. В этом основная проблема нынешнего «программирования».
Уже лет примерно 5-8 назад, точно не скажу т.к. это не существенно — я перестал именоваться программистом, когда меня спрашивали о роде деятельности. Примерно в оговоренное время я понял, что лучше мне не называться программистом вовсе, т.к. к ТАКОМУ ИДИОТСКОМУ ПРОГРАММИРОВАНИЮ я не имею никакого отношения вообще.



da-nie 20 сентября 2018 в 20:35

0

> стало доступно для широких масс

Нифига себе, для широких масс... Я как учебник по Си++ ни открою, так просто офигеваю от его сложности. А как открою «Эффективный Си++», так и вовсе в простации от концепций. Это что *ли*, низкий порог вхождения? ;)

InterceptorTSK 20 сентября 2018 в 21:27

+7

Так точно. Так и было. Закрывали и дасвидули.
Вы открыли ровно то, что юзалось как раз таки 20 лет назад повсеместно, потому что ничего другого не было. Ну был паскаль, ну фортран. Что то там еще. Это не совсем то.

Сейчас же вы можете вмиг наколхозить на более чем 20-и языках свою поделку и не париться. Вот смотрите как оно было с той же мобильной джавой. Вам шло где-то 15 базовых классов, типа систем, эрей, дисплей, вектор и т.д. И НИ ОДНОГО ЗАРАНЕЕ НАПИЛЕННОГО КОМПОНЕНТА) Пишите сами. Ищите сами, например что-то похожее ищите, перепиливайте под себя и юзайте. Мобильная джава в себе не содержала фреймворка. Напишите его сами. Сможете? Как напишете — так оно и полетит. И раньше такое программирование было повсеместным. И оно приводило к тому, что людей заставляли понимать, что они делают.

Сейчас же базовые вещи раздали всем. И оно позволяет программировать как бы без проблем. Т.е. программа-то запускается, и даже поведение адекватное. Но человек безумно собравший из кусков свое поделие вообще не понимает как оно работает. И отсюда все беды конечно же.

Ну вот вам пример. Я на днях увидел как некое тело пишет реализацию ISO639, ну там кому-то надо было что бы все по стандарту и т.д., не суть, пришлось прикрутить. Это стандарт языков если что, в смысле языков на которых люди общаются. Ну я поглядел как оно. Это же бред какой-то) Во-первых зачем-то прикручена база данных. Накачаны какие-то компоненты, какое-то гавно. Ну просто фееричное гавнище абсолютно не нужное никому кроме дегенератов. Ну типа что бы база работала и там что-то еще. Я вообще не понимаю зачем база данных, если база статическая))) Тут любой массив обычный подойдет. МАССИВ))) ОБЫЧНЫЙ. Правда не совсем обычный, а разряженный, ну не суть. И дергайте вы с него все что захотите и как хотите. Вот значит его тело начало писать какие-то безумные скрипты, что бы наколхозить сначала базу данных, причем накачало каких-то пакетов, затем какие-то проверочные скрипты, какой-то ужасающе-зверский навоз.
Ну есть готовая база на сайте ISO, но это же не значит, что ею надо пользоваться напрямую)) Это же не эффективно.
Вот пришлось переписать это на уже обычнейший шарп на структурах, и оно зверско полетело в сравнении с поганым мсюзлем убогим. Получилась обычная либа, с массивами внутри. И функциями поиска в них. На 3570к например из либы можно дергать поиском 200 миллионов языков в секунду. Такая вот производительность. Можно даже чуть быстрее, но пока не нужно. Мсюзль и рядом не валялся, зачем он? Ничего сложного) Никаких зависимостей, никакой фигни. Зачем оно? Собрали на втором фреймворке, подписали и работать оно будет всегда и вечно. И память вообще не ест, с чего? И обновлять это никто не будет))) ЗАЧЕМ??
И подобных либ у нас уже сотня. Ну нет смысла юзать что-то в чем вы вообще ничего не понимаете. Пишите все что есть сами. С нуля. Только это позволит вам действительно разобраться. И так и писали раньше повсеместно.

Делов на пол-дня, зато у нас появилась очередная библиотека, которую в след.раз мы примотаем за 2 клика хоть куда, и она железобетонно будет работать.

Так щас же писать разучились. Ручками писать. Накачают какого-то гавна, которое называют пакетами, компонентами и прочей парашей — скрутят в кучу и смотрят как оно падает зловонно воняя на весь комп абсолютно неадекватной хренью, выжирая память проц, и в итоге подыхает в конвульсиях, захлебываясь в фееричном количестве собственных ошибок и исключений.

VezhLos 20 сентября 2018 в 23:38

0

Плюсую всем руками и ногами. Всё в точку. Последний абзац ну просто с языка сняли. Эх, Ваш бы коммент да распространить повсеместно…

PsyHaSTe 21 сентября 2018 в 01:26

+1

Могу ответить одной ссылкой: habr.com/post/280099/#comment_8820183

Нет ничего хорошего в том, чтобы велосипедить свою реализацию если можно взять готовый компонент. Обратное явление называется «копи-пастой» и «велосипедостроением» и не особо практикуется.

А то я так могу реверснуть ваш пример: вы написали супер-оптимизированную офигенную реализацию, а я взял и переписал на использования nuget-пакеты, которые *еще* на 20% производительнее, и в котором починена куча всяких неприятных уязвимостей и багов.

Жду не дождусь какого-нибудь всеобъемлющего npm/nuget/... репозитория, где можно будет хранить пакеты в виде какого-нибудь LLVM IR, который можно будет подключать к проекту на любом языке. И убрать вообще всё дублирование, как это сейчас бывает, даже если есть хорошая либа для X на одном языке, не факт что она будет на таком же, но похожем.

InterceptorTSK 21 сентября 2018 в 08:31

0

Проблема в том, что готовых компонентов вагон и телега. И далеко не все документируются. И совсем единицы подвергаются тестированию. И уж совсем один или ажс(!) два имеют спид-тесты. Вы о чем говорите вообще?) Т.е. вы хотите, что бы я реально взял чье-то поделие, и прикрутил себе? ВЫ В СВОЕМ УМЕ ВООБЩЕ?)))) Я понятия не имею как оно работает. Зачем оно мне? Я же это людям выдам, и я за это отвечаю, а не пейсатель пОкета. И как мне тестить таких же аналогичных пОкетов десяток? А если их 20? А если у них разная реализация? Что сложного прикрутить несколько массивов? Ну да, надо знать что такое массив, а как иначе? Надо знать как эффективно в нем искать. Надо знать как собрать структуру, что бы она была пригодна для поиска и положить ее в массиве именно в то место, что бы поиск был за O(1). Вот тупо поиск мгновенный получается. Вообще моментальный. Вот прям поиск без поиска) И да, такое запросто возможно, и это никакая не заумь, ЭТО ПРИМИТИВЩИНА первого курса любого факультета связанного хотя бы как-то с программированием. Вы поймите, если вы вообще ничего не понимаете в этом, так вы не программист вовсе. Любой программист из бывших вообще выпучит глаза на все эти ваши пакеты, закроет рукой лицо и самовыпилится сразу же. Это же для дегенератов сделано. Ну т.е. не для программистов совершенно. Ну есть там вы или домохозяйка Люсия, вот для них оно сделано. Собрали как-то, запустили, скрепит-пердит, но работает. Правда падает через раз, но для домохозяек пойдет. Вы домохозяйка что ли? У вас вообще хотя бы минимальное самоуважение хотя бы на уровне студента первого курса есть? Ужасно…


0xd34df00d 22 сентября 2018 в 06:35

+2

Это хорошо, когда дело ограничивается, скажем, поиском по массиву (*а чего вы массивы сами не писали, кстати?*). А вот если вы мне предложите взять и самому написать, например, вычисление обратной матрицы, или топологическую сортировку, или свои комбинаторы для парсеров, или вообще какой-нибудь хулл, я начну немножко удивляться.


da-nie 22 сентября 2018 в 12:25

0

> Вы открыли ровно то, что юзалось как раз таки 20 лет назад повсеместно, потому что ничего другого не было. Ну был паскаль, ну фортран. Что то там еще. Это не совсем то.

В смысле, открыл? Я с 1995 года всем этим занимаюсь. Но это совершенно не значит, что я владею и понимаю приёмы того же Александреску или Мейерса, а также знаю и применяя все паттерны и современный стандарт Си++, а так же гору алгоритмов. Всё это совсем не просто. И никуда не делось.

> Сейчас же базовые вещи раздали всем. И оно позволяет программировать как бы без проблем. Т.е. программа-то запускается, и даже поведение адекватное. Но человек безумно собравший из кусков свое поделие вообще не понимает как оно работает. И отсюда все беды конечно же.

А вот о каких базовых вещах, розданных всем, вы пишете, я не понимаю. Это про web что ли? Ну, тут я полный ноль. В C#, python и java кстати, тоже.
А так, обычные приложения и 20 лет назад клепали в Builder и Delphi не особо вникая в работу системы. Но такие плохо понимающие разработчики писали для себя или мелких контор и в серьёзный софт попадали редко (хотя, QIP вроде как на чём-то подобном был написан, но, думаю, его автор хорошо понимал, как оно вообще работает).
Если честно, я даже не знаю, как сейчас пишутся приложения для PC. В чём, например, рисуют интерфейс? К чему он подключается? Я всегда использовал MFC и формы в ресурсах. А сейчас как?

Anthony_K 20 сентября 2018 в 19:33

+4

Обратная сторона такой деградации производительности — резко возросшая, невероятная скорость разработки нового ПО. Если на разработку Windows 2000 Микрософт потратила более 5 лет, то сейчас новая версия винды выходит чуть ли не каждые 2 года (про то, что всякой прикладной аплихухи за день сегодня пишется больше, чем писалось за год 15 лет назад умолчу. При чём, как в штуках софта, так и в строчках кода). Короче говоря, наблюдаем переход качества в количество.


dartraiden 21 сентября 2018 в 02:16

0

И скорость портирования этого ПО. Можно долго ругать Electron, но он позволяет быстро выпустить приложение под все основные платформы.

Контрпример: есть мультипротокольный мессенджер, который при 4 активных протоколах (ICQ, IRC, Tox, Jabber) потребляет всего 8 мегабайт оперативной памяти. Но это достигнуто за счёт максимального использования WinAPI. И как только появилось желание портировать его для Linux — упс, приходит осознание масштаба работ и понимание того, что в 2018 году разница между 8 и 80 мегабайтами ощущается совсем не так, как в 2000, когда оно писалось.


Massacre 21 сентября 2018 в 03:36

0

Вот там выше в комментах (volt.ws) пробуют написать аналог сразу с поддержкой win, macos, linux. Пожелаем им успехов в развитии :)

DrPass 21 сентября 2018 в 02:47

+4

> Обратная сторона такой деградации производительности — резко возросшая, невероятная скорость разработки нового ПО

Я честно, кроме редких исключений, вижу строго обратную картину. Изменения внедряются больно и мучительно, новых фич мало, чаще тратится время на перерисовывание картинок, нежели на функционал.
Да даже ваш пример с Windows 2000. Да, на разработку NT5 потрачено пять лет. Но количество новых функций там было беспрецедентным. Серьезные изменения в файловой системе, новая архитектура драйверов, огромный набор инструментария для управления инфраструктурой, от консоли WMI и до Active Directory и т.д. Я не слишком ошибусь, если предположу, что за эти пять лет в Windows было реализовано больше нового функционала, чем за последующие 18 лет от Win 2000 до Win10.


Alex_ME 21 сентября 2018 в 03:22

+2

И зачем нам новая версия винды раз в два года? (Нынче вообще rolling update)

Anthony_K 21 сентября 2018 в 09:05

0

Богам маркетинга надо приносить жертвы, иначе они прогневаются. 90% пользователей ничего не понимают в технической стороне продукта, но хотят что бы было «модно и современно».


BD9 22 сентября 2018 в 04:55

0

Потому что начиная с Windows Vista ОС у M$ внутри меняются мало. До нынешних скоростей многие пинали M$ за задержки с выпуском Windows Vista, потом за тормоза и плохой интерфейс, и что не всё воплотили.

theTeacherOfEnglish 20 сентября 2018 в 19:33

+2

Возможно эта история прозвучит как оправдание, но все же немного объяснит плохую производительность некоторых приложений, и следовательно плохое качество их кода.

Когда я окончил учебу и получил степень бакалавра, мне нужно было искать работу, накопились расходы, долги. Меня захайхантили и предложили следующий проект: за 3 месяца нужно было написать обучающее веб-приложение с разными заданиями, множеством анимаций, и чтобы оно запускалось на Андройд телефонах и планшетах, на айфонах и айпадах.

Деньги были очень нужны, и я согласился. В команде были сценарист, аналитик и я в роли фронтэндера. Мы поставили это приложение, и на кордове переконвертировали это в мобильные приложение (иначе было просто не успеть). Приложение работало в браузере, но на телефонах и планшетах жутко тормозило. Наверняка кто-то пользуется этими приложениями и до сих пор злится и пишет подобного рода статьи.

Возможно виноват я (что взялся за это дело), возможно заказчики (что слишком жадный), а возможно рынок виноват. Но таких случаев очень много, и с этим ничего не поделаешь.

НЛО прилетело и опубликовало эту надпись здесь

Anthony_K 20 сентября 2018 в 20:03

+1

Опенсорс — такое же тормозное г-но, как и проприетарное. У последней убунты требования к системе уже выше, чем у винды.

НЛО прилетело и опубликовало эту надпись здесь

Xandrmoro 20 сентября 2018 в 22:14

+1

Я не смог заставить работать убунту на абсолютно типовом железе.
Вообще.

lxsmkv 20 сентября 2018 в 23:13

+1

Я пять лет назад страдал «дистросвитчингом», баловался линуксами, гонял их на виртуалбоксе. Месяц назад мне на работе понадобился линукс. И на последнем виртуалбоксе у меня из коробки не инсталлировалась ни убунта, ни федора. Я сильно удивился. Раньше все это ставилось на раз.

Anthony_K 20 сентября 2018 в 22:19

+2

Пишу из-под WinXP на AthlonXP. Тормозов нет вообще, старт (от появления лого до приглашения Ctrl+Alt+Del) 25 секунд, шаттауном не пользуюсь — хибернейт рулит. Но один частный случай, как вы понимаете, не показатель. Да, скайп из-под неё не работает ((

sumanai 20 сентября 2018 в 22:36

0

> Да, скайп из-под неё не работает ((

Вы что-то делаете не так, у моей матери работает.

Anthony_K 21 сентября 2018 в 00:44

0

А какая версия, если не секрет?

sumanai 21 сентября 2018 в 11:16

0

Спрошу на досуге.

sumanai 26 сентября 2018 в 19:59

0

7.36.66.150, если я правильно запомнил.

Grox 21 сентября 2018 в 21:42

0

В WinXP возможно, но не на Athlon XP, в прошлых версиях они добавили обязательных инструкций процессора, которых нет у Атлонов. Может быть в Скайп 8 всё переделали и оно заработало?

Vanellope 21 сентября 2018 в 15:10

0

Пишете в мобильной версии хабра или в обычной? На стареньком но боевом Core2Duo E4700 и 4х гигах оперативной памяти пользоваться хабром можно комфортно только в мобильной версии, а текущий тред с овер 800 комментов, боюсь отправит браузер к праотцам.

Massacre 22 сентября 2018 в 02:26

0

У меня E5300 и 4гб, всё нормально можно делать в этом треде. Это ж не Фейсбук.

BD9 22 сентября 2018 в 05:01

0

Под Win XP достали тормоза Огнелиса (Win XP 32 бит), и он перестал обновляться с лета. Линукс и FreeBSD (64 бит) работали быстрее на той же машине.

PsyHaSTe 21 сентября 2018 в 01:35

+2

Ставил убунту на ноут, в итоге на второй или третий день смог найти инструкцию, как завести вай-фай. Нужно всего лишь поставить пару компонентов, склонировать репу и сбилдить его. Для меня как разработчика было не сильно сложно. Но для рядового пользователя это дикость. Но вот после этого победить рандомные зависания я не смог. Пока отложил, дойдут руки, буду разбираться.

Так что такое себе.

> 1) капитализм, где зарабатывают не на прибылях, а на издержках

Какие альтернативы? Из конкурирующих фирм я помню только коммунизм, который представлял собой дикую смесь кумовства и распила: большинство смешных реальных историй того времени — как кто-то договорился о покупке мяса с черного хода, принес фальшивые деньги, а купил мясную отработку, кто как спирал дорогущие космические реагенты и продавал, кто платиновой ниткой себе шнурки завязывал, кто авиационной краской военного назначения за бешенные бабки красил себе туалет на даче... В общем, если это имелось ввиду под «альтернативой», спасибо, не надо.

НЛО прилетело и опубликовало эту надпись здесь

Sychuan 21 сентября 2018 в 12:13

0

Коммунизм был первобытнообщинный. Читайте классиков марксизма-ленинизма. А вообще вы уверены, что система, все попытки построить которую ведут к развалу и которой никто никогда не видел это круто? Может быть лучше «О Граде Божьем» Св. Августина почитать с такими же успехами?

НЛО прилетело и опубликовало эту надпись здесь

evtomax 21 сентября 2018 в 20:03

+2

В «то время» был переходный период от коммунизма к капитализму, ибо партноменклатура осознала себя как класс буржуазии. Коммунизм был с середины 30-х до начала 60-х. Социализм — это неразвитый коммунизм с отпечатками предыдущей формации.

Giryphon88 21 сентября 2018 в 21:29

+1

Офигеть, коммунизм с Вождём — это что-то новое в политологии.

evtomax 21 сентября 2018 в 19:58

0

Вы перечислили как раз ростки капитализма в коммунистическом обществе, а не признаки коммунизма.

PsyHaSTe 21 сентября 2018 в 20:02

+3

Я рассказываю про то, как это было в самой крупной попытке человечества реализовать социализм... Впрочем, я знаю, что вы на это ответите, поэтому просто оставлю еще вот это:



evtomax 21 сентября 2018 в 20:09

+1

А вот и не угадали! В СССР был настоящий социализм. При социализме проблемы старого общества не пропадают внезапно, и при отсутствии сознательной общественной борьбы с ними, старые порядки неизбежно возрождаются.

PsyHaSTe 21 сентября 2018 в 21:56

0

В таком случае лучше отказаться от бесплодных попыток, раз злые буржуа каждый раз никуда не деваются?

user343 21 сентября 2018 в 20:20

+1

А вы читали книгу "Конец техноутопии" (Overshoot, 2006) и повесть "Машина останавливается" (The Machine Stops, 1909)?

"Когда-то люди строили машины, чтоб сделать свою жизнь как можно более комфортной. Теперь они не могут и представить себе существование без всемогущей Машины, обеспечивающей их пищей, водой, воздухом, книгами, связью и вообще всем, что только можно себе представить. Мало кто задумывается над тем, что Машина полностью подчинила себе людей, а того, кто задумается, ждет неминуемая кара — лишение крова и изгнание на безжизненную поверхность Земли. Но однажды Машина ломается..."
(а измельчавшие людишки разучились её чинить)

PsyHaSTe 21 сентября 2018 в 22:02

+2

Нет, не читал, но посыл «давайте жить хуже, чтобы если что-то пойдет не так мы были подготовленными» мне не кажется верным.

user343 21 сентября 2018 в 22:18

–1

Уже пошло не так, замаскированный энергетический кризис на повестке дня. Мощные пылесосы и лампочки не просто так запретили. Рост скорости процессоров не успевает за отуплением программ. Гонки в цифровую пропасть заканчиваются.

"Новые техпроцессы для производства микросхем все чаще откладывают"
https://habr.com/company/it-grad/blog/422501/

"2019 — год, когда Intel остановился" https://habr.com/post/424035/

"В школах Франции запретили мобильные телефоны" https://rg.ru/2018/07/30/v-shkolah-francii-zapretili-mobilnye-telefony.html

PsyHaSTe 22 сентября 2018 в 03:24

+2

Уух… Покайтесь грешные, ибо грядёт Он. И воздаст Каждому по делам Его… А нет, это немного из другой оперы.

Эти сказки про мировой кризис, падение доллара, заполонение Европы средним сирийским классом, вендекапец и прочее уже не вызывают должного интереса. Три ссылки вроде как должны придать доказательной базы, но по сути просто похожи больше на случайные новости из ленты. Кроме интела есть другие компании, техпроцессы откладывают по очевидным причинам, и это было понятно всем, кроме маркетолухов. И ради бога, при чем тут французские школьники? Очень обидно за них, конечно, но как они влияют на замаскированный энергитический кризис (или они сами его и маскируют?) не совсем понятно.

user343 30 сентября 2018 в 22:02

0

> как они влияют на замаскированный энергитический кризис (или они сами его и маскируют?)

Устройства для передачи сжатых данных при малом трафике или без него потребляют энергии меньше, чем при полной нагрузке — новость для вас?

"Системы сотовой связи проектируются исходя из максимальной плотности абонентов, на основе которой определяется необходимая емкость базовых станций (число радиоблоков). Поэтому, если в зависимости от времени суток число активных абонентов претерпевает сильные изменения, емкость базовых станций в течение некоторых периодов времени не используется полностью. Системы динамического энергосбережения позволяют отключать неиспользуемые радиоблоки, что снижает суточное энергопотребление на 10–15% в зависимости от условий работы станции [24]. Наибольший эффект такие системы энергосбережения дают ночью, в то время как днем постоянная активность абонентов не позволяет активировать соответствующие функции."
—

Взято из источника [24] — Lubritto C. Telecommunication power system: energy saving, renewable sources and environmental monitoring // Trends in Telecommunications Technologies. — InTech, 2010. — P. 145–164. (статья https://naukovedenie.ru/PDF/30evn413.pdf)

Вторые и третьи признаки энергетич.голода:

Ютуб перестал отдавать HD видео, а google maps — 3D карты Земли нищебродам со старыми ОС и браузерами — с них рекламной конверсии меньше, чем с богатеньких инкубаторских пользователей распоследнего г-хрома с шпионских ОС и на новом "железе". Но электричество на повышенную загрузку дата-центров расходовалось зря при отдаче кому попало высокого качества.

Вот нахрен 2.5D картам Земли видеоускорители не ниже предпоследнего поколения у клиентов, если раньше и программно всё это прекрасно рендерилось и в скорость интернета упиралось, а не в объём видеопамяти (>512МБ сейчас требование) или скорость граф. чипа?

PsyHaSTe 3 октября 2018 в 00:08

0

Просто потому что старые браузеры поддерживать дорого, Это вам в голову не приходило?..

DrPass 3 октября 2018 в 15:45

0

Дорого, это если с нуля писать. А развивать уже существующую и работающую систему, не ломая совместимости, обычно отнюдь не дорого.

PsyHaSTe 3 октября 2018 в 16:53

+1

Ну вот сказал заказчик вам добавить не знаю, резиновую растягивающуюся хрень на сайт. И у вас вариант, взять флекс и забить на старые версии или костылить что-то на старой технологии. Вполне себе дорого получается.

user343 4 октября 2018 в 00:19

0

Достаточно было не ломать и не делать chrome-only веб-версию google earth.
Почему-то у ebay и aliexpress нет дурной привычки перетаскивать всех на новые браузеры со шпионскими функциями.

Гуглокарты прекрасно **работали** в своём полу-"3D" режиме на нижеперечисленных ОС и железе, нет там ничего сверхсложного. Но потом корпорация добра бабла стала отсекать лишнюю нагрузку — посетителей с недостаточно мощным железом, от которых рекламной прибыли мало.

"В следующих операционных системах Google Карты также работают, но режимы 3D и "Земля" недоступны:

> Windows XP и Vista.

3D-графику в Google Картах не поддерживают следующие видеокарты:

> Intel 965GM;
> Intel B43;
> Intel G41;
> Intel G45;
> Intel G965;
> Intel GMA 3600;
> Intel Mobile 4;
> Intel Mobile 45;
> Intel Mobile 965."

https://support.google.com/maps/answer/3096703?co=GENIE.Platform%3DDesktop&hl=ru

user343 2 октября 2018 в 23:37

–1

~~Нагадить в карму проще, чем признаться в своём "пиксельном мышлении"?~~

В РФ грядёт скрытое повышение тарифов на эл. энергию (потому что ВВП запретил повышать расходы на жильё сверх инфляции):
http://www.aif.ru/society/smi_v_rossii_vnov_obsuzhdayut_vvedenie_normy_energopotr

"Предлагается установить социальную норму в размере 300 кВт•ч в месяц на домохозяйство. При потреблении свыше 500 кВт•ч будет действовать «экономически обоснованный» тариф. По словам экспертов издания, большинство домохозяйств будет укладываться в этот показатель (средний уровень потребления сейчас составляет 220 кВт•ч в месяц. – Прим.). При этом список потребителей, приравненных к населению (сейчас это, например, садоводческие и огороднические товарищества, гаражные кооперативы, места лишения свободы и ряд других. – Прим.), предлагается сократить, и постепенно отменить льготные тарифы для селян и квартир с электроплитами и электроотоплением.
По данным издания, схему одобрил вице-премьер Дмитрий Козак. Министерства энергетики и экономики, а также Федеральная антимонопольная служба должны внести проекты соответствующих документов к 15 января 2019 года."

PsyHaSTe 3 октября 2018 в 00:11

0

Вам в карму я лично не гадил, я вообще минусы никогда не ставлю, разве только к комментариям. Либо плюс, либо игнор.

Причем тут тарифы на электроэнергию? Мне кажется у вас некоторое расстройство…

0xd34df00d 22 сентября 2018 в 06:36

0

Не надо судить об опенсорсе по убунте. *Гентушникам обидно.*

IRainman 22 сентября 2018 в 12:09

+1

Убунту из всего опенсорсного ОС как то более всего человеко ориентирована и для простых людей. ИМХО конечно.



san-smith 22 сентября 2018 в 19:40

0

Как по мне, Linux Mint будет поудобнее для начинающего линуксоида. Всё-таки убунтовская Unity вещь довольно специфическая.

Кстати, DistroWatch тоже считает, что Минт популярнее Убунты.

VolCh 23 сентября 2018 в 08:56

0

Юните почила в бозе :(Теперь гномЗ



san-smith 23 сентября 2018 в 09:20

0

Да? Буду знать. Просто я после Убунты 12.04 сначала пересел на Xubuntu, а потом на Mint.

IRainman 23 сентября 2018 в 13:48

0

Хм, интересно, возможно в следующий раз её попробую. Но сейчас, особенно с учётом изменения вектора развития Microsoft и выхода 10, пусть пока она всё ещё для меня сыровата и 7 лучше, на Linux для десктопа я вообще не смотрю, а в роутерах и серверах (по большей части) и так Linux но без оболочки.

НЛО прилетело и опубликовало эту надпись здесь

НЛО прилетело и опубликовало эту надпись здесь



PsyHaSTe 21 сентября 2018 в 01:36

0

Странные у вас представления о корпорациях… За каждой корпорацией сидит куча людей, которые постоянно считают деньги. И уж они то как раз за стабильность. А в опен-сорсе все такие «а давайте на юнити — давайте, там круто!!», а потом «а давайте обратно — давайте, тож ничо!».



0xd34df00d 22 сентября 2018 в 06:41

0

Не нужно ничего ни с кем согласовывать. В дебиане вам нужен просто проверенный опытом мейнтейнер, готовый взять вас под крыло на время. В генте вам даже это не нужно, открываете пулл-реквест у них на гитхабе и всё. Я так туда свой нужный полутора анонимусам комбайн запихнул.



0xd34df00d 22 сентября 2018 в 06:40

0

Мне вот интереснее сравнивать, скажем, clang с icc по части поддержки стандартов. Или с cl.exe, хотя они вроде как зашевелились последнее время.

Asxer 20 сентября 2018 в 19:43

+2

Я в таких случаях привожу аргумент «Твоё приложение может быть бесконечно оптимизированным, но какой в этом толк, если оно не работает?». А почему оно может не работать? Ну например потому что заказчик постоянно хочет фичи и меняет требования, а конкуренты поджимают. Ни один продукт не живёт в состоянии стазиса, он всегда развивается и именно развитие продукта является тем самым испытанием, которое диктует приоритет на том, чтоб брать стабильные, готовые решения даже если они медленнее, чем каждый раз писать своё и быстрое решение. Пользователь конечно заметит, что приложение грузится чуть медленнее, но когда вы начнёте путаться в тоннах самописного кода и будете допускать баги(а при таком подходе вы неизбежно к этому придёте), то это намного более критичная ситуация.

Windows 95 работало быстрее? Хм… Ну давайте время отклика калькулятора посчитаем, но сколько приложений вы на нём запустите? А пользователи ждут, толпа жаждет хлеба и зрелищ! И когда вы начнёте удовлетворять потребности этой толпы вдруг окажется, что придётся писать менее оптимизированный, но более гибкий код.

Пост как по мне, так борьба с ветряными мельницами «Этот мир живёт неправильными приоритетами!», но десятилетия разработки ПО привели именно к этой расстановке приоритетов не просто так

Emulyator 20 сентября 2018 в 22:38

+2

Пользователи зачастую уже давно ничего не требуют, ну разве что «вернуть, как было». Но кто их слушает? Вот вам пользователи новая непривычная, тормозная, глючная версия, с добавлением сомнительного функционала, покупайте, осваивайте, привыкайте. Поддержка и продажа старой версии окончены, выбора нет.)

Asxer 21 сентября 2018 в 05:12

0

Ну про глючность это как раз аргумент в пользу того, что лучше переиспользовать готовые решения, чем писать каждый раз оптимизированный низкоуровневый велосипед.
Если дизайн надо заново осваивать, то это большой вопрос к дизайнеру проекта и к теме разговора отношения не имеет.
Новый ненужный функционал? Так говорите как будто его программисты придумывают:)

На счёт тормозов. Пользователь не хочет приложение, которое работает максимально быстро. Это никому не нужно. Пользователь хочет приложение выполняющее свою функцию в приемлемое количество времени. Разница между 0.05 с и 0.5 с для рядового пользователя настолько несущественна, что её никто не заметит. А это разница в производительности на порядок! А вот то, что ради этих 0.45с вы угробили почти весь бюджет заказчика, при этом писали исключительно свой код не используя фреймворки потому что они тянут лишние зависимости и занимают место. И как следствие, на разрежение лапши в коде тратится неприлично много времени, что приводит к появлению багов.

Итог приложение открывается на 0.45 с быстрее, но вот при нажатии на нужную пользователю кнопочку крашится. Зато быстро:)

Каждому решению своё место. Там где производительность является ключевым фактором, там приложения оптимизируются на минимальное время отклика и открытия, но в большей части софта это не нужно.
Когда тормоза в приложении начинают мешать им пользоваться, тогда приложение оптимизируется, но во-первых все современные решения позволяют оптимизировать приложение до того самого приемлемого уровня, а во-вторых определяющим фактором и узким местом становится уже не железо, а пользователь, который начиная с определенной задержки перестаёт чувствовать разницу и траты бюджета на дальнейшую оптимизацию становятся неоправданными.

Emulyator 21 сентября 2018 в 15:31

0

Вопрос в том, зачем что-то писать вообще, если уже работает за 0,05с и пользователей устраивает?
Вы взяли и замедлили производительность на порядок, на порядок раздули приложение сторонним кодом, стабильность которого по сравнению со старым никто не оценивал, наняли опытного дизайнера, который перерисовал интерфейс так, чтобы привыкнуть было не очень сложно, и все потому, что якобы приложению «калькулятор»(например) не выжить без испытаний постоянными обновлениями?)



PsyHaSTe 21 сентября 2018 в 15:47

0

Ну например чтобы починить старый баг? По-вашему старые баги чинить не надо и пользователи и так привыкли? Ну например, был баг в библиотеки шрифтов. Разраб залез, увидел дикие костыли, нашел на гитхабе стороннюю либу, которая это чинит. Проверил, всё работает, даже новые фичи появились, равно как и внешняя зависимость. Коммит, багфикс летит в прод.

Или вы предлагаете каждый баг костылить, чтобы одни и те же вещи люди каждый раз писали с нуля, "кастомизированно"? Ну это какой-то NIH в крайней стадии.

Emulyator 22 сентября 2018 в 15:51

0

О каких старых багах речь, если пользователей все устраивает?)
Я просто указываю на очень распространенные ситуации, для которых придумали пословицу «лучшее — враг хорошего», ну или «работает — не трогай».
Я понимаю, что можно долго обмениваться частными гипотетическими и реальными примерами, но они не дают права делать обобщения и уж тем более выводить универсальные правила создания и обновления программ.
Мой взгляд такой: обобщить и выдать универсальный подход по созданию и обновлению софта за пару абзацев невозможно. Можно накидать несколько общих размытых рекомендаций, которыми без опыта и удачи даже не каждый способен правильно воспользоваться. Например, если ты опытный авторитетный программер, разработал и поддерживаешь легкий, шустрый продукт с минимальным использованием сторонних библиотек, имеешь благодарных пользователей, ценящих эту легкость и шустрость, то у тебя будет один набор доводов добавлять или нет стороннюю, незнакомую тебе, тяжелую библиотеку с зависимостями. Если же ты начинающий, не уверенный в своих силах разработчик, или просто по каким-то причинам работающий в стиле «все уже придумано до меня, никаких велосипедов», то у тебя будет другой набор доводов.

Asxer 22 сентября 2018 в 17:21

+1

Я категорически не согласен с таким предвзятым подходом, что человек пишущий код заново и экономящий на спичках называется опытным и авторитетным, а человек строящий приложение на готовых решениях — начинающим и неуверенным.

Может быть мы с вами в принципиально разных сферах работаем, я работаю в сфере веба и мобильных приложений, где ваш подход смерти подобен. Самые жуткие проекты, которые мне доставались шли одним и тем же сценарием. Был разработчик построивший приложение на велосипедах потому что они быстрые. Фреймворки не нужны, сторонние библиотеки нужно сначала прочитать и тд. И всё вроде бы неплохо работает пока не появляется она — ФИЧА!
Это закон программирования — каждое изменение кода — потенциальный крах системы. Программист, который в вашем понимании должен оцениваться как опытный и авторитетный разработчик, начинает рушить свой идеальный, быстрый, но построенный на велосипедах код пытаясь встроить в него новую фичу. Откуда взялась фича? Ну у заказчика поменялась бизнес-логика, что-то изменилось в бизнес-процессе да мало ли! Это не вопрос программирования.
И вот код программиста уже выглядит не так хорошо, а тут появляется ещё фича, потому что новые изменения являются новыми и устаканиваются по ходу дела. Появляются новые баги — и всё это потенциальные краки системы. И тут велосипедеры дают о себе знать. И вот ваш изначально быстрый код крашится и валится, но при этом ещё и представляет из себя нерасступываемую лапшу. Заказчик терпит повышения бюджета месяцы, годы. Но потом наступает момент, когда проект просто умирает. Не из-за заказчика, а из-за программиста велосипедофила. И вот заказчик меняет команду разработчиков и потом нам приходится это разгребать.

Проект, где не появляется новых фичь это какая-то экзотика, я даже не припомню такого… Такое может быть только на проекте, который либо очень узконишевой, либо которым мало кто пользуется.

Emulyator 22 сентября 2018 в 19:39

0

Ну, начнем с того, что у меня вовсе не присутствует подхода, с которым вы *категорически не согласны*. Вариант, где программист «по каким-то причинам работающий в стиле «все уже придумано до меня, никаких велосипедов=» специально упомянут, чтобы избежать обвинений в предвзятости. Это может быть вполне опытный разработчик с 15 летним стажем, который трезво оценивает современные требования к разработке в своей нише, и понимает, что разумнее использовать чужие готовые наработки, чем набивать свои шишки. Т.е. это вполне положительный пример.
И я не престану утверждать, что в каких бы областях ни творил разработчик, какие бы парадигмы, подходы, инструменты ни диктовали бы рынок, мода, гуру, — нельзя без вдумчивого анализа конкретной задачи (и много чего еще в), принимать решение о том, как писать, на чем писать, что использовать, что нет. Коротко такой подход можно назвать здравым смыслом.)
А что касается веба, то вот подходящий к теме длинный пример из моей практики.
Контора, где я работаю, заказала у сторонней фирмы немного изменить сайт. На стороне, потому что в своем штате ни один айтишник вебом не занимался и желания не проявлял, а человек, делавший последнюю версию, толи денег много захотел, толи перестал интересоваться темой, не важно. Сайт обновили: перешли с самописного движка (php) на движок друпал, просто визуально/функционально повторив сайт. На том же хостинге новый сайт стал тормозить… По замерам на порядок — на два, и это объяснили большим уровнем абстракции, а значит гибкостью, универсальностью, последующей легкостью в изменении и поддержке. Я даже согласился с этим объяснением, и по совету разработчиков сайта, выбил у руководства перенос на рекомендованный, проверенный многократно, но более дорогой хостинг на vps. Худо бедно, но все зашевелилось со скоростью на уровне «есть сайты и потормознее, сойдет для сельской местности».
Очень скоро появились жалобы от посетителей, нет, не на скорость, а на то, что люди, условно говоря, подавали заявку, сайт «говорил» им, что успешно её зарегистрировал, но по факту в базу эти заявки не попадали, или на то, что сайт недоступен в короткий период подачи заявок на получение, скажем, товара повышенного спроса.)
Проблема возникала не регулярно, но довольно частая, и в результате многократных переговоров с разработчиками сайта дошло до того, что последние обвинили посетителей во лжи, и что у них все работает, проверили много раз. В конечном итоге выяснили, что посетители не виноваты, что компетентность разработчиков не позволила быстро и точно локализовать проблему, потому что глубоко они в друпале не разбираются, а в php и mysql почти никак. Зато быстро, как конструктором умеют делать сайты на друпале.
Пока шла вся эта тягомотина, мой напарник увлекся node.js, изучил react, поплевался на отсутствие типизации в javascript, освоил typescript, поизучал и, (как он посмел!) написал свой аналог redux, изучил основы работы с vps и нарисовал такой же сайт, на своем react.js движке практически без зависимостей, который летает, в котором ему понятно, что как работает. Несколько месяцев — и новый сайт, все довольны, ну кроме сторонней фирмы, с которой пришлось расторгнуть договор.
Никакой морали, просто подходит к контексту разговора.

Asxer 23 сентября 2018 в 02:46

0

Чтоб переписать изначально рабочий сайт с нуля требуются достаточно веские основания. А в вашем случае программисты просто рукожопы.
К тому же друпальное комьюнити это какая-то отдельная секта. Переписать сайт на технологию, которую не понимают достаточно глубоко, при этом не имея достаточных оснований для этого потому что… Ну это же Друпал!=) Я бы с ним вообще не связывался. А тут весь набор некомпетентности.
Тут как говорится «с дуру можно и МПХ сломать».

Emulyator 23 сентября 2018 в 11:15

+1

А основание простые. Хоть сайт и не такой уж сложный, но это был чистый php, довольно аккуратно, модульно написанный(даже я разобрался), а современное состояние в веб такое, что часто представители разработчиков даже не рассматривают ситуацию, что сайт может не использовать один из популярных движков. Они первым вопросом спрашивают: «на чем сайт?», и не давая ответить, пытаются угадать, перечисляя глупому заказчику те многочисленные движки, с которыми они умеют работать. А когда узнают, что движка нет в списке, вариант сделать все с нуля — единственный, который рассматривается, а предложение: «может посмотрите, как оно сейчас устроено, может разберетесь и подправите?» сразу отвергаются. И я могу понять этот подход с экономической точки, — «делают то, что умеют», но в результате такие вот истории.)

0xd34df00d 24 сентября 2018 в 07:14

0

Что-то я прямо как-то даже неудобно себя почувствовал, что с друпала мигрирую на генератор статических страниц.

PsyHaSTe 22 сентября 2018 в 23:06

0

> О каких старых багах речь, если пользователей все устраивает?)

Например отсутствие поддержки lf в notepad в винде. Может пользователи привыкли, но это не нормально.

> Я просто указываю на очень распространенные ситуации, для которых придумали пословицу «лучшее — враг хорошего», ну или «работает — не трогай».

Обычно сразу после этого следует фраза «давайте лучше закостылим». Не люблю это правило в большинстве случаев.

Emulyator 22 сентября 2018 в 23:53

0

Вот тут и требуется умение не быстро прикрутить проверенную библиотеку, а взвесить затраты и главное последствия всех подходов. В список подходов вполне может входить и сторонний код, и самостоятельное исправление, и даже тупое игнорирование проблемы. Оно примерно так все и происходит, просто, к сожалению, люди, ответственные за выбор варианта, зачастую либо не замораниваются с глубоким анализом, а работают по накатанной, либо в своем глубоком анализе преследуют цели, отличные от «сделать пользователям хорошо». Например, преследуют цель «сделать быстрее», либо «сделать идеально», либо «не делать ничего», либо «сделать, потому что умею» и т.д.
…
Костылями вполне можно считать как прикрутку тяжелых библиотек с ненужными зависимостями, так и написанную на коленке кривую заплатку.

Asxer 21 сентября 2018 в 16:46

0

Ну появление новых требований диктуется не разработчиком, а заказчиком и пользователями.
Зачем делать редизайн? Ну может потому что у конкурентов дизайн адаптивнее(новые устройства появляются регулярно и ваш старый сайт на новом экране может выглядеть не так хорошо), красивее и современнее.
Плюс заказчик просит фичи потому что… Ну это уже от проекта зависит. Требования пользователей год назад, когда ваше приложение только появлялось и было в новинку и требования сейчас, когда они его распробовали будут серьёзно разниться. Да, господи, просто посмотрите как изменился контакт. Если сейчас выкатить его в старой форме приблизительно лет десяти назад со стеной, отсутствием аудиозаписей, только одной аватаркой и тд и тп, то этим сайтом просто никто не будет пользоваться.
Ваш продукт должен быть дружелюбен к изменениям потому что они наступят ГАРАНТИРОВАННО. Расчитывать при разработке на приложение, которое будет оптимизировано один раз и не будет модифицироваться, значит изначально писать либо крайне узконишевой продукт, либо мёртворождённый.

Почему потребовалось переписывать калькулятор? Не ко мне вопрос, калькулятор некоммерческий проект, денег не приносит и монетизации не имеет. Тут скорее всего потому что просто кому-то захотелось, но сделать так, чтоб бюджет сильно не тратить и потому сделали всё халтурно. У меня нет полного ответа в частности про калькулятор.

Emulyator 22 сентября 2018 в 16:48

+1

> Ну появление новых требований диктуется не разработчиком, а заказчиком и пользователями.

Да, так бывает, это очевидно, но также очевидно, что бывает, когда решения об обновлении принимают сами разработчики (например, когда конторка маленькая, или вообще разработчик один, он же маркетолог, дизайнер и т.п), или когда конторка большая, и решение принимают не разработчики, а специально обученные люди, иногда даже вопреки пожеланиям пользователей.

> Зачем делать редизайн? Ну может потому что у конкурентов дизайн адаптивнее(новые устройства появляются регулярно и **ваш старый сайт**

а если не сайт, а если 3д редактор, или операционная система, или текстовый процессор, или калькулятор или просто у конкурентов сайт еще менее адаптивный? А может просто потому, что вкладывая незначительные усилия в ненужное пользователям перекрашивание и тасование кнопок и закладочек, можно объявить о выходе новой версии, а затем рекламой, юридическими ходами, склонять пользователей к объективно ненужному обновлению?
А может просто начальство требует новый релиз каждый год, а у маркетологов нормальные идеи закончились, а место терять не хочется и к легкому калькулятору придельвают тормозные анимированный кнопки, затем конвертер валют, единиц измерения, вычислитель ипотеки, построитель графиков, редактор хода, магазин скриптов, чат, но так и не делают нужный некоторым юзерам квадратный корень?)
Я ради интереса недавно зашел на sql.ru где когда-то участники раздела про MS Access переживали неудачную политику развития этого изначально весьма удачного продукта. Так вот, там до сих пор бытует устойчивое мнение, что не стоит использовать версии выше 2003, а 2003 до сих пор успешно применяется многими в разработках включая коммерческие. Это каких же таких «полезных» изменений надо было надобавлять, что опытные пользователи уже 15 лет отказываются обновляться?)

Asxer 22 сентября 2018 в 17:04

0

Ну вот извините, но практически весь ваш комментарий — попытка разобраться, откуда исходят требования. Но речь-то не про источник требований, а как развиваться кодовой базе в условиях постоянно появляющихся хотелок и требований. Источники могут быть разными, часто разработчики с ними не согласны, иногда не согласны с ними и пользователи, но заказчик неумолим. Сути это не меняет — есть требование, которое нужно удовлетворить, за него деньги платят. Если каждый разработчик будет критиковать механизмы монетизации приложения, то это прямое нарушение разделения обязанностей на проекте. У меня были заказчики, которые очевидно стартовали проекты, которые по моему мнению не окупятся, мы говорили заказчику, но он был неумолим и проекты не окупались. Но точно так же были и проекты, где мы предупреждали заказчика об аналогичных проблемах и его фичи стреляли и приносили деньги. Каждый смотрит со своего ракурса и не всегда то, что видит разработчик соответствует тому, что видит заказчик. Пусть каждый решает свои задачи.
Постоянно появляющиеся требования разработчиками должны восприниматься как данность, как закон природы, к которому они **должны быть готовы**.
ТС же причитает, что разработчики не экономят ресурсы железа. Это плата за гибкость к изменениям. Тут применим эволюционный процесс и выше указали в комментариях на это — были разные подходы и разработчики, но самым жизнеспособным оказался подход — главное писать гибкий код, дружелюбный к новым требованиям, чем код, скрупулёзно считающий такты процессора. К сожалению это очень часто взаимоисключающие направления.

Emulyator 23 сентября 2018 в 00:17

0

Если бы ТС причитал необоснованно и в гордом одиночестве, то тут не набралось бы под 1,5К комментов. А значит «не все так однозначно»(с).
Что касается эволюционного процесса, то да, это он. Но я даже где-то выше (или ниже) довольно развернуто указал на минусы такого процесса.
Грубо говоря и чтобы не плодить простыней: эволюция «сосет» у разумного замысла. И ТС, подымает сложную проблему, в рамках которой невольно предлагает вспомнить, что мы — разумные существа, а не крысы выживающие в лабиринтах построенных социально-экономическими реалиями.)

PsyHaSTe 23 сентября 2018 в 03:57

0

Есть спектр решений, так вот гибкое решение почти всегда лучше, чем быстрое. Эволюция относится ко всем вещам, включая и разработку. У нас нет никакого «бога», чтобы скоординировать общие усилия. Изменяющиеся требования — закон природы, и если вы сделали классный индекс на уникальный номер пользователя в БД, и завязали на это кучу логики, а потом оказалось, что в реальной базе есть дубли (хотя вроде не должны быть, но вспомнили один кейз, почему так происходит иногда), и всё пошло коту под хвост. Невыдуманная история.

Оптимизации возможны при большом количестве априорной информации о системе. А этой информации может быть недостаточно, она может быть просто ложной, и она может быть достаточной и корректной, но поменяться в будущем. Чем больше оптимизированно решение, тем больше оно опирается на некоторые допущения — количество ядер, количество запросов, ценность ЦПУ относительно памяти, и многое другое… Каждое такое допущение — потенциальная дыра и недели исправлений.

Emulyator 23 сентября 2018 в 11:40

0

Ну тогда можно сказать, что «вся боль» о том, что решения стали слишком гибкие, слишком универсальными, слишком много слоев абстракций, потому что у нас нету «бога», способного сказать: «Стоп! Твой монстр еле шевелится, а пользователи убежав к конкурентам, облегченно выдыхают, мгновенно загружая простенькую программу, которая при этом не содержит и 90% функционала твоего монстра». И, возвращаясь к теме «заказчик заказал, программист просто исполнает», я тут затеваю ремонт в квартире, смотрю как и все «ботаники» ролики про ремонт на ютубе. Очень понравился подход одного опытного мастера, который настаивает на том, что нужно делать как надо, а не как хочет заказчик, это в конечно итоге будет лучше для обоих. Понятно, что это все общие лозунги, но и лозунги «заказчик платит», «клиент всегда прав» явно требуют осмысления в каждом конкретном случае.

PsyHaSTe 23 сентября 2018 в 13:55

0

То есть если вы закажете плитку в ванной положить, а исполнитель постелит линолеум с формулировкой «я 20 лет этим занимаюсь, тут линолеум лучше» вы скажете «ну ок, вы же специалист»?

Emulyator 23 сентября 2018 в 15:42

0

Если отвлечься от абсурдности надуманного примера, то да. Вы же не заказываете хирургу проводить аппендэктомию через задний проход, потому логично рассудили, что так шрама не останется? Вы, если повезет, вольны выбрать хирурга из нескольких, передать ему максимально точно свой анамнез, упомянуть об аллергиях и хронических заболеваниях, а дальше просто довериться выбранному профессионалу. Парикмахеру и строителю конечно можно указывать в более широком диапазоне, но если исполнитель 20 лет в профессии, значит есть отзывы, рекомендации, законченные объекты, на которые можно посмотреть, потрогать, спросить мнение заказчиков и сравнить с другими. Да, желательно, чтобы строитель мог объяснить причины по которым он отказывается делать водяной теплый пол в многоэтажке, или отказывается присоединять лоджию в доме с внешним фасадным остеклением, но когда количество «хотелок» такое, что вместо ремонта придется обучать заказчика, то право исполнителя задрать ценник или отказаться от объяснений. Почему в программировании должно быть иначе? Из-за того, что конкуренция такая, что заказчик уйдет к студенту, который за еду будет исполнять все причуды пока не разочаруется в программировании? В каждой области встречаются и халтурщики и профессионалы, и я думаю есть профессионалы на которых есть спрос и высокий ценник, которые позволяют себе отказаться от заказа, если клиент лезет не в свою область, не понимая последствий.

PsyHaSTe 23 сентября 2018 в 16:55

0

Да нет же, как раз строитель должен все объяснить, верно. Но если клиент говорит «мне пофиг, все равно хочу так», нужно делать так, а не саботировать своим мнением продукт.

khim 23 сентября 2018 в 18:04

0

> Но если клиент говорит «мне пофиг, все равно хочу так», нужно делать так, а не саботировать своим мнением продукт.

В строительстве есть требования, за нарушение которых начинают всё жёще и жёще наказывать.

Думаю и в программировании этим кончится.

Emulyator 23 сентября 2018 в 18:10

0

Угу, а потом получить уголовку за экспериментирование на живых людях для хирурга, или для строителя судебный иск за ненадлежащий ремонт, из-за которого управляющая компания требует от прикинувшегося неинформированным веником владельца квартиры устранить в 3-х дневный срок нарушения. Или получить смерти людей, потому что кому-то несущая стена помешала расширить метраж. Да, можно сказать, что примеры притянуты, что в разработке таких последствий не бывает, но думаю что бывают, пусть реже, но даже просто прогнозируемый гемор в дальнейшем — уже повод задуматься: «а стоит ли связываться с таким упертым заказчиком?».

PsyHaSTe 23 сентября 2018 в 21:58

0

Вы куда-то не туда ушли со своей аналогией. Врачи вообще не про то, они с жизнями работают. От того, что на сайте будет кнопочка немного другая никто не умрет. Не надо притягивать плохие аналогии, а потом делать далекоидущие выводы.

Emulyator 23 сентября 2018 в 23:08

–1

Плохая аналогия это когда некий профессионал знает свою работу хуже, чем заказчик и отказвается класть плитку в ванной. Но далеко не все врачи прямо таки с жизнями работают, и не все разработчики красят кнопки на сайтах в угоду капризным клиентам. В современном мире программисты в медицине реботают не меньше, чем отдельные врачи. Если так уж угодно, никому не понравится когда перегруженный фичами по желанию заказчика сайт электронной записи на прием муниципальной больницы упадет из-за всплеска нагрузки от посетителей и оставит людей без медицинской помощи, или позволит хулигану засрать расписание фейковыми заявками, потому, что заказчик свято верил в невероятность такой ситуации и сэкономил на контроле. Я уж не говорю об ответственности разработчиков в областях связанных с различными мед.исследованиями. Я был личным свидетелем когда толи неправильно настроенная толи содержащая ошибку банальная программа по ручному заполнению формы электронной карты пациента после флюорографии в полях новой карты не очищало поле диагноза, смущая врача, и заставляя его предположить наличия какого-то экспертного алгоритма, которого там в помине не было. Тоже наверное кто-то настоял, не стирать, чтобы меньше теста набирать если совпало.)

ainoneko 21 сентября 2018 в 09:56

+3

> Ну давайте время отклика калькулятора посчитаем,

Спасибо, что про него вспомнили: недавно кое-где и калькулятор сломали.

Ogra 21 сентября 2018 в 10:00

0

Достал, проверил. Ох ты ж блин…

Asxer 21 сентября 2018 в 10:35

0

Ну тут проблема понятна — пока отрабатывает предыдущая анимация кнопка знака не срабатывает и получается каша. По идее просто халтурно сделанное приложение и проблема отношения к тестированию софта, не ожидал от Эппл такого.

Но к теме разговора это никакого отношения не имеет. Проблема-то не в том, что телефон НЕ УСПЕВАЕТ отработать калькуляцию и даже не в том, что установочник калькулятора весит много, хотя мог бы весить всего ничего и даже не в том, сколько ресурсов кушает данное приложение(а именно эти проблемы заявлены в статье), а в том, что разработчики халтурно подошли к реализации и тестированию своего софта

Tarson 20 сентября 2018 в 20:01

+3

Windows 10 больше ибо в ней все предыдущие версии Windows.

minusnaminus 20 сентября 2018 в 21:53

+1

И видимо, все нерожденные версии тоже.

sumanai 20 сентября 2018 в 22:37

0

Ладно хоть не все последующие.

minusnaminus 20 сентября 2018 в 23:24

0

Так говорили что десятка — последняя. То есть, следующая ось будет называться не виндоус, а как-то иначе

PsyHaSTe 21 сентября 2018 в 01:39

0

Ну логично, учитывая, что все последующие ОС по сути являются обновлениями десятки.

На самом деле любопытно, выродятся ли все эти Singularity/Midori/Barrelfish/… во что-то реальное или нет. Пока что я видел из применения — пикопроцессы в винде. Но это — капля в море… Груз обратной совместимости слишком велик, и вообще непонятно, удастся ли от него избавиться… Видимо только если мы все в какой-то момент перейдем на какой-нибудь электрон (избави бог), и можно будет сменить ОС на что угодно, что умеет в JS…

kovserg 21 сентября 2018 в 09:02

0

С грузом обратной совместимости можно было справиться с помощью виртуализации. Посмотрите на DOSbox это полная эмуляция старого компа. Win3.11 пускается под ни даже на планшете.
Но микрософт страдает фигнёй выпуская кучу версий одного и того же но другого net runtime-ов, vcredist-ов. Вместо того что бы навести порядок и выпустить новую ос которая могла работать паралельно они пытаются разделить runtime на кусочки api-ms-win-*.dll которые не являются библиотеками, зато уже имеют разные версии. Не смотря на все годы микрософт так и не смогла предложить простой и удобный инструмент для разработки gui приложений. Зато теперь пытается активно возвращаться к терминалам, добавляя wsl и улучшая консоль.

PsyHaSTe 21 сентября 2018 в 13:16

0

> С грузом обратной совместимости можно было справиться с помощью виртуализации.

Этот путь уже попробовали пройти с Itanium, чем это закончилось всем известно

K0styan 21 сентября 2018 в 15:55

0

То 15 лет назад было. Сейчас Google неплохо справляется с виртуализацией, поддерживая нативные приложения Android и Linux в своих Хромбуках. Microsoft позволяет запускать x86 приложения (не только UWP, но и нативные) на ARM процессорах. Разумеется, и то, и то — с оверхэдами. Но в целом тенденция радует: возможно, от проклятья обратной совместимости удастся избавится довольно скоро, что даст дорогу свежим платформам.

Am0ralist 21 сентября 2018 в 17:47

0

Она то позволяет, да Интел ей запретила…

uchitel 20 сентября 2018 в 20:02

0

Переводчику зачет.

Автору повезло, что он не жил в России и не сталкивался с нашим софтом, который покупает государство для налогоплательщиков на деньги налогоплательщиков, сделанное по принципу: «Если работает — не лезь, не трогай, не спрашивай!». Однажды мне попался один такой «разработчик», спросил его сколько он получает (!) 30000 рублей (!!!) в Новосибирске. И этим софтом вынуждены (принуждены) пользоваться тысячи людей. Представьте, вам нужно в онлайне заполнить таблицу, всего-то 20 полей и 15 строк — всего-то. Нооо это слишком скучно! Пусть после каждого перевода курсора на следующую ячейку происходит задержка, секунды по 1,5-2, зато данные (наверное) сразу подгружаются на сервер. Хотя нет и это слишком неинтересно, когда таблица будет почти готова, пусть выскочит сообщение «Неизвестная ошибка. Нажмите „Отправит“ (именно „Отправит“, а не „Отправить“) и мы постараемся ее исправить». А данные, на самом деле, на сервер не подгружаются. Все равно скучно и некрикативно. Пусть после каждого перевода курсора, происходит автопрокрутка страницы в ее самое начало, а при масштабировании заголовки полей масштабироваться не будут. Вот теперь нормально, но мы еще обязательно что-нибудь придумаем… постойте-постойте! А пусть оно, типанормально работает только на десктопах и только под IE.

Python — это вообще в точку! С его дефакто-стандартом в области MLAI! Не, когда начинаешь, то всё и правда розовое вокруг. Потом всё тоже не плохо, так, всего-то парочка трещин на очках, как у Гари Поттера. Но вот когда дело доходит до оптимизации… когда нужно лезть в дебри исходников большинства импортированных библиотек, когда ты понимаешь «Что документация — это для слабаков и отстойщиков». Что под капотом-то оказывается Fortran, C, C++, Java, CUDA. Вот тогда-то розовые стекла и сыпятся.

На Питоне начал писать медленно, реально медленно. Иначе хорошо не получается.

Самое обидное, что если ты пишешь хорошо но медленно — то ты безработный. А если ты пишешь хорошо но быстро (работаешь еще и по ночам), (!) даже подробно комментируешь код, то ты все равно безработный, потому-что «Нафиг он нужен? Если он свалит (а поработав у нас он обязательно свалит), то после него в этой мутатени никто не разберется, а если и разберется, то попросит больше».

kvasdopil 20 сентября 2018 в 21:36

+3

Автор текста — программист из Новосибирска, кстати.

PsyHaSTe 21 сентября 2018 в 01:46

+1

«Нафиг он нужен? Если он свалит (а поработав у нас он обязательно свалит), то после него в этой мутатени никто не разберется, а если и разберется, то попросит больше».

Нафиг места, где так думают) У меня два раза такое происходило, и оба раза я уходил в более хорошее место. Нужно писать так, чтобы не было за себя хотя бы стыдно. Так что самосовершенствоваться никому не мешало. Я вот тут на текущую работу когда устроился, моя продуктивность возрасла просто раз в 5 по сравнению с предыдущим унылым местом. Для сравнения — на третий рабочий день я за 8 часов успел проект на 350к строк превратить в проект на 50к строк без потери функциональности. Просто проанализировав всё, что мне попало в наследство. На следующий день у нас был разговор с СТО на тему «нужно как-то кэшировать загрузку с сервера». После прояснения, стало понятно, что первоначальный запуск приложения занимает 6 минут и это очень дофига, надо бы как-то кэшануть. Еще один день разработки — программа грузит все данные за 8 секунд, кэширование перестало быть нужным (а с ним отпал и целый пласт связанных проблем, ну вы знаете..). И это первая рабочая неделя только, где я просто знакомился с проектом и людьми… А всё почему? Потому что реально атмосфера, где я могу делать что-то классное, где мое мнение имеет значение, где плоская иерархия и я спокойно могу зайти к гендиру и пообщаться на тему того, почему обещанные заказчику сроки и требования несовместимы, и вместе с ним и СТО выработать план действий…

Очень много зависит от атмосферы, на текущей работе я понял это сильнее, чем когда бы то ни было.

uchitel 21 сентября 2018 в 09:53

+2

Когда был студентом, пригласили в лабораторию. Быстро вник что к чему, моделирование рынков и все такое. Слепил в wolfram mathematica (программировать на тот момент вообще не умел — можно сказать, был дебют) скрипт, все довольно неплохо заработало. Самое главное — это казалось вообще волшебством (не мой скрипт), а сама wolfram mathematica. Но приехал руководитель лаборатории из штатов, привез какую-то программулину и сказал, что они вычисляют по ней, значит, мы тоже будем на ней работать. Все мои наработки вместе со мной, моими вопросами и предложениями нафиг.

Ребята потом рассказывали, как он привозил литературу, тыкал пальцами в американские графики и говорил «Во! У нас так же получается!». Его даже по телику показывали, приглашали экспертом в программы.

Все что меня сейчас выручает, так это старые хорошие знакомые. А вот с атмосферой, почему-то никогда не везло. Просто мои знакомые там, как-то могут работать, а я нет.

Поэтому и сижу, дома в своем кабинете.

Konachan700 20 сентября 2018 в 20:08

+10

Всё нормально, ложная тревога.
Первые ездовые животные тоже были своенравные, медленные, опасные и в целом неудобные. Прошло время и мы получили прекрасных лошадей. А занимайся мы выведением сразу идеальной лошади, так и сидели бы в пещере до сих пор. Первые паровозы и автомобили нещадной коптили, гремели, ломались, и были диким ужасом. Прошло время и мы получили отличные поезда и прекрасные автомобили. А займись полировкой паровоза/автомобиля тогда — до сих пор конский навоз с улиц вывозили бы. И так со всем, за что не возьмись. Сначала технология сырая, кривая и неудобная, и пока ее взрывной рост не закончится, она будет таковой — урода никто не будет исправлять, пока за ним стоит очередь. Когда все наедятся уродцами, взрывной рост закончится, и технология войдет в стадию полировки — начнут и оптимизировать, и выкидывать ненужное, и иными способами вылизывать продукт.
В IT сейчас стадия взрывного роста, самый ее пик. Продукт тормозит? Досыпте ресурсов облаку, это

дешевле найма дефицитных кадров. Приложение полгигабайта? Держите айфон на 512 гигов и успокойтесь. И это хорошо на самом деле. Это помогает развитию индустрии. В любом случае рано или поздно будет пойман потолок технологий, где экономически станет выгоднее нанять 10 человек, а не добавить условных 100 мегагерц процу, и в тот момент начнется работа не на увеличение мощности, а на оптимизацию использования того, что есть.

novice2001 21 сентября 2018 в 10:24

+1

Совершенно наоборот.
У нас уже **были** отличные автомобили (или что угодно ещё). Но потом кто-то подумал — а давайте приделаем к машине красивую мигалку. ОК, приделали. Но возросло аэродинамическое сопротивление, расход топлива, упала динамика. Давайте поставим мотор помощнее. ОК, поставили. Динамика вернулась, но расход топлива вырос еще больше. А еще пришлось переделывать подвеску, потому что мотор тяжелее, и шины большей размерности ставить. И т.д.
В итоге мы имеем монстра, который все так же перевозит водителя и четверых пассажиров, зато облеплен мигалками, ленточками и рюшечками. При этом потребляет в 10 раз больше бензина и облагается вдесятеро большим налогом. Но некоторым именно это почему-то кажется прогрессом.

volchonokk1 21 сентября 2018 в 11:33

0

Вообще-то экономичность автомобилей в последние годы только растёт. Этим озабочены как автопроизводители, так и законодатели. В США пробег на одном галлоне бензина пассажирских машин вырос с 15 миль в 70х до 25 миль сейчас — т.е. потребление топлива упало в 1.5 раза. Это уже не говоря о возросшем интересе к электромобилям в последние годы. Конечно ещё остались всякие спорткары, которые жрут непомерно, но ими пользуются только энтузисасты.
Всё потому что автомобили подошли к лимиту добавления рюшечек и лимиту топлива, и ресурсы автопроизводителей переключились на экономичность. Тоже самое будет и с программами. Просто сейчас до лимита рюшечек и ресурсов компов пока ещё далеко. Если в мобильники можно впихнуть 8 гигов оперативы и полтерабайта накопительной памяти — никто не будет особо думать об оптимизации.

novice2001 21 сентября 2018 в 11:58

+2

Вы в аналогии не умеете? Комментарий на который я отвечал сравнивал программы в том числе и с автомобилями. И в моем комментарии автомобили — это не автомобили, а именно программы. Про состояние автомобилей я прекрасно осведомлен и без вас.

BD9 22 сентября 2018 в 05:17

0

Не только софт.
В новых материнских платах Gigabyte (Aorus) вместо разъёма для com-порта размещён разъём для светодиодной ленты. Разъём для M.2 накопителя — под видеокартой, зато с радиатором.

novice2001 22 сентября 2018 в 09:40

0

Ну, com-порты сейчас и правда нужны далеко не всем. Но эта мода на светодиодную подсветку всего и вся — да, типичный пример тупости потребителей и падкости на моду.

Tarson 20 сентября 2018 в 20:17

+4

Вот поэтому длина клетки человек ДНК аж джва метра! Там те же проблемы. А птички смогли укоротить, с лишним летать тяжело.

Mad__Max 22 сентября 2018 в 03:27

+1

И «рабочего» кода если верить ученым всего около 2%. А 98% занято не пойми чем — в основном огрызками старого, уже давно неиспользуемого кода, включая код вирусов, которыми когда-то были заражены далекие предки. Хотя некоторые куски иногда оказываются задействованы в неожиданных местах…

IRainman 23 сентября 2018 в 18:16

0

На счёт 98% не пойми чего, там многое не понятно. Коды вирусов, например, если они включены в ДНК в обратной последовательности к тому как они расположены у вируса приводят к тому, что вирус уже не может размножаться в такой клетке… собственно так, примерно, и работает вакцинация, это база данных того что встречалось ранее. Ещё не до конца понятно где и что отвечает за инструкции по сборке ибо клетки плода и стволовые клетки изначально одинаковые, а потом всё это развивается в гораздо более сложную систему. В общем недоизучено пока что.

InterceptorTSK 23 сентября 2018 в 19:00

0

Экспрессия генов есть же на это. И сборки нет никакой вовсе. Клетка от какой-то условной клетки будет множиться влево достаточно точно 50 раз. Эти же клетки будут множиться вправо 150 раз. Таким образом вы и получите как бы «сборку». Только это сборкой не назовешь. Каким генам когда срабатывать — почитайте. Все это достаточно понятно и объяснимо, причем давно. Некоторые детали реализации не совсем ясны, но глобально как и почему — объяснено полностью.
Про 98% непонятного тоже объяснено. Оно для экстремальных условий. И снова работает тут экспрессия генов. На дрожжах же ставили опыты. У них какой-то процент генов в норме работает. А остатки как бы не нужны и не понятны. Начните подсушивать дрожжи — и внезапно будут кодироваться белки из этого якобы «ненужного» набора генов, они обеспечат выживаемость дрожжам в такой вот экстремальной ситуации.
И т.д.

firehacker 20 сентября 2018 в 20:35

+2

Очень приятная статья. Я думал, я одинок в своих взглядах на то, насколько катастрофична и печальна ситуация в отрасли, но статья и одобрительные комментарии прямо подняли настроение.

К сожалению, всё сложнее и сложнее найти заказчика и работодателя, который позволил бы тебе работать *правильным образом*.

lxsmkv 20 сентября 2018 в 23:31

0

В продуктовой среде (то что можно пощупать) такая же беда. Если ты точно знаешь, что тебе нужно, можно дни напролет безрезультатно лазить по интернет-магазинам. Есть только «барахло», и копия «барахла», но дешевле. Если ты неразборчив и принадлежишь к 80 % обыкновенных потребителей — то выбор, конечно, бесконечный.

Мне не понятно зачем на рынке столько моделей тостера, которые отличаются только формой и материалом. Уникальных особенностей нет ни у одного. Потому что никто не ставит такую задачу.

Пока покупателя будут называть «потребителем» — ничего не изменится :) Получается мы сами виноваты в своем горе. Мы — неразборчивы. Мы — не требовательны.

vdasus 21 сентября 2018 в 10:27

+2

На самом деле, из моего опыта, очень немного людей понимают что означает «писать правильно». Более того, за последние два года (из ~30 лет айти) я сам сильно поменял свое мнение на некоторые, кажущиеся «правильными», вещи. А самое печальное, когда решения может принимать человек, который думает, что он-то как раз знает как «правильно» (а на самом деле подучиться бы ему чуток). Так что сложнее не найти работодателя, а все еще сложнее.

Rohan66 20 сентября 2018 в 20:53

+2

Тут поминают лунную программу и атомную бомбу. Я видел ТЗ на создание атомной бомбы — это 1 (одна!!!) страница текста…
А про софт — была отличная писалка дисков Nero. Постепенно скатилась в монструозное поделие и, вроде, померла.
А Ворд? 90% пользователей используют 10% функционала. Да и версии между собой не полностью совместимы.

khim 20 сентября 2018 в 21:09

0

> А про софт — была отличная писалка дисков Nero. Постепенно скатилась в монструозное поделие и, вроде, померла.

Как померла? Зайдите на www.nero.com/burning_rom/2018 — пожалуйста: Nero Platinum 2018. Если вы дисков не пишете, то это не значит, что никто их не пишет!

Другое дело что смысла в ней нет — версия 10-летней давности будет в 10 раз меньше, быстрее… и будет делать то, что нужно. То же самое — с офисом и прочими программами…

Rohan66 20 сентября 2018 в 21:36

0

> пожалуйста: Nero Platinum 2018

Я его перестал отслеживать (и использовать) после Nero 7.0. Так что его жизнь меня не интересует.))))
И так происходит со всем софтом (практически). Добавляют свистоперделки, наращивают непонятную мошbbbb (а заодно и размер), а вот надо ли это — никто, вроде, и не задумывается.
Для себя — я не вижу большой разницы между Word 6 и Word 2016. Быстрее по клавишам я стучать не стал, а форматирования мне и в 6-ом хватало.
А вот когда мне пришёл грифованный документ в 2013, а я открыл его в 2010 и потерялась страница — вот это была проблема! Причём потерялась из-за какого-то форматирования (а не реально).

khim 21 сентября 2018 в 01:18

0

> Для себя — я не вижу большой разницы между Word 6 и Word 2016.

А я — вижу. В Word 6 нет поддержки Unicode и, соответственно, сделать документик на русском языке с парой-тройкой немецких слов — это страшная боль. Вот только… всё это появилось в MS Office 97. В MS Office 2000 появилась возможность делать copy-paste для HTML-документов.

Больше я ничего не знаю такого, что было бы реально критично для существенного процента населения.

> А вот когда мне пришёл грифованный документ в 2013, а я открыл его в 2010 и потерялась страница — вот это была проблема! Причём потерялась из-за какого-то форматирования (а не реально).

При переключении с принтера с 300dpi на принтер 360 dpi (24-игольчатый матричник) или наоборот такое умел делать Word 6 уже (а может и Word 2).

Так что это не новая фича…

Rohan66 21 сентября 2018 в 09:32

0

> При переключении с принтера с 300dpi на принтер 360 dpi (24-игольчатый матричник) или наоборот такое умел делать Word 6 уже (а может и Word 2).

Сталкивался с таким во времена 95-98 винды.))) При разных установленных принтерах. И, на мой взгляд, от dpi не зависело особо. И, честно говоря, думал, что этот баг пофиксили.
Но в моём случае всё оказалось хитрее. Примерно 4/5 документа в обоих версиях были идентичны, а вот дальше потихоньку начались расхождения — примерно по предложению на лист и к концу документа лишний лист и набежал…

khim 21 сентября 2018 в 22:25

0

> Примерно 4/5 документа в обоих версиях *были* идентичны, а вот дальше потихоньку начались расхождения — примерно по предложению на лист и к концу документа лишний лист и набежал…

Так оно и работало всегда. Изменение DPI меняло характеристики шрифтов, после чего слова переставали влазить в строку (или наоборот — влазили там, где раньше не могли), строки сдвигали абзаци и так по нарастающей… и вот уже у вас книга в 399 или 401 страницу вместо 400…

Принципиально нерешаемая проблема программы, где вся вёрстка строго заточена под принтер, на котором документ будут печатать.

> И, честно говоря, думал, что этот баг пофиксили.

Тогда попыла бы вёрстка у людей, которые переходят с одной версии на другую без смены железа. Microsoft старается этого избегать.

Rohan66 21 сентября 2018 в 23:10

0

Ну-ну… Старается… Две одинаковые машины, с одинаковыми принтерами. Только на одной 2010 офис, а на другой — 2013.
Как же тогда народ документами обменивается? Была бы проблема в DPI — вообще бы совместимости не было…
Ну а про преобразование формул в рисунки в разных версиях? Тоже принтеры виноваты?

khim 22 сентября 2018 в 03:51

0

> Как же тогда народ документами обменивается?

С трудом.

> Была бы проблема в DPI — вообще бы совместимости не было…

А её и нет. И никогда не было. MS Word был изначально заточен строго под подготовку документов к печати. И несмотря на огромное количество костылей, вбитых за долгие годы, ни с чем другим он на 100% не справляется.

> Ну а про преобразование формул в рисунки в разных версиях?

Нет, это проблемы с лицензиями. В старых версиях Microsoft пользовал урезанную версию MathType, а в новых — сделал свою приблуду. Разумеется так как исходников MathType у них никогда не было — получилось… как получилось. Хорошо ещё что OLE2 толком не взлетел — а то сейчас бы новая профессия была бы: «криминалист-исследователь документов», который бы умел понять — какие версия каких программ нужно установить на компьютер, чтобы прочитать конкретный документ.

Похожая история с поддержкой ZIP-файлов в Explorere'е…



Amöralist 22 сентября 2018 в 10:41

+1

Если честно, главная проблема в том, что люди редко делали документы — нормально.
А так я видел и книги, набранные в ворде, которые нормально между компами переносились и ручное оглавление в которых всегда оставалось верным. Ну, то есть если правильно использовать табы, разрывы строк, страниц и прочее, а не отбивать энтерами и пробелами, то, внезапно, документ так не бился.
И которые потом в мелких типографиях печатали спокойно. Вот прям из ворда.



Alexeyslav 24 сентября 2018 в 15:44

0

Это не баг. Это фича. Вёрстку в текстовом процессоре ЗАТАЧИВАТЬ не нужно, предполагается что вы работаете с текстом в текстовый процессор(Word) делает всё остальное чтобы распечатать текст красиво и по правилам. Но тут люди начинают вмешиваться в логику программы — там слово не влазит сделаю шрифт чуть поменьше и оно влезет, там пробелами подравняю и слово станет на своё место. В итоге получаем такие приколы. Чтобы этого небыло, нужно изучить все функции текстового процессора и его логику, но кто же этим утруждается кроме профессионалов?
Для верстки существуют несколько другие программы другого назначения где всё завязано на дизайн а не на текст.

sumanai 20 сентября 2018 в 22:40

+2

> Nero Platinum 2018

Всего лишь 1,68ГБ в репакнутой версии…

Fracta1L 21 сентября 2018 в 09:03

+1

При том что есть софтина для записи дисков размером в 398 Кб и не требующая установки: avt-lab.ru/scdwriter.htm

А вообще, легковесных программ до сих пор предостаточно: 3dnews.ru/975310

Rohan66 21 сентября 2018 в 09:33

0

Нормальная ОС меньше места занимает! (вместе с программой для записи дисков!)

Mad__Max 22 сентября 2018 в 03:34

+1

Nero551020.exe — 12.7 МБ

Большой «путь» однако они проделали. Ожирение в 130 раз за 13 лет.

vis_inet 21 сентября 2018 в 12:37

0

Я пользовался программкой «SCDWriter» — наверное, самый минимум — всего 400 Кб!
ru.wikipedia.org/wiki/Small_CD-Writer

vis_inet 21 сентября 2018 в 12:35

0

> Я видел ТЗ на создание атомной бомбы

Дайте и нам посмотреть?

Rohan66 21 сентября 2018 в 12:55

0

С удовольствием бы! На стене в рамочке висело. Но нельзя там было фотографировать. Это музей ЯО в Сарове. Есть только фото рядом с Царь-бомбой!)))))

Alexx31 20 сентября 2018 в 20:59

−1

Не расстраивайтесь так. Однажды порядок наведётся сам собой, буквально в течение нескольких лет будет добровольно переписан в сторону оптимизации весь прикладной софт на нашей планете. Появятся также и новые среды разработки, блистающие своей эффективностью и унифицированностью.
Что для этого нужно? Ограничение на оборот мощных процессоров среди простых граждан. Рано или поздно это произойдёт в целях борьбы с терроризмом, когда точно рассчитать и подстроить несчастный случай будет способен какой-нибудь смартфон. И получится, что Айфон с простеньким исполнительным механизмом будет опаснее, чем кило взрывчатки.

minusnaminus 20 сентября 2018 в 20:59

+1

Почитав коменты, мне, не программисту, более всего это напомнило чем-то войны массовых армий. Начало войны: немного подготовленных солдат и офицеров, и их быстро вбивают. Продолжение войны: набор огромных масс новобранцев, резкое проседание качества младших офицеров, больше ошибок и потерь, как следствие — позиционный тупик и кровавые попытки его преодоления. Окончание войны: командование учится пользоваться тем что есть. Мы на второй стадии?

P.S. Пишу с планшета, десяток слов отредактировал взад за автокоррекцией, сколько весит эта

малополезная приблуда — смотреть страшно

Jun06Man 20 сентября 2018 в 20:59

+2

2012 году я участвовал в одном проекте (военная тайна), использовали «голый» компилятор FPC (лицензия свободная, открытый код, кросплатформенный), начали писать библиотеки к проекту, полностью с нуля, используя API Windows и API ядра Linux. Приложение получилось быстрое, имело малый размер. Но это был суший Ад! Мы постоянно исправляли ошибки, добавляли функционал, делали кучу тестов. Получается, большую часть времени, мы тратили на разработку библиотек. Можно было взять уже готовую библиотеку, в которой присутствует требуемый функционал и исключить все страдания. Мы пытались сделать все то, о чем Вы, пишите в своей статье, оптимизировать ВСЕ! Ключевое слово «ПЫТАЛИСЬ», обычным пользователям, было глубоко НАСРАТЬ, на оптимизацию. Конкуренты, успешно «лепили» похожий проект, используя готовые библиотеки (кучу готовых библиотек).

Rohan66 20 сентября 2018 в 21:42

0

А что и как было сертифицировано? Иногда проще провести сертификацию своего «поделия на коленке», чем использовать что-то стандартное, но без сертификата.
Очень забавно выглядит сертифицированная Винда, Офис, ещё что-то, а вот про драйвера и библиотеки (тот же NET) — никто не думает.))))

Jun06Man 21 сентября 2018 в 11:24

+1

Rohan66, у нас была идея и мы писали код, веря в розовые мечты о совершенном. Код проекта остается закрытым.
С 2017 года я программирую на PHP. Я не дума, что есть смысл, каждый раз писать с нуля laravel или Symfony.

Rohan66 21 сентября 2018 в 12:58

0

Ясно. Я думал, что писали под работу с гос. тайной.

Jun06Man 21 сентября 2018 в 15:29

0

Это тихое и мирное ПО, для анализа серверов на Win или Linux :). Мы допустили фатальную ошибку, начав писать с нуля, в релиз пошли пару версий, с серьезной ошибкой в коде. Эти ошибки, в некоторых случаях приводили к неработоспособности всей системы и естественно администрация (сервера) искала и находила причину. У конкурентов дела обстояли лучше.
У FPC открытый исходный код, мы и туда начали лезть и там успели «накосячить».

olehrif 20 сентября 2018 в 20:59

0

В коде очень много ненужных инструкций изначально, неиспользуемых библиотек, функций, веток кода, которые никогда не исполняются.
Существует несколько слоев ПО. Очень грубо можно представить так.
Один слой отвечает за работу железа.
Один слой за операционную систему.
Прикладное ПО имеет свои задачи.
Каждый слой писался отдельными командами в разное время в разных местах разными инструментами.
Мой знакомый мне предложил простое решение.
Решение должно быть… комплексное, когда абсолютно все ПО имеет единую природу и единый источник, единый инструмент. Без слоев.
Код не должен содержать ни единой ненужной строчки. Исполняемый код — не единой ненужной инструкции.
Все ПО должно быть перестроено за счет выстраивания матриц сущность-связь-действие.
Тогда программы будут очень короткие, эффективные, сверхбыстрые, надежные.
Важна будет математика построения связей, построенных на физико-математических принципах.
Фактически, чтобы создавать программы, нужен искусственный интеллект, построенный на физических матрицах, который будет достраивать прикладными задачами единое операционное поле компьютера. И это ПО будет зависеть от конфигурации компьютера. Каждая новая ячейка памяти ОЗУ может быть учтена при работе программы. То есть имеем бесшовный стык — аппаратура-ПО-функции.
Подступы к созданию такого ПО есть.

staticlab 21 сентября 2018 в 00:02

0

> И это ПО будет зависеть от конфигурации компьютера

То есть в ОС компьютера пользователя будет сидеть JIT-компилятор с ИИ, который будет жёстко оптимизировать программу под железо компьютера?



Oplkill 21 сентября 2018 в 12:49

0

Достаточно 1 раза в момент установки и смене железа и всё

olehrif 21 сентября 2018 в 12:57

0

Нет. Задача ИИ создать отделяемое программное обеспечение без слоёв и поддерживать новый функционал. Чистый клей железо-функции с огромной производительностью и надежностью.



PsyHaSTe 21 сентября 2018 в 01:50

0

> В коде очень много ненужных инструкций изначально, неиспользуемых библиотек, функций, веток кода, которые никогда не исполняются.

В каком конкретно коде?

Все компилируемые сегодня приложения собираются с LTO что говорит о том, что каждый скомпилированный байт достижим из точки входа при каком-то наборе параметров.



Ogra 21 сентября 2018 в 06:02

0

В коде библиотек, например. Вот есть в библиотеке foo функция bar, которая не используется ни одним приложением, но в память эта функция все равно грузится.
Какая-нибудь малюсенькая программка на .net может весить 100 килобайт, использовать три десятка функций из гигабайтного фреймворка. Выше на Джоэла уже ссылались.
Дальше — больше. Компилируемые приложения, это хорошо конечно, но сколько у нас сейчас других приложений? На электроне, на питоне, просто со скриптами внутри?
Я даже не знаю, можно ли при разработке на Электроне, вырезать из него ненужный функционал — вебсокеты и вебртц, например. Скорее всего, нет такой возможности. И в итоге, каждая запущенная программа отъест память под эти никогда не используемые куски. При том, что *теоретически*, если нет загрузки джавоскриптта по сети, можно было бы скомпилировать/слинковать приложение без условного вебртц автоматически. Сэкономив чертову тучу ресурсов.



PsyHaSTe 21 сентября 2018 в 13:20

0

Если мы про динамические библиотеки, то это да, но они могут суммарно занимать меньше месте, нежели какой-нибудь libssl будет статически слинкован во все приложения. Гигабайтный фреймворк (кстати, там сейчас сильно меньше, мегабайт 200?) один на всю систему, и его достаточно для покрытия всех потребностей. Не вижу большой беды в том, чтобы оно было в системе.

olehrif 21 сентября 2018 в 12:58

0

На самом деле создание такого программного клея есть частный случай предложенного подхода.

1. Проблема

При автоматизации различных сфер деятельности человека существует проблема перевода прикладных знаний в конкретную информационную систему. Например, в медицине существует около 130 специальностей, для каждой из которых существуют свои термины и подходы к лечению. А человек – один.
В чисто информационных системах существует проблема импорта и интеграции разнородных данных в единую аналитическую систему.
Ситуация напоминает строительство Вавилонской башни строителями, которые не говорят на одном языке.
Фактически любая задача сводится к согласованию языков и накопления знания о решаемом вопросе.

2. Решение
Все построенные человеком системы – языковые. Соответственно, решение вопроса взаимодействия систем нужно искать в кардинальном решении языковых вопросов.
Однако остановвать в этом случае как лингвисты – использовать для решения вопроса взаимодействия языков — недостаточно.
Необходимы знания математические и физические.
Наиболее подходящими оказались матричные вычисления.
Фактически матричные вычисления позволяют представить перевод с языка на язык как перерасчет системы координат.
Кроме того, система понимания должна иметь контекстные знания в области, где нужны решения.
Современные разработки компьютерных интерфейсов могут дать возможность общаться с системой на естественном языке.
Фактически создание системы может выглядеть как диалог специалиста в нужной сфере с системой.
Получив контекстные знания об области и правила обработки, имея возможность через интернет получать дополнительные знания общего свойства (например, погода или законодательство), можно за короткий срок и с минимальным участием специалистов ИТ сгенерировать конкретную прикладную систему для решения конкретных прикладных задач.
Важна здесь именно минимальность усилий. Не команда в 60 человек в течение 10 лет, а команда в 6 человек в течение полугода и менее может запустить в промышленную эксплуатацию систему.
Работу системы будет видеть, как это показано в фантастических фильмах о будущем: герой фильма задает компьютерной системе вопрос на естественном языке, компьютерная система отвечает, предоставляет топологические, мультимедийные и другие материалы и дополняет ответ своей аналитической контекстной информацией. При этом такая система может обрабатывать информацию из любых внешних информационных систем.

olehrif 21 сентября 2018 в 12:59

0

В любом коде, кроме BIOS.



leossnet 20 сентября 2018 в 21:00

+1

Готов подписать практически под каждым утверждением автора за исключением его глубокого пессимизма в отношении будущего ИТ-отрасли. А будущее, на мой взгляд, вполне сравнимо с упомянутой в начале статьи автомобильной отраслью. Разница между этими отраслями заключается лишь в уровне развития инженерных школ в каждой из этих отраслей.

Например, автомобили марок BMW или Mersedes можно легко выделить в огромной массе автомобилей других марок. В ИТ-отрасли же есть только хорошие программные продукты, плохие программные продукты и программные продукты, написанные индусами. Причем программные продукты всех трех категорий могут выпускаться в рамках одной компании.

Наличие же инженерной школы предполагает гарантию минимального уровня качества и функциональности конечного продукта, а также соответствия его уровня цены и его уровня качества. А это достигается путем непрерывного совершенствования в течение десятилетий и столетий культуры производства (в широком смысле) и передачи накопленных знаний и опыта из поколения в поколение. Автомобильная отрасль этот путь уже прошла, у ИТ-отрасли это еще впереди.

minusnaminus 21 сентября 2018 в 19:47

0

Идем на автофорумы и читаем там глубокий пессимизм в отношении будущего автопрома. Движки уже летят сто тысяч километров не отмотав, чтобы заменить свечи нужно чуть ли не половину машины разбирать, электроники понапихали, в некоторых авто до полусотни разных операционных систем воткнуто… Это если по диагонали читать :)

leossnet 21 сентября 2018 в 21:19

0

То, что автомобили выходят из строя после окончания гарантийного срока, является проблемой не инженеров, а владельцев автомобилей. Для автопрома же эта ситуация как раз очень даже хороша, так как обеспечивает регулярное обновление парка автомобильной техники и, соответственно, рост занятости, доходов и прибылей. Замена механических компонентов на электронные также связана исключительно с удешевлением производства при сохранении функциональности, а так же с обеспечением спроса на услуги специализированных автомастерских, так как самостоятельно такой автомобиль сегодня починить просто нереально. Как говориться, ничего личного, только бизнес.

При этом все эти штучки делаются руками высококлассных инженеров. Отсюда можно сделать предположение, что как только качество производства в ИТ-отрасли достигнет уровня автопрома, то программы будут ломаться или тормозить не потому, что плохо написаны, а потому, что в них заложили закладки, требующие обновления до новой версии за определенную плату. Конечно, 5% пользователей смогут их обойти, но остальные 95% обеспечат ИТ-бизнесу вполне приличный уровень дохода.

minusnaminus 21 сентября 2018 в 21:43

0

И в чем тогда разница, гипотетическая, если раньше ПО глючило потому что руки кривые, а потом оно начнет глючить, так как мотивы (с точки зрения потребителя) кривые. В целом, сравнение с автопромом не совсем корректно, так как там наваливать костыли в таком количестве просто нет возможности. Увы, автомобили катаются в материальном мире, и должны мириться с его законами. В синтетических пространствах возможности для маневра много больше. Вы не можете дистанционно ухудшить конструкцию автомобиля, а вот с программой — запросто. Даже не трогая её саму, можно изменить среду в которой она функционирует. До уровня инженерной школы автопрома тут просто не дойдут, возможностей, а значит и соблазнов, сильно больше.

leossnet 21 сентября 2018 в 22:20

0

А разницы и нет. Что касается костылей, то пока программы в основной своей массе не влияют на жизнь и здоровье людей, то и ошибок в них будет по определению много. Но как только кто-то реально пострадает из-за кривого софта или вообще погибнет, после чего исполнитель сядет в тюрьму, а компания заплатит большую страховку и штраф, то качество программного обеспечения в данном виде бизнеса сразу и резко возрастет.

Такой же эффект может быть получен, если в законодательство и судебную практику будут введены нормы, согласно которым любой потребитель программного обеспечения, столкнувшись с ошибками в его работе, может требовать возмещения моральных убытков и упущенной выгоды, кратных стоимости приобретения этого софта.

Но тогда государству и бизнесу придется очень хорошо вложиться в систему подготовки, повышения квалификации и аттестации программистов, что довольно скоро скажется на общем уровне качества программного обеспечения в лучшую сторону. Хотя, конечно, все эти изыски в конечном счете будут переложены на рядовых граждан в цене товаров первой необходимости.

VolCh 23 сентября 2018 в 11:53

0

Из-за софтовых ошибок уже физически страдали и вообще погибали люди. Но максимум денежными компенсациями отделывались компании, насколько я знаю. Скорее всего из-за того, что все требования к процессу разработки были соблюдены, просто роковое стечение обстоятельств. После этих случаев требования для специального софта были ужесточены.

sim-dev 20 сентября 2018 в 21:17

+3

Почему-то не увидел ни одного комментария, который бы как-то однозначно определил причину такого поведения современного ПО. Попробую исправить ситуацию.
Все дело в современной модели бизнеса: **покупай, чтобы выбросить.**
Вместо того, чтобы оптимизировать ПО, чтобы оно летало на нынешнем процессоре, гораздо выгоднее разработать новый процессор. Привычка делать говняное ПО полезна, т.к. позволяет на новом, только вчера появившемся процессоре, создать новую версию старой программы, которая тут же станет тормозить. Круг замыкается.
Если не будет объективных причин, по которым следует обновлять ПО, смартфон, компьютер, автомобиль и т.п., современная модель бизнеса рухнет. Поэтому криворукий программист — двигатель прогресса!
Берегите его, платите ему побольше, и не дай бог он начнет думать, как автор статьи!

Кстати, проблему автомобиля и расхода топлива в рамках парадигмы развития современных ИТ следует понимать чуть иначе, чем тут пытались сделать: не надо уменьшать расход топлива на 100 км пробега, следует сделать стоимость топлива минимальной, и будут брать. Разве не так? Зачем делать ПО, которое станет работать на 640 килобайтах ОЗУ, если можно сделать стоимость 32 гигабайт ОЗУ меньше, чем 640К?!

minusnaminus 20 сентября 2018 в 21:46

0

Дайте вместо мосинки автомат Калашникова. В кого ни будь-то да попадешь :)

Emulyator 20 сентября 2018 в 22:20

+2

Странно, что не увидели комментариев о причинах, они есть. Что касается тезиса «гораздо выгоднее разработать новый процессор, чем оптимизировать ПО», то он подразумевают заговор разработчиков ПО и производителей железа, но есть объяснения попроще. Также я сомневаюсь, что низкое качество ПО объясняется привычками писать плохо, скорее требование выдавать новые версии как из пулемета и просто низкая квалификация не позволяют писать хорошо. Да, модель бизнеса виновата, но она не рухнет, потому что объективных причин для обновления ПО и железа широким массам уже давно не требуется, а нешироким не вносят существенных искажений в модель.)

PsyHaSTe 21 сентября 2018 в 01:54

–2

> Все дело в современной модели бизнеса: покупай, чтобы выбросить.

Нет такой модели.

Я вот пишу программу. Я знаю, что у меня есть некоторые ограничения по производительности. В рамках этих ограничений я стараюсь сделать максимально грамотную и расширяемую архитектуру, решающую задачу в требуемые сроки. Я не закладываю коварно мины if (not(isLatestProcessor) || isAmd) { throttle() }, не пытаюсь внедрить бекдоры для проприетарного софта, никакой менеджер не стоит над ухом и не совращает. Я просто, Пишу. По. Которое решает задачу, которую хочент клиент. За те деньги, что он предлагает. Не больше, не меньше.

Клиент, который покупает, чтобы выбрасить — такое бывает, если это какой-нибудь госзаказчик, которому важно получить бумажку о сдаче для получения финансирования. Но я с такими не работаю.

И все мои знакомые работают по +- такой же схеме.

Вопрос — кто же тогда эти злодеи, которое делают говяное ПО для новых процев?

i7i 20 сентября 2018 в 21:30

0

Я не силен в фронт-енд вебе(писал что-то для бэкэнд, но в основном софт не связанный с вебом). Понимаю что сайты подружают нужные для работы библиотеки, типа нод жс, а почему бы эти библиотеки не устанавливать с браузером изначально? Допустим, при установке браузера загружается набор самых используемых библиотек с самыми используемыми версиями, а потом вместо повторной загрузки библиотек каждым сайтом, браузер уже дает готовое из своей памяти. Или такое уже реализовано и я не один такой «умный»?

HSerg 20 сентября 2018 в 22:30

0

Если на сайтах используются одни и те же js-библиотеки лежащие в CDN, то браузеры будут отдавать их уже из кэша. Но js-библиотек ОЧЕНЬ много, да и разработчики сайтов предпочитают не зависеть от внешних ресурсов.

swelf 21 сентября 2018 в 02:41

0

Размер библиотек вобще не сильно то на что-то влияет, из кэша она, из интернета или еще откуда. А вот интерпретатор/компилятор js, вэб движок в браузере, который все отрисовывает и потребляет в 10ки и сотни больше памяти, чем размер сайта, вот в чем мне кажется проблема.

VolCh 23 сентября 2018 в 12:02

0

Голые они не так уж много потребляют и по сути одинаковое количество для любого сайта. Но одни страницы почти не потребляют дополнительных ресурсов, а другие жрут память и процессор как бешеные из-за содержимого этих страниц, из-за кода написанного так, что он заведомо будет потреблять много ресурсов.

VolCh 23 сентября 2018 в 13:13

0

1) В браузерах есть библиотеки типа DOM и WebAPI, причём написанные на языках типа C++ и Rust. И оптимизации их много внимания уделяется из-за конкуренции браузеров
2) Подгрузка других библиотек, написанных на JS обычно кэшируется, нередко через CDN, то есть абсолютно несвязанные сайты используют общий кэш для них
3) Основное ресурсы для работы веб-приложений идут на парсинг, показ и исполнение кода собственно приложения, на то что пишут веб разработчики, а не на то что пишут разработчики браузеров.

Intenditore 20 сентября 2018 в 21:40

+1

Боги! Вся боль от многолетнего нахождения в безумном мире современного it в одном посте.
Подписываюсь!

Intenditore 20 сентября 2018 в 21:58

0

Тут бы ещё было хорошо упомянуть тему моего недавнего недоумения — USB Type-C, особенно в купе с Thunderbolt. У мира была возможность наконец избавиться от всего разнообразия разъёмов и стандартов, каждого в чём-то неудобного и ограниченного, в пользу одного ультимативного решения. И что вышло?
Разъём один, а спецификаций, реализаций, стандартов — море! И все не совпадают со всеми. Провода — вроде бы все одни, но нет — через один зарядка не идёт, через другой не идёт быстрая зарядка, третий не предназначен для Thunderbolt, четвёртый не работает с v2. А usb-c наушники — вот уж подлинное безумие! Все с проблемами. Телефоны работают с ними всяк по своему. И снова уйма несогласованных стандартов.
В мире технологий могла произойти потрясающая революция. А всё стало ещё ужаснее чем прежде.

CodeRush 20 сентября 2018 в 23:46

+1

Потому что унификация должна проводиться железной рукой и контролироваться на каждом этапе до состояния «не будут брать — отключим газ», а такое возможно только там, где цена ошибки очень высока — у ядерщиков, военных и авиаторов. И там не рынок, а олигополия, и разработка унифицированных продуктов занимает там годы и стоит миллионы.
Рынку же нафиг не нужна унификация, наоборот каждому игроку нужны свои, ни с чем не совместимые и запатентованные разъемы, кабели, периферия, и собственные стандарты вообще на все, иначе прибыль от продажи всех сопутствующих товаров падает.
Кроме того, если вам сделать один кабель USB-C, по которому можно будет передавать и Thunderbolt, и USB 3.1, и питание, и аудио цифровое и аналоговое, и черта лысого оцифрованного — такой кабель будет толщиной в палец и стоить 50 долларов за метр, а покупать его никто не будет, потому что десяток кабелей попроще можно купить вдесятеро дешевле.
Производителю нужно, чтобы его продукт продавался, и он это делает так, как умеет. Хотите совместимости со всем вообще — покупайте только такие продукты, голосуйте рублем и ногами. Нет таких — сделайте и постарайтесь продать. Не вышло — значит это не нужно никому, кроме вас, прошу искреннего пардону.

khim 21 сентября 2018 в 01:24

0

> Кроме того, если вам сделать один кабель USB-C, по которому можно будет передавать и Thunderbolt, и USB 3.1, и питание, и аудио цифровое и аналоговое, и черта лысого оцифрованного — такой кабель будет толщиной в палец и стоить 50 долларов за метр

Вы всё-таки палку-то не перегибайте! Толщина там ни раз не в палец, цена не 50 долларов за метр, а 50 долларов за два. Метровый же стоит $30 (а на распродаже — так вообще $10). Не перегибайте.

> а покупать его никто не будет, потому что десяток кабелей попроще можно купить вдесятеро дешевле.

Покупать его таки будут — но не все и не всегда. Ибо если вам не нужна скорость в 40Gbit/sec — то можно обойтись кабелем попроще. А вот если нужна… никуда не денетесь…

CodeRush 21 сентября 2018 в 01:31

0

Не перегибаю, аналоговое аудио по активным кабелям не работает на том Belkin, я проверял.
Работу VirtualLink не проверял пока, но надеюсь, что выживет без отдельных кабелей.
Опять же, никто не мешает USB IF послезавтра выпустить еще десяток альтернативных режимов для которых понадобятся свои кабеля, и задача сведется к предыдущей.

khim 21 сентября 2018 в 01:42

0

> Не перегибаю, аналоговое аудио по активным кабелям не работает на том Belkin, я проверял.

Так активные Thunderbolt-кабеля и не соотвествуют спецификации USB, почему вдруг по ним должно работать аналоговое аудио?

Пассивный, метровый, кабель — соотвествует спецификации и отлично работает.

> Опять же, никто не мешает USB IF послезавтра выпустить еще десяток альтернативных режимов для которых понадобятся свои кабеля, и задача сведется к предыдущей.

Нет, не сводится. «Полный» кабель с 10 проводами — это максимум, что бывает. А что Thunderbolt с ним не работает — так это не вина USB-C.

То, что вот таким вот кабелем монитор к ноуту подключить нельзя вас же не удивляет? И HDMI Forum вы не пытаетесь в этой «проблеме» обвинить?

CodeRush 21 сентября 2018 в 01:48

0

Мы с вами находимся на одной стороне спора, мне кажется. Я говорил про вот это:

> Провода — вроде бы все одни, но нет — через один зарядка не идёт, через другой не идёт быстрая зарядка, третий не предназначен для Thunderbolt, четвёртый не работает с v2. А usb-c наушники — вот уж подлинное безумие! Все с проблемами. Телефоны работают с ними всяк по своему. И снова уйма несогласованных стандартов.

Даже с учетом того, что физически разъем выглядит одинаково, его по причинам вполне прозаическим не делают совместимым со всеми режимами, альтернативными или нет, и кабелей таких, по которым все вообще режимы работали бы — нет, и не надо. О чем я и написал, а ваших вопросов теперь понять не могу, т.к. это не я хочу один кабель на все случаи жизни.

khim 21 сентября 2018 в 04:35

0

> а ваших вопросов теперь понять не могу, т.к. это не я хочу один кабель на все случаи жизни.

Ну дык он и есть такой. Это метровый Thunderbolt. А двухметровый — он **выходит** за спецификации USB-C и потому, разумеется, с ним несовместим…

CodeRush 21 сентября 2018 в 04:43

0

Дык нет же, аналоговое аудио и по метровому кабелю Belkin не едет (у меня он на столе лежит, я проверял), а едет ли VL — пока еще непонятно. Все остальное — да, т.е. кабель почти универсальный и не толщиной с палец, конечно, но все равно набор из 3 кабелей попроще таки дешевле и большинству пользователей хватит такого набора за глаза.

RuK 20 сентября 2018 в 22:13

0

Когда я сам пишу какое-то приложение, всегда удивляюсь каким быстрым и компактным оно получается на фоне эти монструозных современных программ.
Иногда складывается впечатление, что их искусственно раздувают и замедляют.
(Чему отчасти есть косвенное подтверждение после истории с уменьшением производительности iPhone)

khim 21 сентября 2018 в 01:26

0

> (Чему отчасти есть косвенное подтверждение после истории с уменьшением производительности iPhone)

Ну там-то как раз всё не так просто: его замедляли, чтобы батарейку не напрягать и у хомячков создавалось ощущение, что в iPhone особенная батарейка, которая не теряется за пару лет 30% ёмкости, а отлично работает пять лет…

RuK 21 сентября 2018 в 01:34

0

Ну, не знаю… У меня iPhone 6 Plus, купленный 4 года назад. И он последние месяцы просто жесть как быстро разряжался.
В итоге надоело, потащил его в Apple Store на замену батарейки за $30.
Ребята не стали заморачиваться, выдали такой же но новый, при чём выпущенный всего 5 мес назад, судя по серийнику.
Так он тормозит точно так же, как и его предшественник с мёртвым аккумом.
Вот теперь жду конца октября чтоб поменять на XR или что там нового вышло.))

khim 21 сентября 2018 в 01:45

−1

> И он последние месяцы просто жесть как быстро разряжался.

Ну дык физику не обманешь — литиевая батарейка за 4 года теряет половину ёмкости. То есть телефоны с несменной батареей (а это **почти все** современные телефоны) — заточены под выброс примерно через такой срок… и после этого какие-то «зелёные» устраивают какие-то там акции и собирают дождевую воду? Смешно…

RuK 21 сентября 2018 в 01:48

0

Физику да, но юзеров обмануть пытаются.))
(Зеленые, те ещё чудаки на букву «м»… Ничем кроме шантажа крупных корпораций ради собственной наживы, по сути, не занимаются)

khim 21 сентября 2018 в 01:56

−1

> *(Зеленые, те ещё чудаки на букву «м»… Ничем кроме шантажа крупных корпораций ради*

> *собственной наживы, по сути, не занимаются)*

Всё ещё трагичнее, на самом деле. Подавляющее большинство «рядовых» зелёных искренне верят, что они занимаются «спасением» природы — ибо в математику они не умеют, а чтобы понимать и сравнивать разные вещи против которых они протестуют — так тем более…



RuK 21 сентября 2018 в 01:59

–1

Про рядовых я даже не говорю — в любом движении/партии/религии это просто попки/статисты/фанатики.



celebrate 20 сентября 2018 в 22:26

+3

> Linux намеренно убивает случайные процессы. И всё же это самая популярная серверная ОС.
Не совсем случайные. Ну и предложите, как сделать, чтобы не убивал.

> Вся архитектура баз данных веб/SQL построена на предпосылке (даже надежде), что никто не изменит данные, пока вы смотрите на открытую веб-страницу.
Предложите свой вариант транзакционности в рамках ACID.

> Например, выбирая Hadoop даже если он медленнее, чем выполнение той же задачи на одном десктопном компьютере.
Предложите свой вариант решения задачи, которая ни по одному исходному параметру не помещается в ваш десктопный компьютер.

И так далее. Большинство аргументов просто притянуты за уши и скорее призваны вызвать эмоциональный отклик, а не рациональный.

Вы жалуетесь на мир программ. А вы посмотрите, что в мире творится. Войны, бедность, ложь повсюду, агрессия на каждом шагу, отношения все сложнее сохранять и поддерживать. Приятного мало, да? И тоже самое, мир становится все сложнее и сложнее. Сложные программы кажутся не такой уж значительной частью той сложности, которая нас окружает. Например, лично для меня сложные программы никогда не станут сложнее такого банального вопроса «как вырастить детей». Программы это просто полная фигня по сравнению с этим.

fivehouse 20 сентября 2018 в 22:28

0

> Сегодняшние веб-страницы не будут работать в любом браузере через 10 лет (а может и раньше).

Нет. Утверждение слишком общее, чтобы быть верным. Если веб страницы сверстаны в минимальном HTMLe (даже с минимальными стилями) то они будут работать почти всегда. Я сам иногда захожу на вебсайт сверстанный в далеком 1998(!) году. Автор добавляет вручную странички и вручную корректирует меню. Все работает идеально и быстро.
Просто теперь время жизни верстки (и, соответственно, срок переверстки) стало параметром разработки. Да, к этому мало кто готов. Так пусть будут готовы!
Ну, а с общесистемной точки зрения мы имеет типичное дерево обхода направлений развития. Такое же как и с деревом эволюции. Какое-то направление развития вообще нормально не рождается, какое-то живет 3 дня, какое-то живет 30 лет, а какое-то будет жить относительно вечно. Если важно время системной деградации, то используйте то, что будет жить вечно (еще 50-100 лет). Чистый HTML 1.0, думаю, будет жить еще лет 400 минимум.

vis_inet 21 сентября 2018 в 12:48

0

Можно посмотреть этот сайтик?

НЛО прилетело и опубликовало эту надпись здесь



ValdikSS 21 сентября 2018 в 13:35

0

http://www.realtek.com/



PsyHaSTe 21 сентября 2018 в 15:50

+1

К слову, очень неудобный сайт. Какой-нибудь www.7-zip.org/download.html или www.rust-lang.org/en-US/install.html сильно удобнее.



0xd34df00d 22 сентября 2018 в 07:03

+1

Вот это дизайн, вау!

springimport 24 сентября 2018 в 17:59

0

Нда, у меня современный сайт, на порядок или более тяжелее, загружается за 1-2 сек. А этот грузится больше 5 секунд.
Неужели так сложно поставить nginx…

sumanai 24 сентября 2018 в 18:10

0

> Microsoft-IIS

Да, несколько сложно.

springimport 24 сентября 2018 в 18:30

0

Не знаю его особенностей. Но уверен что перед ним спокойно станет nginx в роли прокси.



PsyHaSTe 25 сентября 2018 в 14:39

0

Уже довольно давно в моде kesterel + nginx reverse proxy, а не IIS. Уж сколько я с ним намучился, особенно со всякими кастомными плагинами, все еще живет блин…



bogolt 22 сентября 2018 в 22:52

+1

lib.ru отличный пример по-моему

vis_inet 23 сентября 2018 в 15:30

0

Да, вся страница открывается меньше секунды!



Arris 21 сентября 2018 в 16:46

0

> Чистый HTML 1.0, думаю, будет жить еще лет 400 минимум.

Только писать на нём будут единицы (процентов от общей массы разработчиков). Примерно как сейчас на ассемблере.



achekalin 20 сентября 2018 в 22:31

0

> элегантные точные шестерни

«elegant precision gears» — это все же «элегантные **прецизионные механизмы**».

Anthony_K 20 сентября 2018 в 22:33

+1

слово «элегантные» тут вообще не уместно. В русском языке элегантным может быть пиджак, но не шестерня.



achekalin 20 сентября 2018 в 22:38

0

Согласен, диковато по-русски звучит, но элегантным может быть инженерное решение — так что я списал бы это. Я-то писал про то, что «precision gears» — это никак не «точные шестерни». От такого «вольного» *невинженерного* перевода в посте, где человек переживат по поводу вполне себе инженерной отрасли (ИТ), у меня чуть кровь из глаз не выступила. Впрочем, это я зря — заметил, что это перевод, а не самостоятельный оригинальный пост. К другим словам в тексте придираться нет желания, просто переводчик немного поторопился.

Upd: Не удержусь: пробел забыли, пробел:

> Кажется, что rm-rf node_modules является неотъемлемой частью...

Пробел перед -rf же!!!

Fracta1L 21 сентября 2018 в 09:30

0

Наверное, подошло бы «изящные». Да и «прецизионные» мне кажется корявым переводом — точнее, это даже не перевод, а тупо калька. Перевод это что-то вроде «изящные, идеально подогнанные шестерни».

novice2001 21 сентября 2018 в 10:31

0

«Прецезионный» — вполне себе общепринятый инженерный термин, как минимум в электронике.

VolCh 23 сентября 2018 в 13:22

0

Скорее из механики, а потом пришло в электронику.



m0Ray 20 сентября 2018 в 23:16

0

Та ж фигня. В институте учили — ни одного лишнего действия программа делать не должна. И это подтверждалось на практике: хреново написанная программа тормозила.

Нынче хреново написанная программа просто не выделяется на фоне остальных.

krab90 20 сентября 2018 в 23:38

+3

Автор оригинальной статьи явно говорит и пишет на русском, но это перевод его статьи с английского на русский. File.zip.rar.zip.



saintbyte 20 сентября 2018 в 23:47

0

Я попробывал современные страницы на ОС 2005 года с мозиллой 1.5 — открывается но очень криво… за 12 лет только протокол HTTP остался почти неизменным.

yarric 20 сентября 2018 в 23:52

+2

> @tveastman: Я каждый день запускаю программу на Python, она выполняется за 1,5 секунды. Я потратил шесть часов и переписал её на Rust, теперь она выполняется за 0,06 секунды. Это ускорение означает, что моё время окупится через 41 год, 24 дня

А если теперь подсчитать затраты времени миллиона пользователей, которые будут пользоваться этой программой каждый день?

ainoneko 21 сентября 2018 в 10:05

+2

«Проблемы индейцев...»?



Antervis 20 сентября 2018 в 23:56

+1

индустрию двигает бизнес. Которому надо не «хорошо», а дешево и вчера, и для которого вычисления на стороне пользователя *ничего не стоят*. Впрочем, я бы не был столь пессимистичен. Работа постоянно ведется во всех направлениях, в т.ч. и для оптимизаций. В конце концов, закон Мура потихоньку сдает, а алгоритмы усложняются всё быстрее.

А на время компиляции жаловаться грех — в конце концов, именно на компилятор перекладываются микроптимизации софта. Те, самые, которые заметны в общем, а не в частном.

oleg_go 21 сентября 2018 в 00:05

+2

Помню лет двадцать пять назад похожие личности выступали за идею всё писать на Ассемблере, не признавая никаких высокоуровных языков. Программа для посадки на Луну как раз на нем и была написана. И 25 лет назад даже Винды как отдельной ОС ещё не существовало.

А вот теперь представьте себе насколько можно убыстрить и оптимизировать код какого-нибудь Web, если написать его на Ассемблере. Правда сколько это будет стоить и с какими траблами в процессе разработки придется столкнутся даже сложно представить.



ValdikSS 21 сентября 2018 в 02:51

+4

Ассемблерные вставки для конкретных архитектур пишут для алгоритмов шифрования, хеширования, сортировки, аудиовидеокодеков, различных сверхбыстрых сетевых стеков, компиляторов, 3D-обработки.

IRainman 22 сентября 2018 в 12:01

0

И то делают это только для тех мест где компилятор сам не осилил сделать правильно. Хорошее продолжение пути это переносить этот ассемблер в компилятор, но это сложно.



mapron 22 сентября 2018 в 17:16

0

В случае с видеокодеками даже не ждут, сможет ли компилятор что-то сделать правильно. Сразу пишут на асме. libx264, libaom, libvpx и прочие ffmpeg



0xd34df00d 22 сентября 2018 в 19:54

+1

Уже перенесли, называется интринсиками (да, вы теряете возможность аллоцировать регистры самостоятельно, но ИМХО это лучше оставить компилятору).

IRainman 23 сентября 2018 в 13:41

0

Интринсики — это и есть асм, только завёрнут в синтаксис языка высокого уровня. Вместо ассемблер-специфичных конструкций код оформляется как вызов inline-функций. Синтаксический сахар, не более. Т. е. это не решает проблему с переносимостью кода вообще.



0xd34df00d 24 сентября 2018 в 07:19

0

Зато, опять же, не надо париться с регистрами и вручную компилятору объяснять, где вы мусорите, где не мусорите.

А как сделать переносимые абстракции над SIMD, которые бы работали для чего-то чуть менее тривиального, чем сложить два вектора — это какой-то ад на графах как минимум.

novice2001 21 сентября 2018 в 10:37

+2

Как раз таки Windows 25 лет назад уже существовала (NT 3.1 — полноценная 32-разрядная ОС)



CodeRush 21 сентября 2018 в 00:18

+3

Согласен с автором, но причин разочаровываться не вижу. Человечество развивает ИТ таким же образом, как и все остальное, в полном соответствии со своей природой. Программы пишутся так же, как строятся города: кем попало, без плана и на руинах от предыдущих попыток. Сложность современных вычислительных систем уже вплотную подбирается (если уже не превысила) порог их понимания человеческим мозгом, и потому единственный путь вперед — повторять за эволюцией, нагромождая слои на слои, и сравнивая результаты.

Автору посоветую пойти и попробовать разобраться досконально хотя бы с современными архитектурами, т.е. как оно там все работает «на самом деле», начиная прямо с собирания сумматора из подручных материалов (как у Петцольда в известной книге), написания процессора простейшего на верилоге и так вверх до ЖаваСкрипта со всеми остановками. Где-то на реализации первых сложных команд в микрокоде придет понимание того, что писать все так, чтобы летало, при этом было доступно вчерашним школьникам, масштабировалось, обновлялось, свистело и пердело — просто настолько умопомрачительно дорого по всем параметрам, что позволить себе это могут либо огромные корпорации, либо гениальные энтузиасты (вроде Беллара). Все остальные берут то, что уже есть, и добавляют к нему то, чего еще нет, но хотелось бы иметь, либо комбинируют то, что уже есть, нужным им способом. Да, медленно, да 10 слоев, да эмуляция виртуализации внутри, да, никто понять не может толком уже давно, но результат — вот он, решения в массы, деньги в кассы.



koldoon 21 сентября 2018 в 00:56

+1

Если небезызвестный парадокс, который формулируется как «Сделать простое — сложно». А бизнесу (как правильно было замечено) — все равно, лишь бы готово было «вчера»: главное — чтобы прибыль приносило.

InGhost 21 сентября 2018 в 01:36

+1

Вспоминается история из собственного опыта. Года три назад обратился ко мне администратор квест-рума с вопросом:
— Нужно написать программу. Игроки по ходу квеста находят флешку, вставляют в комп. Запускается эта программа, в которой вводятся условные данные, и в зависимости от них генерируется ответ, один из которых правильный.

Казалось бы, проще некуда. Но со слов клиента, он обошел нескольких кодеров в моем городе, и те начали ему говорить что-то про «обязательный виндоуз и какой-то дотнет, пару недель на разработку, как минимум столько же на тестирование, бюджет соответствующий».

Этим же вечером был загружен FreeDOS, к ней же FreePascal. Несколько строк кода, и программа уже читала данные из текстового файла с вопросами и ответами. Еще несколько строчек погружали игрока в интерфейс а-ля «матрица» и давали почувствовать себя кулхацкером. Всё это прописалось в autoexec.bat и завернулось на usb-носитель. Клиент доволен, нет необходимости ни в Windows, ни даже в HDD для небольшого компа, собранного где-то на свалке. Жаль, флопа не нашлось :)

А потом я задумался над тем, почему так произошло, но ответ для себя так и не нашел: либо «программисты» попросту забывают о тех элементарных вещах, которые преподавались на первых занятиях по программированию, либо сейчас этому вовсе не учат. И правда, зачем учить элементарным вещам, если достаточно курса «Full Stack Python за месяц», и можно уже искать первые заказы? А потом же этот программер (который не знает основ) пойдет учить новое поколение…

khim 21 сентября 2018 в 01:49

0

> либо сейчас этому вовсе не учат.

Именно так. О существовании DOS'а и о том, что его **всё ещё** можно загрузить на современных компьютерах большинство разработчиков даже не подозревает.

> А потом же этот программер (который не знает основ) пойдет учить новое поколение…

Ну это, всё-таки, не совсем основы, так, история. На машине с UEFI без модуля CSM вы никакой DOS не загрузите…

InGhost 21 сентября 2018 в 02:05

0

> Ну это, всё-таки, не совсем основы, так, история.

На эту тему вспомнилось:

> Народ, не знающий своего прошлого, не имеет будущего.

Rohan66 21 сентября 2018 в 13:07

0

Работаю в колледже. Больше всего проблем (ничего не работает, не могу ввести пароль и прочего) приходит от преподавателей по программированию!))))

kreatr 21 сентября 2018 в 01:37

0

вот так чуть более жизненно:



NIKOSV 21 сентября 2018 в 01:52

+2

Почему бы просто не голосовать рублем? Все те кто «забивают» на оптимизацию вымрут с голода, «олдскульные» разработчики получат больше денег еще больше оптимизаций и софта. Ах да, ведь у вторых зачастую получается глючный, постоянно падающий, неюзабельный с точки зрения функциональности и UX/UI софт, зато быстрый так как 90% времени тратили на оптимизации и издержки из-за выбора низкоуровневого языка и(или) пиление своих велосипедов.

Никто не забивает и не забывает как оптимизировать софт, посмотрите на те же рантаймы джавы и .net, там оптимизация на оптимизации уже позволяющие делать приложения соизмеримыми по производительности с native. Просто с оптимизацией замораниваются выборочно, там, где она нужна, там где плюсов будет больше чем минусов. Я, к примеру, лучше буду серфить сайт «с ненужными красивыми свистоплясками» который открывается за 1 секунду, чем сайтик с видком как из 90-х но открывающийся за 0.1 секунду.

picul 21 сентября 2018 в 03:18

0

> Ах да, ведь у вторых зачастую получается глючный, постоянно падающий, неюзабельный с точки зрения функциональности и UX/UI софт

Да что Вы говорите! Это точно у квалифицированных специалистов такое получается? Ах да, квалифицированный специалист — это же не тот, кто умеет писать нормальный софт, а тот, кто за пару минут выведет 300 байт текста, прогнав 200МБ жабаскрипта.

НЛО прилетело и опубликовало эту надпись здесь

picul 21 сентября 2018 в 12:27

0

А что, запрещено использовать слова, которые не встречаются в предыдущем комменте? Я то думал сейчас как заюзольную свой скудный словарный запас…
Если что, «квалифицированный специалист» употребляется в смысле человека, который хорошо разбирается в экосистеме, в которой работает, и следовательно умеет в оптимизацию.
И проблема не в том, что оптимизация занимает много времени, она на самом деле много времени не занимает. Но вот чего она требует — так это много мозгов и много понимания работы используемого инструментария. Проблема в том, что сейчас модно не голову включать, а выстраивать красивые абстракции на тоннах фреймворков, чем большинство и занимается.
Кстати, хотелось бы пример оптимизированного софта, который невозможно использовать, и взлетевшего не оптимизированного аналога, но это больше к @NIKOSV.

khim 21 сентября 2018 в 03:46

+1

Голосовать рублём не получается. У меня, к примеру, есть версия Nero десятилетней давности. Честно купленная, да. Но она меня устраивает и обновлять её мне **не нужно**. Вот то самое поделие на гиг с лишним мне **не нужно**. По большому счёту, мне хватает бесплатного IMG Burn. ACDSee 3.0 **всё ещё** работает великолепно — и не тормозит. То же самое — с офисом и кучей другого софта. И мне очень жаль, что XV очень тяжело запустить на современных дистрибутивах Linux…

Соответственно люди, которые вроде как бы и хотят нормального софта и делают всё то, что вы говорите… а деньги-то в результате получаются с того кошмара, который производится сейчас! Что побуждает производителей порождать ещё больше… добра.

Интересно — чем вся эта история в конце-концов кончится…

Massacre 21 сентября 2018 в 04:20

0

У меня в своё время тоже был Nero, но перешёл на Alcohol 120%.

novice2001 21 сентября 2018 в 12:47

0

А если красивый сайтик будет загружаться 10 секунд (и так по каждой ссылке), а некрасивый за одну? Да и в целом, красивый сайт вполне можно сделать на html/css безо всяких тормозящих скриптов.

domix32 21 сентября 2018 в 12:50

+1

и прописать в css @media с шрифтом на пару метров

mapron 22 сентября 2018 в 17:18

0

И анимаций в css напихать еще.

domix32 22 сентября 2018 в 20:05

0

Сами анимации довольно дешевы, а вот всякие хитрые селекторы для того чтобы их прикрутить стили могут выливаться в довольно противные проблемы с производительностью.

InterceptorTSK 22 сентября 2018 в 20:22

+1

Точно. Причем шрифт запороть кривой белкой зачем-то. Нет, ну а как иначе? Все же порят шрифты белкой. Причем все 2000 глифов шрифта юзеру отдать) Нет, ну а как? Юзер же должен видеть зимбабвийский, латышский, койсанский и турецкий? Причем даж попытаться выпилить ненужные глифы белкой. Тока белка их выпилит абсолютно криво)) Если открыть такой вот файл шрифта, и поглядеть после белки на такое поделие, в котором килотонны ошибок, то возникает абсолютно адекватное чувство — а чего бы не скинуть термоядерную тактическую боеголовку на сервак белки? Ну а как иначе??
И такое гавно бродит в интернетах массово. Какие-то непонятные шрифты переделанные абсолютно криво какой-то белкой. И эти шрифты естественно криво пристегнуты к страничкам и т.д. и т.п.

Даже термин такой есть — шрифт белочку словил) Ахаха!!! И такое видится в 99% всего веба целиком. Как такое может быть вообще?

Граждане, обычный шрифт даже самый лихой с оставленными русскими-английскими, цыфрами и пунктуацией, с некоторыми таблицами в шрифте типа кернинга и еще кое чего весшает всего 10-15кб на начертание. Начертаний на вашем сайтике не более пяти-семи. Кто говорит что шрифты тормозят — так это он просто в них ничего не понимает.
А БОЛЕЕ ВАМ НИЧЕГО НЕ НУЖНО ОТ ШРИФТА. УЖЕ РАЗБЕРИТЕСЬ В ШРИФТАХ, И ПРИВОДИТЕ СВОИ ШРИФТЫ В БОЖЬЫЙ ВИД. Или даже на это мозгов не хватает совсем?

ОТКРЫВАЕМ СТАНДАРТЫ ШРИФТОВ, ИХ НЕ МНОГО, И РАЗБИРАЕМСЯ. НА ЭТО УЙДЕТ ПОЛ-ГОДА. ЗАТО ВЫ БУДЕТЕ ЗНАТЬ ЧТО ТАКОЕ ШРИФТ.
Или как? Снова будете ждать манну небесную?) Или надеяться на то что вам сервис прикрутят допиленный? Неа) Кто пилил белку — сказочный дебил))) И других не будет, не надейтесь. Бесплатно — значит в любом случае плохо.

nmrulin 21 сентября 2018 в 14:04

+1

А почему же тогда взлетел майнкрафт с графикой 90-х годов (правда скорость работы у него всё-равно современная)?

Mabusius 21 сентября 2018 в 15:03

+2

Новизна же. Я могу ошибаться, но до майнкрафта и ретрографику никто не использовал и ноубрейн песочниц не было.

Neuromantix 21 сентября 2018 в 17:08

+2

Майнкрафт при этом отлично иллюстрирует все описанное в статье. Пик популярности майнкрафта пришелся на 12-14 годы, к нему было написано множество модов, которые позволили держаться этой популярности так долго. Только вместо написания модAPI разработчики выпускали и выпускали бесконечные обновления, которые настолько сильно переписывали игру, что старые моды переставали поддерживаться сразу, а старые миры — через версию-две. В итоге АПи нет до сих пор, гора популярных модов загнулась — авторы их не обновляют, основные поклонники сидят на древних версиях, и лишь небольшая кучка фанатов радостно сходит с ума при каждом обновлении. При этом при каждом обновлении производительность все падает и падает.

domix32 22 сентября 2018 в 20:06

0

Это едва ли не первая 3д песочница же.

tonsky 21 сентября 2018 в 02:22

+3

Спасибо за перевод!

DelphiCowboy 21 сентября 2018 в 05:49

−1

Это всё из-за моды на фреймворки!
Не только не видно и не понятно, что «под капотом», так ещё автоматом тащятся множество зависимостей.

Ogra 21 сентября 2018 в 06:20

0

На мой взгляд, проблема с тормозящим софтом и медленными сайтами была всегда. Абсолютно.
И не связана разработка более быстрого софта с какими-то дикими затратами — можно сделать все то же самое быстрее, меньше и лучше за примерно те же деньги. Проблема в другом.

Просто разработчики очень мало профилируют свой код. Да, они один раз услышали, что «преждевременная оптимизация — корень всех зол, Кнут так сказал», и теперь оптимизируют только тогда, когда все начинает тормозить. На машине разработчика, с i9 и 64Гб оперативки, гигабитным интернетом до сервера в соседней комнате все работает отлично, и *нет повода запустить профайлер*.
Я сейчас говорю даже не об условиях по оптимизации, которые сжирают время и деньги, а о куче мелких решений, которые разработчики всех мастей принимают каждый день. Да, можно было бы написать на массивах, вместо SQL, но это же сэкономит 0.1 секунды, какой смысл? Да, можно при сохранении картинки в Фотошопе нажать «Save for web» и пооптимизировать 10 секунд ради 100 Кб, но какой смысл? В итоге тут потеряли чуть-чуть, тут потеряли самую малость, и *даже не заметили этого*. А вот клиент на своей машине с i3, без SSD и запущенными программами — заметил 3 секунды тормозов.

Иными словами, программы всегда будут немного тормозить.

khim 21 сентября 2018 в 06:51

0

> Да, они один раз услышали, что «преждевременная оптимизация — корень всех зол, Кнут так сказал», и теперь оптимизируют только тогда, когда все начинает тормозить.

Ну дык каждый слышит только то, что хочет услышать. Но когда вы слышите эту фразу — стоит немедленно вспомнить и другую фразу **из той же самой статьи Кнута**: *«В устоявшихся инженерных дисциплинах улучшение на 12%, достижимое без чрезмерных затрат, никогда не считается несущественным и я верю что такая же точка зрения должна быть применена в программировании. Конечно я не буду делать подобные оптимизации в программе, которую я буду запускать один раз. Однако я не хочу ограничиваться инструментами, которые не позволят мне достичь подобного выигрыша»* (вся статья тоже интересна, но этого отрывка достаточно).

Итак: 12% — вот где граница «преждевременной оптимизации». И если вы не можете уложиться в 12% замедления — то это **заведомо** плохой, негодный, неэффективный инструмент. А хороший инструмент — это скорее 2-3%… Много вы знаете инструментов и фреймворков, которые укладываются в этот предел?

Вопрос скорее риторический…

VolCh 23 сентября 2018 в 13:37

+2

Программы будут тормозить ровно до того момента пока метрики времени отклика, реакции и т. п. не станут обязательными в бизнес-требованиях.

НЛО прилетело и опубликовало эту надпись здесь

Ogra 21 сентября 2018 в 06:57

0

Разработчик. i7, 16Gb, гигабит. И ситуации, когда на моем компе все работает, а вот в продакшене начало тормозить, случались. Иногда фиксы делались в одну-две строчки.

periestius 21 сентября 2018 в 07:22

+1

> А зависимости? Люди бездумно ставят переусложнённые «полные пакеты» для простейших проблем, не думая о последствиях

Вспомнил случай из своей практики. Больше 10 лет назад было. У пользователя не работала маленькая утилита, написанная на Fox Pro. И мой руководитель тогда выдал «отличное» решение: установите на том ПК IDE Фокса — это решит все проблемы!

nmrulin 21 сентября 2018 в 14:11

+1

Помню, у дельфистов было поветрие писать базы на BDE, даже когда её можно было реализовать в виде текстовых файлов. В результате, когда BDE накрылось, повезло только тем, у кого был под рукой кодер, который быстро переделывал всё на ADO. Иначе приходилось пользоваться старыми машинами до упора.

gshamshurin 21 сентября 2018 в 07:37

+2

IT-наиболее быстро развивающаяся отрасль, и такие процессы естественны. Пока ты будешь пилить оптимизированное ядро браузера — конкуренты и индустрия накопипэйстят новых фишек, которые ты не успешь реализовать и.т.д. В общем, ты не выпустишь его никогда.

В конструировании железа — я имею в виду не PC hardware, а именно железки, тоже есть подобные вещи. «Это опытная хрень, на один раз и непонятно пойдёт-нет, короче скидай как получится, между основными задачами», говорит тебе босс, и ты (уже ближе к дедлайну) с домашней машины, цепляясь по RDP к рабочей, вяло отмахиваясь от жены и отвлекаясь на поиграть с ребёнком, делаешь неоптимальную конструкцию из большого числа сложных деталей. Потому что на работе никогда — а босс просил, продажники просили, «по-братски». Короче, лишь бы нормально работало и не сильно гемороно собиралось, а от тебя наконец все отвалили — делаешь. Через некоторое время узнаёшь — изделие в серии, клиенты довольны. Причём все, от рабочего и продажника до клиента и твоего директора, видят как можно упростить и удешевить, но… всем пофиг. Работает? Не трогаем. Большой заказ пришёл, а изделие долго производится — так деньги ж есть, фигня делов, ещё станок поставим. Оператор за одним станком следил — будет за двумя следить. Мы ему зарплату накинем, процентов на 10. Потому что эффективность. Это быстрее и проще чем сначала ре-инженирить, а потом договора-спецификации пересогласовывать.

Так и живём. Достаточно большая часть любовно и хорошо сделанных продуктов остаётся в виде концептов, зато самый эпичный трэш который стыдно людям показывать — живёт годами.

CodeRush 21 сентября 2018 в 07:55

+1

У PC HW все точно так же, и за некоторые решения хочется натурально убивать голыми руками, но всех все устраивает, продукт выпущен, все довольны, недовольных шлют по вектору. Работает — не трогай, перестанет — будем посмотреть.

Symsym 21 сентября 2018 в 07:46

0

Костыли (жопа) сначала растёт в голове, а потом уже в коде. Если сначала разобрать конструкцию костылей, то и код проще будет.

razielvamp 21 сентября 2018 в 08:02

0

я бы еще в толку концепцию ci/cd подкинул.
Может мое понимание не правильное, но это скорее маркетинговая фишка нежели техническая.
Исходя из определения это выглядит как постоянный круг загрузки сорса из репозитория, компиляции, автоматизированного стандартного теста и вброса на продакшн на растерзание конечного пользователя = тестера (эффективно и бесплатно)

В итоге имеем постоянные новые версии винды, андроида, мак осей с циклом в пол года. Зачем? На Windows XP один только сервис пак больше года выходил. А сейчас надо выкачать обновление на 4 гига, чтобы убрать пару дыр в протоколе самба, и добавить коновое контекстное меню с иконками в браузере.

На мой взгляд это прямой результат ci/cd, которая гласит, что мы должны постоянно что-то выявливать, важен цикл а не содержание.

В результате имеем Mac OS у которой вместо подсказки пароля отображается сам пароль, после переключения раскладки на азиатские языки к кириллической (арабской) раскладке можно вернуться только после ребута.
На Windows почти не сижу, поэтому таких эпичных фейлов не припомню подробно, но они точно были.
Зато каждые полгода новая ОС!

Производители смартфонов вообще не успевают за андройдом, иногда текущую последнюю версию андроида можно получить только на новой модели смартфона, который анонсирован через два месяца, а в предыдущей уже забили на обновления, и получается что какой-нить гипотетический LG1 работал на 5й версии, а LG2 уже на 7й, а то и 8й. Спрашивается, зачем вообще была нужна 6ая? Для пары гуглофонов?

В общем, согласен со всем написанным за исключением сравнения размеров. Все-таки в 30ти мегабайтном windows 95 были «скучные обои» и палитра в 256 цветов. И с этим ничего не поделать, в современных мониторах и смартфонах разрешение экрана растет, поэтому и картинки/кнопки надо делать «веселее», ярче и больше. Ну и поддержка оборудования тоже важна, вряд ли Windows 95 поддерживал такое же

количество принтеров как современная ОС.

Losted 21 сентября 2018 в 17:21

+1

CI/CD, на мой взгляд, это не обязательно отсутствие тестов (можно ж и ручной этап делать). Это скорее про автоматизацию и повторяемость процессов от сборки до выката. Это про то, что у окружения все одинаковые, а не немного уникальны по-своему (частое последствие работы руками). Про то что артефакт приложения всегда именно тот, что тестировался, а не немного уникальные версии «от Маши» и «от Пети» из-за грязной сборки на дев машине.

vs74 21 сентября 2018 в 08:02

−1

Согласен с автором полностью. Вообще стыдно за ИТ, стыдно.

Arris 21 сентября 2018 в 08:08

0

Вы думаете, что закон Мура перестал работать? Нет, не перестал. Просто включился закон антимура: **«Каждые два года количество багов удваивается»**

semmaxim 21 сентября 2018 в 08:15

+1

Что-то меня прям проняло… Пойду изучать rust и писать под него либы.

lmp59 21 сентября 2018 в 09:28

+2

Посмотрите какие компы сейчас у разработчиков, по 2-3 монитора и ОЗУ 16гигов это считается тормозом, а нужно им выдавать что-нибудь из 90-х годов, тогда сразу при отладке-тестировании будет видно как медленно работает программа. И поневоле начнешь писать чтоб хоть не беситься дожидаясь когда программа отработает. А потом пользователи могут запускать и на современных компах.

Simplevolk 21 сентября 2018 в 16:13

+1

Чтобы у них кроме прог еще и IDE+окружение тормозило.

PashaNedved 22 сентября 2018 в 16:09

+1

На студию неоднократно выпускались обновления, при которых начинались тормоза даже на топовом железе. И вообще, vs ide 17 лагает и крашится на железе образца 2012 года

IRainman 22 сентября 2018 в 11:18

+2

Нет не надо. На компах разработчиков тоже тормозное ПО и тормозит оно дико. Вот для тестовых машин таки да крайне необходимо использовать слабые конфигурации просто обязательно!

VolCh 23 сентября 2018 в 14:22

0

Просто количественные требования к производительности должны быть зафиксированы в ТЗ для их формальной верификации при приёмке

hzs 21 сентября 2018 в 09:08

0

Тоже очень бесит что современный софт работает дико медленно и при этом жрёт дофига ресурсов.
Сын в 4,5 года попросил меня дать ему железо, чтобы он сам мог сам собрать себе компьютер.
Я люблю старое железо и выделил ребёнку железяки примерно 2000 года, он сам собрал на стенде свой первый комп, я только поставил ему на комп 98 винду.
Конфигурация что-то вроде Celeron 633 мгц, 384 мб оперативки, весь софт того времени просто летает.
Сейчас браузер отжирает несколько гигов оперативки на несколько вкладок, это просто писец, товарищи!
Ради интереса, посмотрел сколько весит мой сайт: Вес самого шаблона 120,5 килобайт, при загрузке в браузере со всеми ресурсами движка: 859 килобайт, что, я считаю, многовато. Время полной загрузки страницы без наличия кеша: 2,665 секунды.

НЛО прилетело и опубликовало эту надпись здесь

NIKOSV 21 сентября 2018 в 09:41

0

Ну так прям медленнее. Сегодняшний софт на сегодняшних компах за 1500 баксов работает в разы шустрее чем старый софт тогда на компах за 1000 баксов (1500 баксов сейчас с учетом инфляции). И это при том, что сегодняшний софт в разы красивее и функциональнее. Та же винда тогда более-менее сносно работала на десктопах от среднего и выше, а сейчас 10ка летает даже на бюджетных ноутах за 1000 баксов. Да софт сейчас более высокоуровневый и менее оптимизирован, но who cares если за счет железа производительность в итоге все-равно растет.

Neikist 21 сентября 2018 в 10:06

+1

> бюджетных ноутах за 1000 баксов

Мне кажется или это оксюморон? У меня комп стоит лишь чуть дороже и я его мощным довольно считаю. А уж EDC ноут и вовсе за 250$ примерно брался.

Rohan66 21 сентября 2018 в 13:18

0

> а сейчас 10ка летает даже на бюджетных ноутах

А Вы в 10-ке работаете? У меня, например, разный софт установлен и мне важна его производительность, а не «летающая 10-ка»! Я рабочий стол вижу только тогда, когда на «летающую 10-ку» обнова приходит (да будут они прокляты!).

NIKOSV 22 сентября 2018 в 04:01

0

С софтом та же история, сегодняшний софт на сегодняшнем железе работает быстрее чем старый софт на старом железе, и это при том, что новый софт красивее и функциональнее.

daniilk 21 сентября 2018 в 09:27

+1

Автор подменяет понятие эффективности.
ПО является продуктом коллективного разума, что означает, что каждый конкретный программист находится в определенных условиях, а следовательно на его решения влияет главным образом среда в которой он оказался. А среда стремится сделать все как можно быстрее. Разработкой руководят процессы которые быстрее всего позволяют выпустить продукт на рынок. А процессы которые бы позволяли бы создавать компактный, быстрый код, да еще штатом в тысячи людей пока еще не создали, да и столько хороших программистов нет.

Osnovjansky 26 сентября 2018 в 00:43

0

Windows 2000 делалась вовсе не сверхчеловеками и была вполне быстрой, для железа выше среднего (для железа ниже среднего была Win95). Эту систему писали вовсе не сверхлюди — обычных программистов, я думаю там было больше половины.
Так что, если кому-то приходится работать над говнопродуктом — это его выбор. И не надо придумывать оправданий.

daniilk 26 сентября 2018 в 00:55

0

Разговоры про убогость Windows ходят с начала ее создания. Да и в 1995 языку c++ только 12 лет, так что рынок IT он мизерный, и следовательно программистов было меньше, и лучше. А сейчас любой крупный банк имеет штат IT по бюджету сопоставимый с бюджетом некторых стран. Надо смотреть контекст. IT это промышленость, массовое производство с вытикающими отсюда последствиями.

Iwanowsky 21 сентября 2018 в 09:27

−1

Согласен со всеми доводами автора. ООП, фреймворки и визуальное программирование убили искусство программирования. Если раньше программировали на C/C++, Паскале и пр. (порой, с использованием ассемблерных вставок), то сейчас, используя библиотеки готовых объектов (классов), накидав формы и объекты в визуальной среде программирования, легко и быстро написать программу любой сложности. И если раньше программный код даже растягивали по строкам при оплате по количеству строк программы, то сейчас программы могут иметь миллионы и десятки миллионов строк кода: MC Виндовс 7 скомпилирован из примерно 40млн строк кода (Вин3.1 — 2,5млн), MC Офис 2013 — 45млн строк, Линукс 3.10 — 13млн, Андроид — 12млн, Файрфокс — 18млн, Гугл Хром — 7млн, новый Фотошоп — 10млн (Фотошоп 1.0 — всего 128тыс.!); а управляющее ПО самых современных автомобилей может иметь и 100млн кода (т.е. производители просто накидывают готовые программные блоки, и никакой оптимизации кода). Любая универсализация и унификация программирования в виде ООП и фреймворков подразумевает огромную избыточность кода. Большие объемы устаревшего и неиспользуемого кода накапливаются в любых крупных проектах (как проприетарных, так и опенсорсных) с выходом все новых версий, негативно влияя на производительность и отзывчивость программ; и периодически программисты проводят чистку этого кода и АПИ, ускоряя работу программ, что потом ставят в достоинства вышедшей новой версии ПО. Те же Демки со сверхсложными алгоритмами генерации изображений занимают всего 1024, 256, 128 и даже 64 байт (по условиям конкурсных программ).
То же самое с версткой сайтов и веб-программированием. Изначально все сайты выглядели минималистично и просто (наподобие еще более старого Гофер), а сейчас даже минималистичная версия сайта Яндекса (ya.ru) — это почти 50КБ кода; при том что печатная страница простого текста (если ввести без стилевого оформления в Блокноте) — примерно 2КБ).

InterceptorTSK 21 сентября 2018 в 09:56

0

А нынче и не может быть искусства программирования. Потому что оно уже запорото все на уровне опирацыонок насмерть. Уже побороть почти ничего не получается… Смотрите на пальцах как оно происходит, да и скорее всего вы и сами знаете.

Собрали не так уж и давно либу под себя. Смысл следующий. Надо что бы шрифты рендерились даже на вин2к, но шрифты начертаний семи-болд, ультра-лайт и т.д.
Вин2к вообще такого не знает, ибо gdi тупо вообще не знает таких начертаний в принципе. Ну мы перепилили и шрифты) Подменили в самих шрифтах идентификаторы начертаний и оно поехало. Заодно выпилили лишние глифы, да и вообще шрифты переколупали знатно, ибо багов в них еще больше чем в ПО. А что толку?) Шрифты в винде ставятся в вёнду, и нужны права одмина. Потому что какой-то казюп из майкрософта собрал текстбокс как обертку на древнючем компоненте, которому 20 лет как почти. Смысл в том, что нельзя из своего ПО рендерить в етом текстбоксе своим шрифтом) Проверьте)) Можно, только если шрифт установлен в систему.
Как это решилось? Насильно закинули изначально в gdi замаскированный шрифт, ну семи-болд же gdi не видит и не хочет видеть, затем т.к. шрифт уже в gdi, то создаем уже шрифт из фреймворка, и его нет в шрифтах системы) Ну и фигачьте спрайтами какие хотите строчки. Это самый простой и железобетонно работающий вариант. Все остальное полное гавно, с которым придется пониметь неиллюзорный бдсм-секс, или оно работать не будет вообще никак.

Да что тут говорить, кто пишет разбираясь в том что пишет — так же трахается каждый день…

InterceptorTSK 23 сентября 2018 в 11:15

0

Кстати — искусство основано на чувствах. Искусство — это способ передачи чувств. Не более того. Искусство и программирование — вещи не совместные никак. Ибо программирование полностью лишено чувств напрочь. Расчет и только расчет. Ничего другого быть не может.

micro-CMS 21 сентября 2018 в 09:27

+1

Согласен полностью. Яндекс и Гугл хотят от нас скорости сайта, а их же страницы на мобильнике не открыть.

freuser 26 сентября 2018 в 11:57

0

Была у меня хохма с яндекс-картами:
открываю на кнопочной нокии m.yandex.ru. Работает. Открываю по аналогии m.maps.yandex.ru. Страница отсутствует. Ладно, пытаюсь войти на взрослую версию maps.yandex.ru. Та-дам, срабатывает редирект по user-agent и меня перекидывает на m.maps.yandex.ru! Всё, приплыли.
Сообщил в техподдержку, на первой линии меня не поняли (я так понял, саппортер проверял на своем смарте, который не перекидывало на мобильную версию карт). После третьего однотипного ответа забил.

ledocool 21 сентября 2018 в 09:27

0

>Тем не менее, каждый второй сайт не может обеспечить плавную прокрутку страницы на 60 FPS на последнем топовом MacBook Pro.

Это из-за javascript апплетов, управляющих прокруткой. Да спасут тебя свидетели noscript. Очень заметно на странице google, кст.

Fastman 21 сентября 2018 в 09:27

+1

— Нет проблем технических. Есть проблемы с людьми. Они такие какие есть.
И получают ровно то чего хотят.
— Прийти покритиковать, объявит манфест «мы за все хорошее против всей х*йни»
это конечно же решение. Да.

AndrewStr 21 сентября 2018 в 09:27

+5

Я конечно согласен в чём-то с автором, но решил проверить сам, как он сам следует тому, о чем пишет здесь. Пройдя по ссылке в его блоге tonsky.me/projects, я заметил что эта абсолютно статическая страница сразу вытягивает 2 мегабайта! А всё потому-что JPG'и не сжаты правильно для веба. Почему же автор, если он фанат такого перфекционизма, не довел до перфекционизма хотя бы свой собственный блог?
Ответ простой, либо не подумал, либо лень, либо подумал, ну, и так сойдет.
ИМХО, вот эти 3 либо и «и так сойдет» и есть причина того, что мы сейчас наблюдаем.

arvitaly 21 сентября 2018 в 09:32

+1

Я понимаю боль автора, но хочу заметить, что категоричное отнесение создания софта к чистой инженерно-математической науке не имеет никакого смысла. В первую очередь, это прикладной инструмент для общечеловеческой деятельности, в абсолютно каждой ее сфере, в малейших приспособлениях и крупнейших проектах.
Сетовать на то, что инструменты для использования миллиардами, созданные за пару десятилетий формально несовершенны, по-детски максимализировано.
Хотя, я понимаю, почему именно сейчас это становится так заметно, закон Мура не оправдал себя и, если раньше, языки и технологии тупо не успевали за новыми мощностями, нужно было успевать хоть что-то, хоть как-то, то теперь в железе мы, временно, остановились, а избавиться от такого количества legacy и переписать его на новый лад выгодно лишь тем, кто использует мощности по полной, остальным никто (пользователи) за это не заплатит. В общем-то ничего страшного, никаких критичных функций люди на айфоне не делают, чтобы переживать из-за скорости или качества, поэтому и оплачивать рефакторинг не будут. Да и новые платформы не за горами, пора уже о них думать)

VadimRublev 21 сентября 2018 в 10:02

0

Надо сделать робота-программиста.
Хотя бы (для начала) для десктопного веба.

Romario21 21 сентября 2018 в 10:07

+4

Проблема в том, что программирование в настоящее время больше похоже на хайп.
Все соревнуются, кто больше новомодного и крутого запихает в проект, на выходе имеем как правило простецкое приложение с парой кнопок, но зато под капотом термоядерный реактор из всего хайпового и современного.
И еще:
Куча школьников + куча менеджеров = однокнопочное приложение размером 250мб.

kleeen 21 сентября 2018 в 10:07

0

Не корректно думать, что затраты на добавление символа в ворде и в лексиконе соизмеримы.
Надо просто сесть и написать с нуля, например проверку орфографии. Чтобы четка отдавать себе отчет, что сколько ресурсов требует. А есть еще всякие сглаживание, форматирование, оглавление и т.п. И все это любит есть ресурсы. А пользователи уже привыкли, что все происходит на лету.
Хотя конкретно с WEB -согласен. Зачастую там наблюдается победа дизайна над здравым смыслом. А было же правило не больше 30 килобайт на страницу =))

Massacre 21 сентября 2018 в 14:02

0

Проверка орфографии и прочее форматирование не очень подходят, как пример, это всё отлично умели и редакторы под DOS :) Тот же известный в определённых кругах редактор «Слово и Дело».

А в плане JS сейчас влияет не объём кода, а результат его исполнения.

yusman 21 сентября 2018 в 10:10

+3

Как бывший инженер ДВС, категорически не согласен с тем что современные авто достигли предела 98%. На мой взгляд авто самый не эффективный, костыльный и опасный способ перемещения. Каждую секунду на нашей планете в ДТП погибают люди, только подумайте, каждое мгновение и мы считаем это нормой, мы привыкли к этому! Мы сжигаем сотни тонн топлива ежесекундно загрязняя планету. И это только потому что мы в определенный момент история выбрали такое вот странное и нерациональное решение. Да современный автомобиль это инженерное чудо, но построенно поверх большого и толстого и не эффективного костыля, от которого сложно избавиться.
Поверх этого костыля мы строим не удобную инфраструктуру, все для автомобиля, парковки вместо игровых площадок для детей, дороги вместо велодорожек и тротуаров. С кодом ситуация совершенно другая, современный код хоть и не эффективен, но людей не убивает. На экологию влияет не сильно, да и жизнь делает лучше). Хотя сравнивать IT и другие инженерные искуства на мой взляд не совсем корректно

IvanTamerlan 21 сентября 2018 в 11:27

0

> С кодом ситуация совершенно другая, современный код хоть и не эффективен, но людей не убивает.

Код очень даже убивает. Есть еще ошибки автопилотов (у них сложный и непонятный ИИ с нейросетями), хотя вот этот автопилот отработал на куда более скромных системах без всяких нейросетей и фреймворков.
Даже на хабре есть статья с перечислениями фатальных ошибок.

GrigorGrl 21 сентября 2018 в 18:30

0

> Каждую секунду на нашей планете в ДТП погибают люди

Ошибаетесь примерно на два порядка.

Пример — в 2015ом умерло 1.3 млн людей. В году 3600*24*365 секунд. 31 536 000/1300000 = 24 секунды. 24 секунды в среднем чтобы умер 1 человек, когда вы утверждаете что каждую секунду погибают люди (во множественном числе).

Учитывая что машины несут пользу, в том числе и скорая помощь, не выглядит так страшно.

Osnovjansky 26 сентября 2018 в 00:56

0

—

Dee3 21 сентября 2018 в 10:23

-1

Чтобы двигаться вперед и делать новое по-новому, нужно оказываться от старого.
Прежде всего от старых убеждений.

Я подозреваю что очень у небольшого количества инженеров, программистов, а тем более команд разработки или компаний достаточно крепкие яйца для этого.

Посмотрите сколько возни поднимает отказ от физических кнопок или 3.5 джека в смартфоне.

ledocool 21 сентября 2018 в 12:17

+4

Вот лично вас хочу поблагодарить за сбитые об экран пальцы и невозможность слушать музыку при зарядке.

Чтобы отказаться от старого надо чтобы новое было как минимум не хуже, а у нас мозгов нет.

Dee3 21 сентября 2018 в 13:16

0

Никакого «не хуже» не будет без принудиловки, потому что не зачем делать новые технологии если спрос есть только на старые. Более того это экономически не выгодно — делать и продавать вещи, о которых люди не знают что они им нужны.

Да в общем то все что вы видите вокруг себя базируется на людях, кто не боялся двигать и продвигать новое.

Если представить что через десять лет все аудио будет беспроводным и его стоимость сравнится с аналоговым, то глядя назад в прошлое будет смешно читать жалобы про «ну вот они сделали ~~компьютер без Floppy~~ телефон без 3.5 джека.

ledocool 21 сентября 2018 в 20:08

+1

> >Чтобы двигаться вперед и делать новое по-новому, нужно оказываться от старого.
> >Никакого «не хуже» не будет без принудиловки, потому что не зачем делать новые технологии если спрос есть только на старые.

Я, наверное, читаю как-то не так, но вы реально считаете, что прогресс пойдет только под принудиловкой? Впрочем, неудивительный вывод если принимать во внимание, что вы считаете деградацию от Apple прогрессом.

А если вы не согласны с такой формулировкой, то вот вам аналогия: Эппл для того, чтобы все влезали в их супермоднявые штаны отрезала всем мужчинам-клиентам половой член. Выбора не оставила. А другие производители начали подражать, но правда большинство все-таки делало обрезание или удялало одно яичко. Теперь клиентам эппл чтобы заняться этим самым нужно купить патентованый агрегат, который делает все то же самое, только хуже — оргазм не достигается.
Прогресс, твою налево.

В случае с флопиками, кстати, совершенно не аналогия — флопики вполне естественно вытеснились сидюками, которые могли тупо больше. Сами пользователи имели возможность сделать выбор. И выбор пал на то, что было объективно лучше, больше, быстрее.

VolCh 23 сентября 2018 в 15:12

0

Пример нового без принудиловки: электромобили. Не по всем характеристикам они лучше, в чем-то объективно хуже как сами, так и экосистема, но продажи растут без принудиловки, выбор какой мобиль купить есть вроде во всех категориях, а где нет — это электромобилей ещё нет.

Другой вариант FOSS: можно выбирать любую версию для установки, разве что уж совсем древние поискать придётся без гарантии, что найдёшь. А попробуйте легально сейчас установить на совместимый комп windows 98+ office 97 + ie 6. Даже win8 и office 2012 сейчас легально установить непросто.

SirEdvin 23 сентября 2018 в 15:26

+1

> Пример нового без принудиловки: электромобили.

У них вроде есть налоговые дотации.

PsyHaSTe 23 сентября 2018 в 16:56

0

Потому что сложившиеся цеопчки очевидно будут дешевле, и для конкурентноспособности на первых порах нужно помогать.

khim 23 сентября 2018 в 18:25

0

Это, может быть, и правильно, но когда вы говорите: либо вы выпускаете XX% электромобилей, либо нарвётесь на штрафы — то это максимально далёкий от «примера нового без принудиловки» варинт.

PsyHaSTe 23 сентября 2018 в 22:44

0

Квоты это плохо.

khim 24 сентября 2018 в 02:28

0

Тем не менее именно с их помощью продвигают электромобили и Тесла получает с их помощью дополнительные субсидии — не слишком большие, но заметные.

Так что не надо про этот «пример нового без принудиловки» — принудиловка там огого какая. Хорошо это или плохо — другой вопрос.

PsyHaSTe 24 сентября 2018 в 10:52

0

Дотации это хорошо. Квоты это плохо. Это моё мнение по любому вопросу. Включая квоты на женщин, мексиканцев и марсиан в университетах, и многие другие применения.

ledocool 27 сентября 2018 в 10:20

0

Не принудиловка, а побудиловка. Это разные на вкус фломастеры.

Принудиловка это «быстра аффтамабиль сделал мне, а то ша как дам».
А побудиловка это «сделаешь машинку — получишь конфетку (дотации)».

khim 27 сентября 2018 в 18:55

0

Конкретно в случае с электромобилями есть и то и другое…

ledocool 27 сентября 2018 в 20:32

0

Про побудиловку уже сказали. А в чем принудиловка то? Государство угрожает угнать теслу, если тесла не сделает автомобиль-трансформер в 2020?

khim 27 сентября 2018 в 21:11

0

Не «автомобиль-трансформер», а «ZEVs», «advanced PZEVs» и «PZEVs». И да — можно нарваться на разного рода санкции если вы их продаёте меньше некоторого процента.

Тесла, в частности, зарабатывает на том, что помогает обычным производителям откупиться (грубо говоря им позволяют за денежку защитать автомобили, произведённые Тесла в счёт «ZEVs», «advanced PZEVs» и «PZEVs», которые должны были быть произведены всякими GM'ами и Ford'ами).

Но это временное решение, пока разрешено производить 90% обычных автомобилей. Если этот процент будет уменьшен до 70% или 50% (а планы такие есть), то «откупаться» встанет слишком дорого.

Am0ralist 21 сентября 2018 в 23:50

0

> Посмотрите сколько возни поднимает отказ от физических кнопок или 3.5 джека в смартфоне.

правильно, даешь блютуз наушники и еще больше аккумуляторов, которые дохнут каждые пару лет, вынуждая покупать новые гарнитуры и еще больше захламлять планету.

intet 21 сентября 2018 в 10:30

0

Как видно мы сегодня страдаем от огромного числа уровней абстракций. Значительная часть абстракций появилась из-за того, что компоненты которые хотелось было бы использовать не выполняли свои функции или требовали через мерного погружения в себя отвлекая от непосредственного процесса разработки.
Если бы системные программисты написали бы более полные и простые библиотеки, то можно было бы выкинуть огромные число оберток и пользоваться ими.

Пример с которым я с толкнулся. Надо было добавить на страницу таблицу из картинок выровненных по сетке. Что хотелось бы — компонент на вход принимающий массив ссылок/файлов картинок и занимающийся всем самостоятельно.
В реальности же готового варианта нет. Да отобразить картинки возможно, но при наличие необходимости хоть как-то изменить список без ручных манипуляций с dom/использованием библиотек не обойтись. Добавление функции сортировки drag'n'drop это горы js скриптов.

eirnym 21 сентября 2018 в 10:42

+2

Приложение Facebook для iOS весит более 512 мб, большая часть из которого — фреймворки с названиями вроде "rarely used framework".
Приложение Facebook для Android — 70 мб
Приложение Facebook Lite ~5 мб… ну вы поняли



lwanowsky 21 сентября 2018 в 10:56

+1

Работающий ФейсБук на Андроиде нещадно жрет батарею. Останавливаешь его в настройках Андроида — и нормальное энергопотребление смартфона восстанавливается.

eirnym 21 сентября 2018 в 10:58

0

Он такой на всех устройствах, особенно, если в ленте много графики или видео. Их я предпочитаю смотреть на компьютере, потому что иначе проигрыватель убивается, но неизвестно когда…



serbod 21 сентября 2018 в 10:59

0

Lite недавно испортили. Теперь там мелкий шрифт, обрезаны строки, видео и картинки на весь экран не показывает.

Hashinger 21 сентября 2018 в 11:26

0

А ведь когда-то смеялись над Nero.

p_fox 21 сентября 2018 в 11:00

0

Мне вот тоже интересно — почему Винда (7) так долго обновляется?
Накопилось обновлений на пару сотен метров, винда перезагрузилась и ПОЛЧАСА писала мне про обновление системы.
ЧТО она делала полчаса? Или это было 200 метров кода, который Винда компилировала прямо у меня?

eirnym 21 сентября 2018 в 11:02

0

wsusoffline делает ту же самую работу через тот же интерфейс установки обновлений, но *намного* быстрее!

DMGarikk 21 сентября 2018 в 11:21

+1

у винды огромная обратная совместимость, и она буквально на каждый апдейт сканирует систему в поисках программ с ним несовместимых и запускает кучу всяких миграций на миллиард вариаций конфигурирования системы
p.s. не верю конечно что это нельзя было бы сделать быстрее

ledocool 21 сентября 2018 в 12:16

0

Бэкапила ваш hdd на сервера мелкософта.

user343 21 сентября 2018 в 16:01

0

Тогда бы логи роутера это показали. Скорее всего тучу отладочных логов пишет паралельно ("индусский код"), а HDD не столь многопоточный.

ledocool 22 сентября 2018 в 08:48

0

Это шутка, не парься. Думаю, даже Гейтс не знает что его винда делает во время обновлений и ПОЧЕМУ ТАК ДОЛГО!!!!??? Скорее всего такая хрень происходит из-за того, что процесс непрозрачный и никто не видит явных проблем в его реализации. В лине ж как-то debian дистрибутивы умудряются делать full-upgrade за меньшее время и с меньшим числом боли.



ValdikSS 22 сентября 2018 в 09:57

0

> и с меньшим числом боли

Ну уж нет, вылет программ из-за обновления библиотек в Linux — типичная ситуация. В Debian-based дистрибутивах нет транзакций и автоматического отката обновлений в случае каких-либо проблем. Чуть-что-пошло не так — всё, apt/dpkg сломан, нужно исправлять вручную.
С RPM-based все лучше, есть транзакции и откаты, но и аналог dist-upgrade может происходить полчаса-час, и при обычных обновлениях программы так же вылетают.

Все проверки, которые делает Windows, оправданны.

ledocool 22 сентября 2018 в 10:42

0

А в винде принципиально такого быть не может, т.к. каждое приложение тянет свои библиотеки, поэтому опять же не понятно на кой все эти проверки на over9000 часов.

А касательно debian-based дистрибов — у меня деба не падала. Даже у убунты ничего не ломалось до тех пор, пока я случайно доступ к /tmp не запретил. (Игрался с chmod и опечатался.)

IRainman 22 сентября 2018 в 11:03

0

> А в винде принципиально такого быть не может, т.к. каждое приложение тянет свои библиотеки

Нет. Посмотрите как ставятся браузеры, например, на Linux дистрибы, там тоже в довеске много чего лежит потому что так меньше вероятность глюков. Кроме того тут речь идёт в основном про компоненты ОС.

ledocool 23 сентября 2018 в 16:03

0

А как это опровергает то, что я сказал про венду?



0xd34df00d 24 сентября 2018 в 07:20

0

Куда посмотреть в случае firefox или chromium, собираемых на моей машине?

Более того, чувствую, с firefox так вообще везде.

IRainman 22 сентября 2018 в 11:01

-1

Поддержу @ValdikSS ибо меня боль с ручным разгребанием в Ubuntu server так достала, что я обновляюсь там просто образом, т. е. тестирую апдейт и апгрейт на резерве, а потом если всё прошло хорошо просто выкатываю образ поверх рабочего ничего не трогая. С Windows вообще никакого секса, если апдейт не встал, то происходит автоматический откат и всё загружается в работающую систему где ничего не сломалось.

ledocool 23 сентября 2018 в 16:05

0

Спорить не буду — нет опыта. Но вот вопрос из интереса: а не помогает сделать апдейтовые скрипты?

IRainman 23 сентября 2018 в 16:55

0

Какие именно апдейтовые скрипты имеются ввиду? В этом плане оно и так автоматизировано более или менее.
Создаётся образ того что работает.
Пробуется обновить всё включая ядро с полной зачисткой после.
Если сфейлилось, то результат удаляется и пробуется обновиться только пакетам без ядер с полной зачисткой на исходный образ.
Если сфейлилось, то всё. Если получено более менее живое, оно загрузилось и нудное запустилось, то производится тест в живую, если работает как надо и в конфигах нигде не наступила жопа, то образ целиком выкатывается вместо того что работало и система ребутится. Если что то пошло не так, то ещё месяц ничего не трогается.
Альтернатив я не знаю, возможно тоже нет опыта в этой области, вероятно с rpm будет лучше но в Ubuntu Server у меня без геморроя получается только так. Чинить и разбираться в глюках вообще не вариант ибо абсолютно бесполезное красноглазие там где должно просто работать. Я бы с радостью на всё это нанял админа, но средств на это нет.

ledocool 23 сентября 2018 в 20:06

0

> Какие именно апдейтовые скрипты имеются ввиду?

Да чет установка генты вспомнилась. Там вот как раз скриптами можно было бы.

> Альтернатив я не знаю, возможно тоже нет опыта в этой области, вероятно с rpm будет лучше но в Ubuntu Server у меня без геморроя получается только так.

Есть еще одна мысль, что можно попробовать. Ubuntu это нестабильная деба. От слова совсем нетабильная. У дебы от убунты до stable проходит три релиза. Если запаздывание по софту не парит, можно попробовать поставить именно debian stable.

IRainman 27 сентября 2018 в 14:54

0

Ну я с тем же соображением использую Ubuntu LTS

ledocool 27 сентября 2018 в 20:31

0

Из того, что я слышал, даже у LTS убунты стабильность ниже. Возможно на сервере стоит юзать дебу. На домашней же станции по-моему поровну.

IRainman 11 октября 2018 в 13:58

+1

У меня несколько иной опыт, да ещё и опыт друга, который работает админом в Яндексе говорит о том, что Ubuntu самая адекватная ибо в Debian слишком много всего не работает или работает дико криво из коробки, в т. ч. железо выпущенное за последние несколько лет, а это вообще ужас от которого я бегу и который даже не пытаюсь разгребать.



Massacre 21 сентября 2018 в 14:07

0

Скорее всего, применяла каждое по отдельности, создавая точки отката и наполняя базу winsxs. Вообще, раньше в таких случаях сервис-паки выпускали…

IRainman 23 сентября 2018 в 17:01

0

Очистку диска от админа попробуйте :) Долго она могла разгребать кладбище ранее использовавшихся версий и пакетов. Ещё возможно что какая то проблема с базой, если файл "%SYSTEMROOT%\Logs\CBS\CBS.log" или логи в архивах рядом ОЧЕНЬ большие, то явно второе.

P.S. У меня 7 установлена на часть машин ещё в июле 2009 (как только на msdn образы появились) и до сих пор работает, SP1 на неё тоже встал просто апдетом. С апдейтами совсем диких проблем нет.

svcoder 21 сентября 2018 в 11:03

-2

Не вижу никаких проблем, т.к. многих ограничений старых компьютеров (вычислительная мощность, объем оперативной памяти, скорость и объем дисковой памяти) на данный момент не существует, что открывают дорогу более универсальным и понятным методологиям разработки. На мой взляд автор застрял в 90-х



serbod 21 сентября 2018 в 11:14

+3

Просто очень обидно, что в играх за секунду отрисовываются десятки полноэкранных картинок целого мира с сотнями тысяч элементов, а в текстовом редакторе за это время только предпросмотр печати одной черно-белой страницы отрисовался.



WizardrylB 21 сентября 2018 в 11:21

+1

Автор прав в одном и это — **отсутствие жестких стандартов**! Разработка ПО, да и ИТ в целом проходит лишь первый этап эволюции, когда технологии изготовления продукта или изучены слабо или совсем не изучены… И на данном этапе, когда спрос выше предложения, балом в ИТ правит маркетинг. Но не за горами и следующий этап, ведь эволюция циклична по своей природе…

svcoder 21 сентября 2018 в 11:21

0

сложность игрового движка значительно ниже сложности Word 2016, если считать количество возможных сценариев работы. с игровым движком не требуется интеграция из других систем, он не может содержать компоненты других систем, у него нет тысяч вариантов отображения текстов и макетов страницы. там сплошная математика, часть из которой делается силами GPU.

KaneUA 21 сентября 2018 в 11:34

0

Посмотрите на досуге на Unity.

ledocool 21 сентября 2018 в 12:15

0

А вас не смущает, что простой по своей концепции макрософт вход оказывается сложнее игрового движка?
А вас не смущает, что его аналог (open office write, кажется, называется) вроде умеет все то же, но чет не тормозит?
А вас не смущает, что свистоперделок в этом новом «вроде» больше чем функций?

VolCh 23 сентября 2018 в 15:17

0

Вот далеко не всё то же. По мелочам вроде, типа отсутствия возможности печати сначала нечетных а потом чётных страниц, или достаточно адекватная проверка грамотности, но из таких мелочей и накапливаются предпочтения.

ledocool 23 сентября 2018 в 16:17

0

> По мелочам вроде, типа отсутствия возможности печати сначала нечетных а потом чётных страниц

Можно. Вероятно вы привыкли делать это как в Word, а тут это может быть по-другому.

> достаточно адекватная проверка грамотности

Может я какой-то кривой, но чет что то фигня, что это фигня. Если слова по словарю проверяет более-менее, то вот речевые ошибки тот же ворд настойчиво предлагает сделать. Так правильнее по его мнению.

А вообще из всех методов набора текста мне больше всех понравился tex. Но если надо быстро и плевать на того, кто читает твои опусы, то по мне так либра будет получше — уж извините, но курсор в новом офисе плавает по экрану как палка в дерьме. Из-за этого теряется «тактильная отдача» при наборе текста и из-за этого при быстром наборе очень тяжело.

nmrulin 21 сентября 2018 в 11:26

0

Нельзя требовать производительность. Иначе бы вместо 100500 новый языков народ бы программировал только на Дельфи/C++ с ассемблерными вставками в узких местах. Тогда можно было бы обучить больше кодеров, чтобы они более-менее производительно кодили. И постоянно бы переписывали старый код, двигаясь к большей производительности. Но штаты всё равно бы были вдвое больше против теперешний. Вывод — программистов было бы больше. Но получали бы они вдвое меньше, чем сейчас. Кроме того труднее было бы делать отбор. Сейчас делать отбор просто — подними хайп с каким-нибудь Haskell, и посмотри, кто его выучит. Кто выучит — они обладают лучшими способностями к обучению. Потом, когда на этом языке будут криво/косо кодить все, надо поменять язык для новой программы. И можно опять проводить отбор. Правда производительно кодить никто не будет ни на одном языке. Ну да ладно. Также людям бы пришлось учиться писать более менее читаемый и интуитивно-понятный код. Иначе бы по 20 лет не никто бы не захотел работать с одним и тем же C++ кодом.

DrPass 21 сентября 2018 в 12:04

0

Ой, ну вот только не надо так. Как раз современные инструменты и рассчитаны на то, чтобы можно было любого посадить писать код, потому что надо больше кода, потому что все хотят новый софт. И как раз одна из причин ухудшения производительности и качества софта в том, что порог входа в профессию значительно упал.

nmrulin 21 сентября 2018 в 13:58

0

Я что-то не уверен, что можно на Haskell посадить любого.
И что есть хоть один современный язык проще Турбо Паскаля 30-ти летней давности.
Разве что не менее древний Оберон.

Модулей, да в современный языках больше. Нажал кнопочку и типа работает. Но при этом 99% функций из библиотек не используется. А написать что-то своё под узкую задачу никто не умеет.

Вот порог «написать что-то своё» не упал а наоборот, катастрофически возрос.

DrPass 21 сентября 2018 в 15:04

+2

> Я что-то не уверен, что можно на Haskell посадить любого.

:) Ну так и в прошлом все тоже не Лиспе и Прологе начинали программировать. 99% армии программистов Хаскель никогда в глаза не видела, и не увидит, и даже не заинтересуется.

> И что есть хоть один современный язык проще Турбо Паскаля 30-ти летней давности.

Порог входа у C#, Java, JS невелик. Ну т.е. для хеллоуворлда вам не нужно знать много абстракций. Зато чтобы писать профессиональные приложения, Турбо Паскаль от вас потребует и ручное управление памятью, и работу с компоновщиком, и директивы компилятора и т.д. Хотите сделать интерфейс приложения — точно так же, создавайте ручками контролы (это если вы дожили до того времени, как у вас в ТР появилась библиотека Turbo Vision, если нет — то сперва ещё напишите их), следите за их временем жизни и т.д.

> Модулей, да в современный языках больше. Нажал кнопочку и типа работает. Но при этом 99% функций из библиотек не используется. А написать что-то своё под узкую задачу никто не умеет.

Вот то-то и оно. В авгиевых конюшнях npm, например, есть тысячи модулей на несколько строк, которые быстрее было бы самому написать, чем нагуглить. Но они тянутся зависимостями в тысячи проектов, множатся и плодятся.

BD9 22 сентября 2018 в 06:04

0

Оберон испытал влияние Паскаля, т.ч. Оберон менее древний, чем Паскаль.

0xd34df00d 22 сентября 2018 в 16:35

0

То самое чувство, когда ты рад, что на хаскеле *никто не пишет*, потому что ни одна из проблем, описанных в посте, гарантированно с ним не связана.

Acuna 23 сентября 2018 в 14:41

0

А можно вкратце почему? Очень оптимально написанные библиотеки, и/или все что нужно из коробки?

Druu 23 сентября 2018 в 14:56

0

На хаскеле не пишут слаки или браузеры или хабры, по-этому они на хаскеле и не тормозят.
Ведь не может тормозить то, чего нет.
Но на самом деле все, что обсуждается в статье, вполне для хаскеля применимо, просто в иной полярности. Классическая либа для решения чего-нибудь на хаскеле это переусложненное ради лулзов автора говно. Потому что на хаскеле не пишут для того, чтобы решать задачи (большей частью, кто-то где-то, конечно, пишет, но это капля в море все), а пишут по приколу, хачкель жи.

0xd34df00d 24 сентября 2018 в 07:25

0

Зато всякие компиляторы (идрис не тормозит!111) или оконные менеджеры пишут.

К счастью, то, что в хаскеле пишут ради лулзов, дешугарится компилятором в подавляющем большинстве случаев (если вы не экзистенциальными типами упарываетесь, конечно, там эрейзинг же) и компилируется во вполне эффективный код.

Впрочем, да, что-то вы мне напомнили, пойду попинаю автора hakyll, чтобы на мой пулл-реквест уже посмотрел, наконец, а то там ппц и concatMap списков.

Druu 24 сентября 2018 в 09:30

0

> или оконные менеджеры пишут.

Которыми пользуется 3.5 анонимуса, то есть чистой воды менеджер ради лулзов.

> и компилируется во вполне эффективный код.

Так я ж и говорю, у хаскеля нет проблем в эффективности, там проблемы как раз с другого края.

Если на условном пыхе пишут андерквалифаед, то на хаскеле — оверквалифед, скажем так.

0xd34df00d 24 сентября 2018 в 16:56

0

Ну я вот недавно проникся ~~профункторами~~ линзочками, например, даже вроде понимаю, зачем оно так сделано, и чем больше своего типобезопасного кода пишу, тем больше понимаю все эти прыжки и ужимки с семействами типов, datakinds, вот этим всем.

И тем больше понимаю, что это всё просто костыли поверх отсутствия завтипов (и плевать на консистентность как логики, чай, не теоремы доказывать нужно, корректности при завершимости достаточно).

Druu 26 сентября 2018 в 13:51

0

> Ну я вот недавно проникся профункторами линзочками, например, даже вроде понимаю, зачем оно так сделано, и чем больше своего типобезопасного кода пишу, тем больше понимаю все эти прыжки и ужимки с семействами типов, datakinds, вот этим всем.

Практика все побеждает, это очевидно. Если каждый день старательно ходить на руках, то привыкнешь, рано или поздно, только так бегать будешь.

Проблема в том что обычно решение оверинжинеред по отношению к достигаемому профиту. Ну то есть накручивается какая-нибудь лютая акробатика за тем, чтобы фиксировать типами какой-нибудь инвариант, который по факту и фиксировать-то не особо интересно.

В мейнстримных языках другой край проблемы — там для людей система типов это нечто вроде того, что та функция получает числа, а эта — строки, ну и можно еще вместо типа сунуть параметр, чтобы не писать много мономорфных ф-й. Что чето там как-то типа фиксировать можно — это обычно невоспринимаемая идея, даже когда показываешь человеку на конкретном примере, что вот — пишем для пары ф-й более конкретную обвязку с типами и нам чекер выдает пол сотни деффектов из которых два — висящие в багтрекере баги. И все равно не понимают, просто другой способ мышления.

Acuna 25 сентября 2018 в 09:32

0

Ахах, ясно, спасибо, это хорошо, потому-что та же Java, например, весьма многословна, одна только запись в файл — это 4 строки минимум, и ладно бы создать пару методов для чтения и записи и пользовать их в своем проекте, а в дальнейшем создать свою либу и подключать ее к последующим проектам, но те разрабы, которые типа не хотят тащить монструозные фреймворки, так и копипастят этот код, поэтому код якобы без фреймворков получается даже больше, чем с фреймворками. В общем, плохо все)

0xd34df00d 24 сентября 2018 в 07:21

0

Отсылка к шутеечке про то, что на хаскеле пишут только факториалы.

VolCh 23 сентября 2018 в 15:23

0

Ну вот в PHP и JS порог не вырос по сути. И это даже расстраивает, что каждый может легко написать, а каждый второй пишет что-то своё, вместо того чтобы использовать одну из готовых популярных поддерживаемых реализаций.

Svbakulin 21 сентября 2018 в 11:59

+1

Немного напоминает сюжет Foundation Азимова. Остаются быстро приобретаемые эмпирические знания, и мало кто владеет знаниями концептуальными и почти никто не пытается понять почему все делается так а не здак «потому что всегда так было».

Проблема что машина может и жрет 100л\км, мно эти сто литров почти ничего не стоят и особо ничего плохого не делают на первый взгляд.

Aingis 21 сентября 2018 в 14:31

+2

Есть достаточно людей с фундаментальными знаниями. Более того, они вполне вырастают из обычных программистов. Да, их не так много, это скорее правая часть «колокола» Гаусса, и стоят они поболее, и конкуренция есть, но вопрос всё же в спросе на таких людей.

Перед обычным рядовым бизнесом не стоит задача «выпустить вылизанное приложение чтобы у всех летало». Обычно стоит другая задача: «выпустить минимально работающее приложение быстрее конкурентов». А затем: «как можно быстрее удовлетворять запросы потребителей на функциональность». Это довольно противоречащие требования, так как «быстро» и «хорошо» при ограниченных ресурсах находятся на разных концах шкалы.

А если не успеть, то от бизнеса ничего не останется. Так как приложение хоть и является важным, но важнее пользовательская база и социальные связи в том числе для удовлетворения финансовых

нужд (инвестирования). Конкуренция сейчас большая, кто не успел на поезд, тот прогорел. Сколько вон писали клонов Фейсбука? Выжили лишь аномалии, и то — в достаточной степени изолированные (в Китае вообще отгороженные от мира).

Svbakulin 22 сентября 2018 в 05:10

0

Все так. Потому и остаются в основном эмпирические знания. Я не разработчик, но в своей области и вижу очень мало людей которые идут хотябы на 50%. Обычно все поверхностно, без достаточного понимания процесса либо вообще без понимания. Дали вход, получили более или менее ожидаемый выход — этого достаточно. И не важно как все работает, правильно ли это, оптимально ли. Черный ящик — работает и бог с ним.
Я к тому что я во много разделяю разочарование автора. В софте похуже — кадый второй думает что может сделать свой собственный форк с BJ&H, поулчается масса негодного недопродукта на который посмотреть без жалости нельзя.
Это если с пессимизмом смотнеть. Можно смотреть по другому :)

aleksandros 27 сентября 2018 в 15:27

0

> . И не важно как все работает, правильно ли это, оптимально ли.

Это следствие всё больше набирающего популярность и всё более «молодеющего» вопроса «А зачем мне это знать?». Люди с детства перестают тянуться к знаниям, а сами процессы познания, размышления представляются «тратой сил», грузом, досадной помехой, отвлекающей от киношки или кальяна.

SirEdvin 27 сентября 2018 в 17:03

0

А в противовес этому есть «стареющий» вопрос «А зачем мне эти ваши новые технологии, мы как работали с windows xp так и будем работать, зато я знаю, как оно все работает».

Я просто смирился с тем фактом, что вы никогда не будете все знать, даже если будете 20 лет сидеть на одной ОС с зафиксированной версией (что в целом невозможно, так как ОС обычно еще и обновляется).

khim 27 сентября 2018 в 18:59

0

Проблема в том, что оба подхода ведут в тупик: если мы пользуемся чем-то, что мы не понимаем, то рано или поздно получим «сюрпризы», которые окажутся как минимум неприятными, а как максимум — фатальными (то, что происходит с IoT-устройствами прекрасный пример). А использовать только старые технологие невозможно потому что они, к сожалению, перестают быть доступными…

fukkit 21 сентября 2018 в 12:14

0

> 2014 — нужно внедрить микросервисы для решения проблем с монолитами.
> 2016 — нужно внедрить Docker, чтобы решить проблемы с микросервисами.
> 2018 — нужно внедрить Kubernetes, чтобы решить проблемы с Docker.

Нужно нанять в два раза больше в два раза более дорогих специалистов (которые теперь «DevOps») и платить в два раза больше «as you go» облачным провайдерам, чтобы содержать весь этот зоопарк контейнеризованных микросервисов и автобилдов на каждый чих.

Такие практики, как *«Пол года работаем над новой минорной версией, тестируем, вылизываем все известные баги, выкатываем, собираем фидбэк, выпускаем фиксы, поддерживаем пользователей **конкретной версии** и, возможно, нескольких предыдущих, не говоря уже о мажорных»* уходят безвозвратно.

Теперь можешь курить прямо здесь! (c)

Сэкономим на тестировании и поддержке! Мы же суровые программисты, зачем на мелочи терять ~~время~~ деньги!?
Вася наколбасил, Петя прочитал-поправил, Саня захреначил в мастер — и понеслось оно по трубам «continuous integration», через тестики да, если повезет, препроды, непосредственно к пользакам на устройства. Все закрыли таски, почувствовали удовлетворение от хорошо сделанной работы и погнали в барбершопы.

Dagron 21 сентября 2018 в 12:23

0

> Приложение Google app, в основном, просто пакет для Google Web Search, занимает 350 МБ! Сервисы Google Play, которыми я не пользуюсь (я не покупаю там книги, музыку или видео) — 300 МБ, которые просто сидят здесь и которые нельзя удалить.

Неудаляемые приложения действительно бесят!
Однако у нас это похоже может измениться
Производителей гаджетов заставят отказаться от неудаляемых приложений

webaib1 21 сентября 2018 в 12:25

0

Какая наивность, считать, что девы проектируют и пишут софт. Мы сидим на митингах, ходим на тимбилдинги, массажы, очереди к смузи/кофе аппаратам занимают оставшееся время. По пятницам у нас пицца/дартс/футбол и пиво с вином, т.ч. уже с утра всякая работа исключена.
Нам никогда ~~этой черной страдать~~ думать и проектировать.
И главное — нашим ХРам тоже никогда думать и нанимать толковых — они тусят всегда с нами, очередь за завтраками и обедами сама себя не займет.
Это я еще забыл пару недель черной пятницы и 2 месяца рождества с ноу деплой стратегией, что бы ничего не сломали))) девы криворукие.

poznawatel 21 сентября 2018 в 12:31

0

Нехотя вынужден согласиться с автором статьи — действительно, ситуация — дрянь, функционирование неэффективных программ жрет количество ресурсов, которое растёт быстрее закона Мура.
Предлагаю натравить на тупой софт «зелёных», как единственную реальную силу, которая относится к транжирству ресурсов отрицательно (остальные части элит такое поведение поощряют).

ZAMnoTEX 21 сентября 2018 в 12:39

+2

Немножко не в тему: в детстве мне врезались в память картинки из школьного учебника информатики. Картинки с большим смыслом. Напомню, это учебник от 1991 года!







domix32 21 сентября 2018 в 12:50

0

А что за намеренное убийство случайных процессов в Linux?

mayorovp 21 сентября 2018 в 13:49

0

Гуглите OOM killer

PsyHaSTe 21 сентября 2018 в 14:49

+1

OOMKiller, полагаю.

YourChief 21 сентября 2018 в 21:52

+2

Сомнительный аргумент, приведённый автором. В линуксе процессы убиваются не без причины и не случайно.

ganqqwerty 21 сентября 2018 в 13:00

+1

Радует, что инженеры начали вслух говорить об этом регрессе. Теперь осталось понять, можно ли извлечь денег от того, чтобы данный регресс превратить в прогресс.

VolCh 23 сентября 2018 в 15:28

0

Это только мнение, что данные процессы регресс. С других точек зрения это можно считать лишь платой за прогресс.

Druu 23 сентября 2018 в 15:48

0

Непонятно вообще, почему это должно быть плохо.
Мощности компьютеров возросли, стало больше памяти. Какая есть причина не использовать эти ресурсы?

PashaNedved 24 сентября 2018 в 12:06

0

500к рублей на железо за 10 лет за 1 не игровой ПК. Почему я должен платить за железо и за софт?

Druu 26 сентября 2018 в 14:01

0

> 500к рублей на железо за 10 лет за 1 не игровой ПК.

150к за игровой компуктер под третьего ведьмака на ультре, 60+ фпс, все дела, из них 50к — за мышку с клавой (механика) и звуковую карту. Куплен чуть больше двух лет назад и в ближайшие два года никакого смысла менять нет.

Вы там чего, свой неигровой пк каждый месяц новый покупаете?

PashaNedved 27 сентября 2018 в 18:17

0

Напечатать этот коммент не могу нормально — тормозит!!! Расскажите мне через год-два, о том что нет смысла менять пк/железо. Если, конечно у вас ssd или другое железо раньше не накроется.



PsyHaSTe 27 сентября 2018 в 22:03

0

Вы можете купить любое железо на рынке, включая проц за 3000$, этот тред будет все так же тормозить… Так что смысла менять действительно нет. С начала нулевых ничего не меняется — комп стоит 15000$ и должен меняться раз в 4-6 лет чтобы быть актуальным.

staticlab 28 сентября 2018 в 01:45

0

У меня этот тред в ФФ работает нормально на десктопе и ноутбуке вовсе не за 15 тыс. долларов. Может быть потому что у меня стоит жёсткая блокировка отслеживающих скриптов типа ЯМетрики, ГАналитики и всяких соцсетей?



rsi 28 сентября 2018 в 06:00

0

У меня на сафари единственное отличие этого треда от других это время загрузки с нуля. Кстати с 5ю отрытыми вкладками, потребление памяти < 300 МБ



PsyHaSTe 28 сентября 2018 в 12:31

0

У меня тоже стоят всякие блокировки. Но нет, я не думаю, что проблема в скриптах, т.к. на других страницах они так же (не)работают и все ок. Проблема в самом браузере, полагаю. Менять же браузер для просмотра одной страницы на хабре считаю не очень удобным.

Для теста отключил все сторонние сайты в адблоке — разницы нет.

staticlab 28 сентября 2018 в 13:13

0

> Проблема в самом браузере, полагаю. Менять же браузер для просмотра одной страницы на хабре считаю не очень удобным.

Может быть просто радикально уйти с тормозного Хрома совсем? :)



PsyHaSTe 28 сентября 2018 в 14:37

0

В предыдущем комментарии был небольшой намек на то, что на остальных страницах браузер показывает себя лучше, чем альтернативы.

Я раньше каждые год-два менял браузер на конкурирующий, когда глюки одного из них доставали. Закончилось это года 2-3 назад, когда я перешел на хром в очередной раз, а с ним не оказалось проблем.

staticlab 28 сентября 2018 в 16:11

0

Тогда почему бы не оформить баг в Хром, чтобы они пофиксили?



Am0ralist 28 сентября 2018 в 16:13

0

Ну вот может разработчики Хабра разберутся, запилят.

PS. 2048 комментарий был у вас, юбилейный…



0xd34df00d 29 сентября 2018 в 02:34

0

Ну хз, я почти с начала нулевых менял раз в пару лет и ощущал серьёзный рост, и перестал это делать, когда вышел SB.



rsi 28 сентября 2018 в 06:02

0

Печатаю тут без проблем. Никаких лагов не замечено.



0xd34df00d 29 сентября 2018 в 02:35

0

Тыща фунтов стерлингов за игровой компуктер под баттлу, 4 и 1 на максималочках в 2560x1440, осень 2014. Чего там куда ещё?

VolCh 24 сентября 2018 в 18:03

0

Не у всех компьютеров мощности возросли. В среднем я бы сказал, что мощности компьютеров растут реже, чем компьютеры просто утилизируются.

Druu 26 сентября 2018 в 14:03

0

> Не у всех компьютеров мощности возросли.

Смотря за какой период.
Ну и так-то давайте будем честными — люди, которые не имеют денег обновить пека или не готовы зотя бы взять кредит на новенький айфоун, обычно не входят в какую-либо ца в силу своей слабой платежеспособности.



julia4545 21 сентября 2018 в 13:09

0

Переходим на Линукс, статичные html странички и консоль.

ua30 21 сентября 2018 в 13:09

+8

Хуяк-хуяк и в продакшн. Есть такая тема. Самое интересное, нормально кодить не намного дороже получается.

Я уже лет 15 отошел от прикладного программирования. Программирую движки сайтов. Знаете, иногда в футере пишется время генерации странички, количество запросов к БД, объем потребляемой памяти… Бывает такую информацию можно найти в HTML-комментариях. Больно видеть как какой-нибудь несчастный форум, или даже блог генерируются десятые доли секунды, а порой и выше. И дело тут не в сервере. И по полсотни запросов к БД, а то и больше. И вы еще не видели сами эти запросы. А память жрет по два-три и даже пять десятков мегабайт для своего выполнения…

Последнее время занимаюсь разработкой движка для достаточно крупного интернет-магазина. Самые сложные страницы генерируются за 0,02 сек (генерация, не кеш). В среднем используется 12-16 запросов к БД. Учитывая время генерации всей странички, можно понять что запросы все хорошие. На 64 битной системе в среднем движок хавает 2Мб памяти на генерацию страницы… По производительности разница в 10 — 100 раз. А я всего лишь не использую сторонние библиотеки и придерживаюсь функционального подхода к программированию. Я как то думал даже на Си писать движки сайтов. Но потом остановился: ну я же не полный задрот в конце то концов!

Что в итоге? Да, разработка не такая быстрая. Но я один справляюсь. Работает все на простом сервере. Я

думаю на том же сервере при такой посещаемости (до 10к хостов всутки) работал бы и обычный движок. Просто медленней. Но по-другому я просто не могу. 10к в сутки это примерно 36,5 миллионов человек в ближайшие 10 лет. А если бы я разрабатывал CMS не для одного сайта, а универсальную, которую установят на десятки и тысячи сайтов? Почему все эти люди должны страдать из-за одного или нескольких разработчиков?

И дело не заканчивается CMS. HTML код чистится и сжимается PHP функцией в десять строк. Хоть на сервере и включен GZIP, но почему нет? Может хоть браузеру будет чуть легче. JavaScript только там, где он реально нужен. Сама верстка — только руками в ~~текстовом редакторе~~ (вру, в PhpStorm). CSS, JS минимизирован. Все загружаемые изображения автоматически оптимизируются и сжимаются. Картинки для дизайна сайта идут в двух спрайтах, а не в двух сотнях файлов. Как то так…

Так что с этой ситуацией как обычно. Проблема есть. Что с ней делать? Не знаю. Наверное, единственное, что тут можно сделать — начать с себя.

michael_vostrikov 21 сентября 2018 в 21:07

+2

> Но я один справляюсь.

Во-во. Потому и справляетесь.

janvarev 22 сентября 2018 в 20:39

0

О, очень знакомо. Тоже начинал разработку с движка для сайта в далеком 2004 — тогда многое приходилось делать ручками и оптимизировать.

В результате движок живет и здравствует во вполне актуальных проектах. А оптимизация осталась… просто не подключалось ничего лишнего. До сих пор не используется PHP autoload (все библиотеки — ручками, благо их немного), доступ к БД давно вылизан в собственной структурной библиотеке в 70 Кб (вместо какого-нибудь Doctrine на 2Мб). Для нужд хватает.

Результат — стабильная генерация страниц в пределах 0.1 с, обычно — в пределах 0.05 (не кеш). Исключение — навороченная статистическая админка (до 3-4 с)… ну лениво её вылизывать, когда ей пользуются два человека и не так часто.

sumanai 22 сентября 2018 в 21:01

0

Только я не вижу ссылок на гитхаб, где можно легко скачать эти оптимизированные библиотеки. А Doctrine, Symphony и прочее — пожалуйста, качай не хочу.

janvarev 22 сентября 2018 в 22:04

0

Библиотеки вылизаны для локального использования. Неоднократно задумывался над тем, чтобы их выложить… но понял, что это не имеет для меня особого смысла.

Положительных отзывов я вряд ли получу, а комментариев в духе «велосипедостроитель» и «а где возможность сделать left join с алиасами» будет полно. Всем рулит обучение и маркетинг, на который у меня нет времени.

Конечно, понятно, что если только начинать, то фреймворки лучше всего — там и практики использования, и компоненты, и лучшая документация; просто рассказываю, так сказать, историю о том, чего можно достичь, если потихоньку пилить свой двиг в течении 14 лет))

PS: Если это был не риторический вопрос, а реальное желание ознакомиться, то ссылки есть на моем сайте, который в профиле.

sumanai 22 сентября 2018 в 22:28

0

> то ссылки есть на моем сайте, который в профиле.

К сожалению, ссылок в профиле не вижу.

janvarev 22 сентября 2018 в 22:54

0

Тут ссылок разве нет? habr.com/users/janvarev
На всякий случай прямую даю: janvarev.ru

sumanai 23 сентября 2018 в 01:40

0

У меня нету ссылок в вашем профиле. Сейчас поискал в исходном коде страницы, и тоже не нашёл.
Спасибо за ссылку.

janvarev 23 сентября 2018 в 08:00

0

Понял, спасибо. Интересно, что в вашем профиле тоже ссылок нет — сейчас зашел в режиме инкогнито — ссылок нет нигде. UPD: Понял, что нужно разрешить показ информации другим пользователям в настройках приватности. Ну, разработчики Хабра…

PS: стало интересно, чем вы занимаетесь, погуглил по заголовку в названии про phpBBex. Мое почтение; я-то phpBB, как старожил, помню и понимаю, что такое поддерживать его развитие вплоть до современного интернета.

sumanai 23 сентября 2018 в 12:56

0

> Мое почтение

С нуля я мало что написал, да и они сами сейчас переезжают на Symphony и прочее. Но за почтение спасибо.

AlexHanter 1 октября 2018 в 11:08

0

Ну у меня масштаб не вашего уровня, но тоже перестал пользоваться CMS и библиотеками, и десять лет назад перешёл к написанию сайтов на чистом html + css и немного php. Никаких jQuery и фреймворков. Для большинства сайтов этого за глаза хватает. Сайт из пяти страниц порядка 2-3 мб занимает. Даже базы данных не использую, всё прям на странице свёрстано :) Усиленно пропагандирую такой подход среди клиентов.

gearbox 21 сентября 2018 в 13:13

+4

Нифига тут наобсуждали. Осмелюсь подвести черту (хотя нет конечно).
Управление сложностью. Человечество столкнулось с этой проблемой в таких масштабах впервые. Это не только программирование, это и космос и фармацевтика и ядерная физика думаю много еще других областей. Собственно поэтому и проблемы — нужны новые инструменты управления сложностью. Дешевые и простые. Конкретно в программировании я думаю это будет формальная верификация и сопутствующая эволюция языков. Чего паниковать то? Утрясется.

marenkov 21 сентября 2018 в 13:40

+1

А как не паниковать, если идет к тому, что для запуска блокнота скоро понадобится 64 терабайта оперативки и 128-ядерный процессор с частотой 2 терагерца? Благо, что хоть потребляемая мощность ограничивается возможностями отведения тепла и вряд ли в ближайшее время выйдет за 1 киловатт (хотя, если посмотреть на историю развитие систем охлаждения — от шести алюминиевых ребер до тепловых трубок с водяным охлаждением, уверенности поубавится).

gearbox 21 сентября 2018 в 14:00

+1

>если идет к тому, что для запуска блокнота скоро понадобится 64 терабайта оперативки и 128-ядерный процессор с частотой 2 терагерца?

Конечно идет, и это хорошо. Решение проблемы начинается с ее признания. Проблему начали признавать, об этом говорят все чаще. Программирование поворачивается в сторону математики, подъем функциональщины, metamath, coq, haskell/idris, ocaml, да даже frama, вот это все — это котел, в котором сейчас варится возможное решение проблемы. Накапливается опыт, оттачиваются инструменты. Есть чекнутые компиляторы, есть как минимум одна чекнутая ось (точнее ядро). Как только это станет доступно (в плане дешевого использования) бизнес возьмет это на вооружение. Возможно сам заказжет развитие если увидит выгоду. Это не решится за один день но это решается, в том числе и вот прямо сейчас.

BD9 22 сентября 2018 в 06:17

0

Контрактное программирование развивается с 1986 года. Насколько массово оно используется?

gearbox 22 сентября 2018 в 13:12

0

Не особо. А к чему вопрос?

0xd34df00d 22 сентября 2018 в 16:39

0

Контрактное программирование имеет к proof-carrying code примерно такое же отношение, как темплейты C++ к GADT.

gonzazoid 23 сентября 2018 в 00:59

0

контрактное программирование имхо подняло вопрос который потом отработали в first class types / dependent types / refinement types что вполне себе используется в упомянутых @gearbox инструментах.

0xd34df00d 24 сентября 2018 в 07:27

0

Контрактное программирование примерно ровесник завтипов (если не чуть младше, надо папиры смотреть), так что вряд ли оно было родоначальником вопроса. В конце концов, идея объединить язык термов и типов достаточно очевидна.

0xd34df00d 22 сентября 2018 в 16:43

+1

При этом проблем много.

Я пару месяцев назад ради интереса доказывал на идрисе, что функция слияния в merge sort выдаёт сортированный список — задолбался, это какой-то ассемблер для математиков. Доказать, что merge выдаёт перестановку объединения входных списков, я уже не смог, ибо наткнулся на баг в тоталити чекере.

На днях я доказывал, что small-step evaluation semantics для простейшего языка с if и булевыми константами тотально. Это тоже, гм, ну, весело.

Как доказывать на идрисе, что small-step evaluation semantics для STLC тотально, я понятия не имею.

Так что, ИМХО, реалистично — от сегодняшних инструментов до решения проблемы как от ассемблера и в лучшем случае алгола-60 до сегодняшних инструментов.

PsyHaSTe 21 сентября 2018 в 15:52

0

Ну и хорошо, у нас в итоге либо будут оптимизированные приложения, либо 64 терабайта оперативки в каждый комп. Не вижу, чем что-то из этого плохо.

marenkov 21 сентября 2018 в 16:07

+1

Плохо то, что нам, потребителям, приходиться тратиться на все новые устройства, которые смогут справляться с приложениями, которые делают тоже самое, что делали старые приложения на старых устройствах. Т.е. нам приходиться тратить свои жизненные ресурсы на то, чтобы оставаться на месте. По сути это запрограммированное старение. Лишь с некоторых сферах, таких как игры и навигация, новые мощности используется более-менее эффективно.

P.S. А могли бы, например, иметь в смартфоне полную модель погоды на земле с возможностью прогнозирования дождя в определенной точке с точностью до минуты.

san-smith 21 сентября 2018 в 19:43

0

> А могли бы, например, иметь в смартфоне полную модель погоды на земле с возможностью прогнозирования дождя в определенной точке с точностью до минуты.

Не могли бы. И дело не в вычислительных мощностях.

Дело в том, что математическая модель всегда является лишь упрощенным отражением действительности. Поэтому мы никогда не сможем получить бесконечно точный результат — лишь некоторое приближение. Хуже того, в силу неточности нашей модели, прогнозируемый результат будет отличаться от реального тем больше, чем больший отрезок прогнозирования мы выбрали (и не важно, с какой точностью мы считали).

В качестве примера: пусть некоторый процесс в действительности описывается функцией f1(x) = (2^0.5)^x, но так как мы не умеем считать корни, было выбрано приближение f2(x) = 1.4^x.

▸ Графики получатся следующие

Можно увидеть, что с дальнейшим увеличением интервала прогнозирования, результат прогнозирования по нашей модели будет сколь угодно отличаться от реального.

marenkov 21 сентября 2018 в 23:23

0

Во-первых, я утрирую.
Во-вторых, я не писал об абсолютной точности, а о допустимой ошибке в одну минуту.

khim 22 сентября 2018 в 03:57

0

> Во-вторых, я не писал об абсолютной точности, а о допустимой ошибке в одну минуту.

Ошибку в одну минуту вы можете получить только если уничтожите все облака и будете поливать планету из лейки.

Ибо достаточно какой-нибудь местной котельной сменить сорт угля — и всё, тучка прольётся не там и не тогда, когда предсказано.

Как вы собираетесь это контролировать?

marenkov 22 сентября 2018 в 13:59

0

Смотрите пункт первый.

khim 23 сентября 2018 в 18:39

0

Ну если первый… Ибо с дождём сложнее всего даже не предсказать когда он случится, а то, куда он при этом прольётся. Со всеми вытекающими ощущениями для субъективных оценок качества прогноза…

san-smith 22 сентября 2018 в 08:55

0

Даже с точностью в одну минуту вы не сможете получить прогноз на период больше, чем полчаса (число взято абстрактно). Выше отметили почему — любые изменения условий, неучтенные в нашей модели, достаточно быстро сделают прогноз не актуальным. С математической точки зрения это можно проиллюстрировать графиками из предыдущего комментария.

А сделать модель очень точной и невозможно, и бессмысленно. Невозможно, потому что мы не знаем и не можем описать все факторы влияющие на этот процесс. Бессмысленно, потому что расчёт по такой модели был бы слишком сложным и к моменту окончания расчётов уже бесконечно устарел бы.

w1ld 21 сентября 2018 в 13:39

−1

Да, такое имеет место быть. В Андроиде, под Виндой, в Вебе приложения раздуты, тормозят и пр. Видимо, нужна другая модель монетизации, роста для софта. Какая-нибудь зависимость поощрения разработчика от его вклада в программу. Если его код используется много раз, то и поощрение больше. Привязать как-то блокчейн, всем сделать аккаунты и списывать с них за использование кода. Добавить возможность поощрения, встроенную. Плату сделать очень маленькой. Аналогия с реальными физическими местами или воздухом. Вход в метро — плата, в кино — плата и пр. Ну и у всех сделать автоматически пополняемый баланс, чтобы реализовалось право на свободу и пр. Чтобы не было бедных, но и чтобы разработчики поощрялись.

marenkov 21 сентября 2018 в 13:47

+1

А ничего, что, по существующим оценкам, блокчейны уже потребляют около 1% вырабатываемой в мире электрической энергии? Я уже молчу о занимаемых объемах памяти.

w1ld 21 сентября 2018 в 14:01

0

Идея в том, чтобы обладатели ПО (разработчики, тестировщики и др.) поощрялись и мотивировались делать лучше, но и чтобы это происходило эффективнее, чем сейчас, чтобы пользователи платили за это. Иначе получается так, что пользователи пользуют, а разработчики вот так проигрывают. В городах же мы платим за дороги, за все обеспечение, за уборку мусора и пр. Почему разработчики должны делать это бесплатно? Они и не хотят и мы имеет такой рынок.

marenkov 21 сентября 2018 в 15:08

0

Разработчики получает оплату за свою работу. Вот только покупатели хотят новый модный «айфон», а не новое более надежное и быстрое приложение работающее на старом. Да и производители железа обеими руками за то, чтобы новые приложение были тяжелыми и требовали самого нового железа.

Альтернативная экономическая модель — это платные обновления ОС и приложений. Но мне трудно себе представить, чтобы набралось достаточно желающих платить за то, что их приложения будут делать то же самое, но чуть быстрее. А жаль.

DrPass 21 сентября 2018 в 15:15

0

> Вот только покупатели хотят новый модный «айфон», а не новое более надежное и быстрое приложение работающее на старом.

Покупатели сознательно предпочли бы в основном как раз второй вариант. Это производитель айфонов заинтересован в том, чтобы продать несколько десятков миллионов гаджетов в следующем году, а для этого ему нужно убедить покупателя, что ему это нужно. Для этого там целая армия маркетологов есть. А убедить можно лишь добавив какие-то фичи в новое устройство, которых нет, и не будет в старом.

PsyHaSTe 21 сентября 2018 в 15:54

0

Ну да. «Покупатель, ты не знаешь, чего хочешь, это Я знаю. Так что держи X. Что значит ты не хотел? А ну бери, пока дают».



marenkov 21 сентября 2018 в 16:17

0

Очередь за новым iPhone.



Маркетологи с автоматами прячутся за столбами.



0xd34df00d 22 сентября 2018 в 16:46

+1

Это какой-то социальный конструкт уже. Можно же, блин, сделать предзаказ.

picul 21 сентября 2018 в 18:48

0

Вот Вы язвите, а зря, ведь пользователь и правда не знает чего хочет. Откуда людям знать, что мобильный клиент для соц-сети/месседжера может работать быстро, если они сидят в Facebook'е/Viber'е? И требования новых фич/редизайна интерфеса/другие поводы убить производительность Вы ведь не напрямую от пользователей получаете, они исходят от руководства, которое думает как раз так, как Вы написали.

VolCh 23 сентября 2018 в 15:37

0

Выбор нового девайса, который имеет больше возможностей вы считаете не сознательным? А обновление софта ради более функциональной версии тоже?

DrPass 24 сентября 2018 в 11:56

0

> Выбор нового девайса, который имеет больше возможностей вы считаете не сознательным?

Считаю сознательным. Но есть нюанс. Я, например, года полтора назад купил новый телефон. Я их покупают редко, но топовые. И служат они долго. В новом телефоне у меня есть такие возможности, которых нет в старом: турбозарядка, сканер отпечатков пальцев и хорошая камера с автофокусом. Это не считая ядер, гигагерцев и гигабайтов, естественно.
И знаете, зарядка, камера и сканер пальцев — полезные, но не настолько нужные фичи, ради которых я бы менял телефон целиком (и не только я, а и многие миллионы других пользователей, которые к гаджетам относятся не как к любимым игрушкам, а утилитарно). Я поменял как раз потому, что в девайсе 2012-го года уже перестало хватать ядер, гигагерцев и гигабайтов. И вот тут как раз переходим ко второй части вашего вопроса:

> А обновление софта ради более функциональной версии тоже?

Ко мне постоянно прилетают обновления из маркета, софт становится толще, неповоротливее. Но я одного не могу найти — новых функций. Под микроскопом, конечно, нахожу. За полтора года, например, в вайбере появились новые эмодзи, возможность цитирования и удаления сообщений. Ок, спасибо, но он раза в два вырос по размеру. И стал тормозить. Что там за обновления гуглокарт или браузера, одному гуглу известно. Я, как потребитель, могу точно сказать — я понятия не имею, чем они там занимаются, я вижу, что мой девайс только хуже работает с каждым обновлением, но новых функций я практически не получаю.

VolCh 24 сентября 2018 в 18:08

0

Тогда зачем обновляете? Сбросили до заводских настроек и отключили обновления, нет? Нет, есть некоторые программы, которые без последних обновлений отказываются работать, но субъективно их мало.

DrPass 25 сентября 2018 в 11:19

+1

Ну то-то и оно — как раз основные и отказываются. Скайп, Вайбер, карты, да даже банальные гуглофоточки отключают свои старые версии. Функционал гуглофоток пятилетней давности меня целиком устраивал (в просмотрщиках картинок последние лет 15 вообще трудно что-то новое придумать), мне нафиг не сдалась фича «автоматически делать коллажи на основе предпочтений пользователя», которая отличает старую версию от новой. Но старую в какой-то момент просто отключили.

w1ld 21 сентября 2018 в 15:33

+1

> А ничего, что, по существующим оценкам, блокчейны уже потребляют около 1% вырабатываемой в мире электрической энергии?

Да, должно быть большое число. Но если сравнить с количеством энергии, которое потребляет неэффективный софт и с временем которые люди теряют на нем, то я думаю что такое сравнение будет далеко в пользу этой идеи "живого софта".

Эта концепция совершенно другой оплаты за софт. Есть например такая сеть как steemit.com, где за чтение можно заплатить. Что-то такое было бы здорово, на мой взгляд, сделать для софта вообще. Если я открыл браузер, то я бы платил за использование всего софта который бы запускался у меня, но очень мало. Как я плачу за дороги в моем городе, но здесь это была бы встроенная функциональность.

Druu 23 сентября 2018 в 16:26

0

> чем сейчас, чтобы пользователи платили за это

Пользователи и сейчас за это могут платить, но не платят. Пользователям это не нужно. То, что вы хотите — в первую очередь будет неудобно для пользователей. А если это будет неудобно пользователям — то кому это нафиг надо-то?

VolCh 23 сентября 2018 в 16:19

0

Кого вы имеете в виду под разработчиками, всех лиц или только физлиц? Другими словами подразумевает ли ваша схема запрет найма программистов, запрет на получение ими зарплаты или вознаграждения по другому договору? Все программисты и прочие разработчики, желающие обеспечивать себя разработкой, должны будут писать код бесплатно, надеясь получить приемлемую прибыль от инвестиций своего времени в разработку какой-то то программы, принимая на себя все риски «невыстреливания»? Останется ли в вашем мире профессия программист или будут только бизнесмены, частью предпринимательских способностей которых будет способность писать код?

stanislavskijvlad 21 сентября 2018 в 15:15

+1

KolibriOS в массы

ua30 21 сентября 2018 в 15:29

+2

Через сутки после прочтения статьи комп стал тормозить еще больше. Заметно больше. Даже там, где раньше не было проблем.



Aingis 21 сентября 2018 в 16:15

+2

А как стала тормозить эта страница… Даже на простую прокрутку уходят секунды. И появление символов на экране имеет задержку после нажатия на клавиши. Навскидку вижу на странице несколько обработчиков на нажатие клавиш.

mayorovp 21 сентября 2018 в 17:09

0

Тормоза хабра на страницах с кучей комментариев не имеют никакого отношения к числу обработчиков — я это проверял. Просто отключал обработчики в девтулзах, но тормоза оставались.

Как я уже писал выше, тормозит сам браузер. Ему почему-то кажется, что появление нового символа в textarea — это повод сделать полный reflow всей странице.

EvilFox 21 сентября 2018 в 19:02

+2

Хотите секрет? Это всё CSS (СТИЛИ) виноваты. Грохните их dr.habracdn.net/habrcom/styles/1537449801/main.bundle.css и всё станет в сто раз шустрее. Можно через F12 во вкладке Source открыть и потерять содержимое этого файла или перейти через Elements на него в стилях элемента какого-нибудь.
Это не новая проблема. Частенько всякие там кривые нормализаторы (сброс стилей) сильно так тормозов добавляют (но в данном случае проблема глубже). Уже не в первый раз вижу такое на сайтах.

user343 21 сентября 2018 в 17:42

0

Хаброрыба тухнет с головы (врачу — исцелися сам).
А на некоторых форумах гостям или проштрафившимся пользователям включали майнинг криптовалют в браузере. Хабр ещё не настолько опустился.
https://habr.com/post/374131/

InterceptorTSK 22 сентября 2018 в 17:34

0

Если рыба тухнет, то тухнет со всех сторон полностью.

EvilFox 22 сентября 2018 в 17:08

0

Вот тут я дал решение тормозов прокрутки.

zzvkaa 21 сентября 2018 в 17:12

+1

Спасибо, хорошо сформулировано, даже сподвигло на хабре зарегистрироваться). Тема эта меня тоже давно мучает. Как так получается? Понять можно, если смотреть на всю картину мира, в которой программистов — от силы четверть процента всех людей. Т.е., разочарование в том, что даже создание технических ценностей — это социальный процесс, идущий по социально-человеческим законам (м.б. идеальным, но пока не достаточно известным), а не по идеальным физ-мат моделям. Неидеальный код — продукт неидеального социума, а не фейл отдельных специалистов. Как-то так.
Например (шаблон сценария «быстрый IT-стартап»):
1. Старт проекта: денег нет, времени нет. Решение — сделать прототип, запустить, раскрутить, на профит его доделать до идеальности.
2. Сделали на коленках, запустили, раскрутили — ура, профит, KPI=100500
3. Вложить в «ref*cktoring»?) при KPI <= 0.001005 ?) неее… лучше перейти к п.1 с новой идеей)

Al-Capona 21 сентября 2018 в 17:12

+1

Отличная статья, полностью согласен со ней! Придала ещё больше уверенности, что «ассемблеро-эрлангский олдскул» жив и потребность в нём только укрепляется.
В какой-то период времени, вдруг начался ажиотаж вокруг фреймворков и могучих, по объёму, ЯП. В те года я решил, что мой опыт разработки высокоскоростных, маленьких, выжимающих все соки из железа, программ, ушёл в прошлое.
Так как заниматься изучением «новейших» технологий мне не позволяло внутреннее отвращение, пришлось сменить приоритеты на «земной» бизнес.
Теперь же, наблюдая картины происходящего в области ПО и развитого интернета, с радостью вижу для себя возможность вернуться в любимую отрасль и «разорвать» на биты этот могильник несовершенства, став участником строительства новой эпохи.
Автору статьи огромное спасибо! Возвращаюсь! Олдскул жив!



0xd34df00d 22 сентября 2018 в 16:48

0

Я от похожей хрени из машинного обучения ушёл, как только туда в больших массах набижали «практики», умеющие только нейросеточки как-то там в питоне ковырять.

scoffer-by 21 сентября 2018 в 17:12

-3

Проглядев комментарии здешней фауны, гордо именующей себя IT-специалистами, нисколько не удивляюсь написанному в статье.

user_z 21 сентября 2018 в 17:12

0

Добрый вечер! Все, наверно, помнят присказку: «Лучшая программа не та, в которую нечего добавить, а та, из которой нечего убрать». На уроках программирования, в той или иной форме, всем должны были объяснить. Это как на уроках труда учили, что деталь из заготовки надо вырезать таким образом, чтобы получилось минимум отходов, чтобы заготовки ещё на несколько деталей хватило. Это и есть оптимизация, любая деятельность приходит к оптимизации, но только при наличии свободного времени.

Нарыл интересные противопоставления, которые исторически объясняют сложившуюся ситуацию. Так вот, все мы знаем как высоко ценятся скрипки Страдивари и другие подобные штуки, мебель, драгоценности, даже железные оградки. Мало того, что ценятся, так ещё дошли до нас не потеряв своих функциональных качеств, такие же прочные, по-прежнему красивые.

Отличительной особенностью того времени являются города-государства, которыми управляли, во многом, общины ремесленников, феодализм уже отходил, а государства-монархии ещё не сформировались, промышленная революция с её капитализмом была ещё только на горизонте. Прекрасное время, отсутствие принудительного труда, постоянные мятежи против полиции баронов, с большим упорством пытающихся обложить ремесленников всё большим и большим налогом.

Так вот, это было время, когда неторопливый труд позволял прокормиться, была возможность потратить несколько дополнительных часов, дней, даже месяцев на рюшечки-завитушечки, архитектура того времени поражает отточенностью деталей.

Со временем, государства-территории подмяли под себя города-государства, обложив рабочих жителей городов непомерным налогом. Промышленная революция внесла свой вклад, создавалось всё больше фабрик, на которых труд был хоть и оплачиваемый, но всё-таки принудительный, альтернатив не было, свободных ремесленников оставалось всё меньше. Фабричное же производство хоть и отличалось стремлением к экономии затраченных средств, но ручной труд становился дороже машинного, капиталу было выгодней произвести 200 деталей без особой огранки машинным способом, чем 5 деталей ручной, ювелирной работы. Тэп-ляп и в продакшн.

Тогда капитализм впервые показал свой звериный оскал, которым до сих пор улыбается нам. Бизнесу не нужна высокая оптимизация, пользователю не нужна высокая оптимизация, а у программиста на высокую оптимизацию нет свободного времени — свободное время для совершенствования в искусствах (искусстве программирования), — удел анархических утопий.

Таким образом, причина сложившийся ситуации — капитализм. Решить ситуацию можно либо гос.заказом, как и делается для военно-космической отрасли, либо созданием городов-государств, ближайшим аналогом которых можно назвать Силиконовую долину, Сколково, вот только работа должна вестись не из под палки, не из страха потерять работу и умереть с голоду, а на творческом энтузиазме, как это делали свободные ремесленники в свободных городах эпохи возрождения.

scoffer-by 21 сентября 2018 в 17:21

0

Истинно так. Экономика ориентирована либо на результат, либо на прибыль. Одновременно не получается.



PsyHaSTe 21 сентября 2018 в 18:37

+3

Вы вольны сидеть дома и делать завитушечки, сколько вам угодно. Причем если вы делаете более удобный продукт, то у вас будут и покупатели. Так что всё зависит только от вас, никто не заставляет на дядю работать.

sevenguest 21 сентября 2018 в 17:12

+4

Глубочайшая признательность автору! Иногда боюсь, что только меня такие мысли посещают.



polnd 21 сентября 2018 в 18:34

0

Согласен с автором. Одним из мотивов создавать свои программные решения для меня является ровно эта мысль — делать софт оптимальным и простым вопреки современным тенденциям.

Добавлю еще, что не только все стало медленнее. Все усложняется специально еще и для того, чтобы заработать на этом еще больше денег. Но закон сохранения энергии проходят в школьном курсе — если откуда-то взял — в другом месте убудет. Но пипл хавает. Тоже призываю инженеров, от которых все зависит по сути, задуматься над последствиями. Нам в этом мире жить.



Kwisatz 21 сентября 2018 в 23:44

+1

Коллега, наступили на больную мозоль а.
Я ничерта не пойму почему cpu расходует от 400 до 1.2 гб на вкладку, почему сайты насилуют процы офисных компов, какого черта каждый Зий сайт позволяет себе загрузку по 20+ секунд.
Почему мне разработчики сайта говорят что запросы с длительностью больше секунды это абсолютно нормально (ну там же целых 100 тысяч строк!!!).
Какого черта Photoshop CS 2018 позволяет себе страшные лаги при отрисовки действий кисти в примитивнейшем макете на машине последнего поколения.

И это только то от чего я впадал в ступор самое последнее время. Сдурели чтоли все вокруг?

khim 22 сентября 2018 в 04:10

+2

> И это только то от чего я впадал в ступор самое последнее время. Сдурели чтоли все вокруг?

Нет. Просто упёрлись в физические ограничения. До примерно 2005-2010 года наладалась идиллия: каждые год-два разработчики софта делали свои поделия вдвое менее эффективными. И каждые же год-два процессоры становились вдвое быстрее, а память — вдвое дешевле.

А потом… случилось страшное: память по-прежнему дешевеет, а процессоры… больше не ускоряются. И замедление, которое раньше было незаметно — теперь приводит к тормозам. Но индустрию-то так просто не перестоить!

Потому, в качестве паллиатива, начали убеждать пользователей «не верить глазам своим» и делать вид, что всё вокруг не тормозит. Опять-таки переход с HDD на SSD дал ещё несколько лет передышки… однако, похоже, резервы уже исчерпаны — и разработчикам скоро придётся думать об эффективности… но за столько лет этот орган у них атрофировался за ненадобностью!



BD9 22 сентября 2018 в 06:31

0

> память по-прежнему дешевеет, а процессоры… больше не ускоряются

Всё намного хуже: цены — растут, скорость — даже иногда снижается
Дефицит 14-нм процессоров Intel растёт вместе с розничными ценами
Intel вернулась к 22-нм нормам при производстве новых чипсетов
Samsung сократит производство памяти, чтобы не снижать цены



Kwisatz 22 сентября 2018 в 13:31

+2

Вот сейчас набирал сообщение вхроме, сдался перешел в FF. Я обгонял отрисовку символов на полстроки, ну да у меня машина загружена, но не настолько же.

Мне кажеться еще очень сильно роляет особенность веб разработки. По сути всем же плевать какой у них сайт. На нормальный денег жмут, идут заказывают за 3к, понимают что он не работает, идут заказывают за 20к, 50,100,500. Но ничего же не меняется. SEOшники давно всех убедили что они полезны, посему уникальный контент никто не создает, сайты деградируют до бесполезных лендингов, за нормальный дизайн для которых тоже никто не хочет платить (хотя уж где где а на лендингах UI должен быть на уровне «бог»), как итог на производительность всем плевать ибо все равно посещаемость полтора человека в час.

А вот когда вдруг набирается посещаемость тут начинаются проблемы, идут костыли, которые вроде первое время спасают, а когда не спасают где взять специалиста, который собственно сможет помочь? Я вот раньше любил получать денежку за полировку бд, но сейчас желания нету, потому как вовремя никто не хочет этим заниматься, а когда они возвращаются с полумертвым сервером я тупо выставляю барьерные цены (ибо сложность работ возрастает многократно, как мне найти хреновые запросы, если сервак уже захлебывается в прерываниях).

Разработчики деградируют вслед за всеми этими тенденциями, 10 слоев абстракции, умные слова, реактивное программирование, а без лага отрисовать табличку на 20 строк мы не можем, ага. ORM, архитектура, тщательное внимание даже к файловым системам и вдруг, тадааа, 100 тысяч строк это уже много вдруг стало. И SQL запросы дольше 1 сек вдруг стали нормой. Да, я буду хранить данные прямо в DOM, потому что не вижу смысла их дублировать и наворачивать дополнительные виртуальные структуры. Нет я не буду использовать ORM, потому что ускоряя разработку, она создает проблем больше, чем решает. А что такое фреймворки я вообще не знаю и знать не хочу (нет правда, так и не понял смысл этого слова за много лет)

Доступные VPS делает эту ситуацию еще хуже. Месяц назад вот разработчики сайта-каталога, заполняющегося парсерами, не моргнув глазом запросили VPS за 50 баксов/мес для размещения оного. Я сначала полчаса офигевал от цены предложенной, поскольку я за более мощную VPS плачу 9 евро, потом офигевал от требуемых ресурсов, затем офигевал от запроса гигабитного подключения (чтоб картинки быстрей загружать, ага (когда спросил про modified-since, меня не поняли)).

Теперь у меня мерзкое послевкусие от осознания того насколько все запущено. У меня сейчас VPS непрерывно загружает/обновляет/выгружает данные из целого ряда crm/erp систем и при этом практически покуривает от нефиг делать, ибо доступных в наше время более чем достаточно для решения весьма широкого спектра задач. Да право слово, когда я занимался вам-сайтами у меня на IMB x346 с 16 гб оперативной памяти и без SSD спокойно и быстро работал сайт на серьезной нагрузке, в базе которого количество строк на таблицу доходило до 750 млн.

Зы вот глаз вопиющего в пустыне
ЗыЗы простите, накипело)



arielf 22 сентября 2018 в 04:10

0

> Это не инженерная работа. Это просто ленивое программирование.

Ах чувак, как же ты прав!

ankh1989 22 сентября 2018 в 06:48

+3

Хорошая тема. Да, и насчёт Хабра. Как написать администрации? Упомянуть @alizar? Почему комментарии так безбожно тормозят? Неужели отрендерить 1000 комментов это такая сложная задача? Почему когда я их прокручиваю (хром, винда), оно тормозит так будто биткоины майнит и почему когда я прокручиваю огромный RFC всё летает? Возьмите своего лучшего JS кодера и дайте ему месяц попрофилировать этот тормозной кусок говнокода. Выкиньте весь этот хитрозакрученный CSS и напишите тупо в *лоб* с глобальными стилями, но так чтобы не тормозило. Забодало.



Am0ralist 22 сентября 2018 в 10:49

0

> Как написать администрации? Упомянуть alizar?

Эм… а через «помощь» -> "обратная связь" не пробовали?
Ну или в одну из прошлых тем администрации заглянуть? Там даже указаны кто есть кто на хабре…
На крайний случай, вызвать Boomburum'а? Зачем обычного автора статей то звать?



Kwisatz 22 сентября 2018 в 13:41

+2

Я просто оставлю это тут



arielf 22 сентября 2018 в 21:47

–1

О Боже мой! Какой ужас! Прав был Денис Драгунский — миром правят страх и ужас!

> – Миром правят, – говорил учитель, – две вещи, две силы, две главные пружины. Это страх, – он вдруг на секунду запнулся, словно бы позабыл слово, – это страх, и… и…
> – И любовь? И отвага? И честь? – наперебой заговорили ученики.
> – Нет, дети. Страх и ужас.

altrus 23 сентября 2018 в 06:13

0

Биг бразер следит за каждым твоим кей апом
И делает это риал тайм и за твой счет.

EvilFox 22 сентября 2018 в 15:18

+1

> Почему комментарии так безбожно тормозят? Неужели отрендерить 1000 комментов это такая сложная задача? Почему когда я их прокручиваю (хром, винда), оно тормозит так будто биткоины майнит и почему когда я прокручиваю огромный RFC всё летает?

Я ответил тут почему.
Вкратце: на хабре очень много не оптимальных стилей.

EvilFox 22 сентября 2018 в 16:20

+1

Главные источники тормозов при прокрутке элементы с position: fixed
По большей части чтобы устранить тормоза прокрутки наберите в консоле (через F12):

```
$("#xpanel").css("transform", "translateZ(0)");
$(".layout__elevator").css("transform", "translateZ(0)");
// можно ещё:
$(".js-ad_sticky_comments").css("transform", "translateZ(0)");
```

С полем ввода сложнее, я пока не понял какие стили гробят.



Kwisatz 22 сентября 2018 в 16:46

+4

Можно еще при прокрутке всей странице задавать mouse-events:none;

Кстати я отписал таки в тп хабра, цитирую ответ: «Вы можете привести пример ресурса на котором статьи с 1300 комментариев загружаются на ваш компьютер без задержек?»

Я конечно могу быть не прав, извините, но меня чуть не вывернуло от такого ответа от тп именно хабра.

EvilFox 22 сентября 2018 в 17:06

+3

> Можно еще при прокрутке всей странице задавать mouse-events:none;

Да, это тоже можно, но в данном случае не обязательно, достаточно обойтись стилями выше и тормоза при прокрутке сразу уходят.

А вот тормоза с полем ввода без рефакторинга не избежать, ибо проблема даже не в количестве тормозных селекторов (а их реально много), а в самом по себе огромном количестве селекторов (даже те селекторы которые не используются на странице всё равно добавляют тормозов).

> Кстати я отписал таки в тп хабра, цитирую ответ: «Вы можете привести пример ресурса на котором статьи с 1300 комментариев загружаются на ваш компьютер без задержек?»
>
> Я конечно могу быть не прав, извините, но меня чуть не вывернуло от такого ответа от тп именно хабра.

«Круто», что тут ещё сказать.
@deniskin, это норм?



Kwisatz 22 сентября 2018 в 18:01

+3

А на вопрос о тормозах в textarea вот так: «Здравствуйте! У нас такая картина не наблюдается. Вы уверены, что она не имеет локальный характер? Больше похоже на некорректную работу какого-то расширения.»

Самое паршивое, что я такие ответы вижу минимум раз в неделю. И каждый гребаный раз «У нас такая картина не наблюдается» и начинается многодневное выяснение, что я не осел. Чуть легче от мысли что обычно разговор идет об офисных компах, достаточно слабых. Но моя домашняя машина на порядки мощнее и тут уже мое лицо стало примерно таким O_O

EvilFox 22 сентября 2018 в 18:35

+2

Они наверное в лисе сидят (в ней не тормозит ни прокрутка ни поле ввода) и посмотреть в хроме (подвержены все на blink) им судя по всему лень (не в их компетенции).
Да стандартная ситуация когда некомпетентные люди отфутболивают.

sumanai 22 сентября 2018 в 19:32

+4

> Они наверное в лисе сидят

О, спасибо, а то в упор не понимал, о каких вы тут тормозах пишите.

Kwisatz 22 сентября 2018 в 21:08

+2

Наверняка, а еще на i7 последнего поколения. А комменты тут пролистать лениво видимо, тут уже раз 20 это упоминалось. Я сразу написал что в лисе проблем нет) Хотя отрисовка при резком скролле на статью кошмарно долгая.

Но лень таки видимо, да:

> Речь о том, что, при наличии 1300 комментариев с системой навигации по ним, довольно сложно реализовать загрузку страницы и переход к конкретному комментарию без задержек с загрузкой, тем более, что нам не известны скорость и качество соединения пожаловавшихся пользователей.
>
> Проблемы с фпс в textarea формы комментария к указанной статье мы не наблюдаем, поэтому склоняемся к тому, что ее порождает ваше локальное окружение (ОС, память, загруженность процессора и т.д.).

JTG 27 сентября 2018 в 17:13

+2

> Вы можете привести пример ресурса на котором статьи с 1300 комментариев загружаются на ваш компьютер без задержек

news.ycombinator.com/item?id=18064537

daleraliyorov 27 сентября 2018 в 17:42

0

Здравствуйте, Вячеслав!

Спасибо, что обратили наше внимание на проблему с комментариями. Мы возьмемся за нее и, как только будут результаты, я расскажу о них здесь же.

И простите за неприятный опыт общения с нашей техподдержкой. Это ошибка. Мы над этим тоже поработаем.

Kwisatz 27 сентября 2018 в 21:46

0

> Спасибо, что обратили наше внимание на проблему с комментариями. Мы возьмемся за нее и, как только будут результаты, я расскажу о них здесь же.

Шикарно. Не завидую пользователям хрома ибо у меня нервы сдали за 10 сек при попытке написать сообщение.

> И простите за неприятный опыт общения с нашей техподдержкой. Это ошибка. Мы над этим тоже поработаем.
>
> Ничего страшного, раз это не политика партии.

Am0ralist 28 сентября 2018 в 10:36

0

У меня вчера просто перестали прогружаться комменты на стрингах)
А в фф — ну да, долго страничка загружается. Зато потом — летает.

botyaslonim 28 сентября 2018 в 14:06

0

Возьмитесь, пожалуйста! Проблема застаревшая, я тоже писал о ней, но безрезультатно. Вообще, хорошая практика — разбивать комментарии на страницы. На Хабре часто публикации набирают по несколько сотен комментариев, и листать такую огромную простыню неудобно в любом случае

androidovshchik 22 сентября 2018 в 08:38

+1

Занимаюсь разработкой приложений для андроид
Бывает выступаю волшебником в глазах заказчиков, присылая apk размером менее 1 мб
А иногда просят специально увеличить размер apk, считая не солидным менее 2-3 мб

Am0ralist 22 сентября 2018 в 12:04

+1

Сегодня я долго смеялся.
Ибо сегодня вышел новый DosBox для которого 1,5 мб — инсталятор, а распакованный он весит 4,3 мб.
> А причина, почему я смеялся:

lieff 22 сентября 2018 в 13:10

0

1.6mb — и можно добавить запись игры в mp4 и стримминг на твич:
github.com/lieff/minih264
github.com/lieff/minirtmp

Когда обычно просто подключают ffmpeg или gstreamer. Причем это будет или зависимость, если это софт из репы линукса (это еще может быть оправдано ради качества/скорости кодирования), или просто скомпиленная дефолтная версия для винды\андройда\мака, чтобы неиспользуемые кодеки порезали и сократили размер до ~мегабайта — не видел такого.

iDm1 22 сентября 2018 в 22:08

0

Так и ffmpeg тоже можно собрать минималистично конкретно под ваши нужды. Приходилось впихивать его во FLASH память роутера для стриминга звука в Opus, занимал в памяти около 900 килобайт.

lieff 22 сентября 2018 в 22:33

0

Так я про это и говорю, порезать кодеки в ffmpeg довольно легко
Оптимизация за 5 минут, но почему-то даже для мобильных приложений, постоянно встречаю в apk в jniLibs/armeabi-v7a/ полностью собранные _все_ библиотеки от ffmpeg (когда используется только h264). Хотя казалось бы трафик пользователя надо экономить.

darksshvein 22 сентября 2018 в 12:24

0

рейтинг инструментов программирования увидеть хочешь ты!

tBlackCat 22 сентября 2018 в 13:05

+1

> Для примера, современные автомобили работают, скажем, на 98% от того, что физически позволяет нынешняя конструкция двигателя.

Вот именно — нынешняя. Но почему она считается пределом совершенства? На самом деле она далека от идеала. Возьмём предельный случай — иной метод извлечения энергии из топлива, например аннигиляцию и неважно, что это дело отдалённого будущего. В этом случае окажется, что автомобильный двигатель работает сейчас на уровне программного обеспечения:
> Только в программном обеспечении считается нормальным, если программа работает на уровне 1% или даже 0,01% от возможной производительности.

Если не ещё менее эффективно.
Но по сути Вы правы в ПО. Чем более старый софт, тем эффективность работы выше и не только в масштабах решаемых задач, но и в целом. Чем ближе к нашему дню, тем бОльший перекос в угоду первенства и прибыли. Пусть программа и малоэффективна, но её фирма выбросила первой на рынок и она снимает сливки. Вторая же будет долгое время питаться крохами, пока её суперэффективная программа будет раскручена и признана.
Это не метод современного бизнеса, да и кушать надо всем. Программистам тоже.
Порочная тенденция, но это реалии рынка.

xFFFF 22 сентября 2018 в 13:15

+2

Согласен с автором. У меня на старом телефоне было много разного софта(ну как у всех). В течении последних четырех лет количество софта не изменялось, но работал он все медленнее, и занимал всё больше места. Недавно купил новый телефон, и все тот же софт снова быстро работает. Получается замкнутый круг.

5oclock 22 сентября 2018 в 21:14

+4

«В браузерах накопилось столько пограничных ситуаций и исторических прецедентов, что никто не осмелится писать движок с нуля.»

Конкретно в браузерах меня донимает то, что при загрузке страницы они постоянно производят изменение области просмотра (ViewBox) — то, что сейчас на экране.
Щёлкнул по ссылке — на экране появилась страница: текст. Начинаешь читать — оказывается там где-то выше ещё «шапка» картинкой догрузилась — надо немного сместить текст. Потом где-то реклама догрузилась, фрейм какой-нибудь — и всё это вызывает изменения в области просмотра, хотя само не видно на экране.
Почему нельзя «загрузить» всё это за экраном? Ну будут бегунки (горизонтальный, вертикальный) уменьшаться в размерах и смещаться туда-сюда, но при этом можно спокойно считывать информацию со статичного, не дёргающегося экрана.

IRainman 23 сентября 2018 в 14:06

0

Всё это из-за «оптимизаций» по скорости загрузке, не имеющих никакого отношения ни к удобству использования ни к эффективности, хуже того там ещё и DOM перестраивается когда страница «уже давно загрузилась» и прочая адища происходит. Про то как такие «оптимизации» сжирают батареи, особенно у смартфонов даже говорить не хочу. Зато маркетинговый эффект работает: пользователю сразу открылось содержимое, то что оно ещё полминуты будет ползать по экрану, подвисать и приводить к зависаниям звука в соседней вкладке уже мало кого волнует.

Kwisatz 23 сентября 2018 в 16:14

0

Вот поэтому лично я рендерю страницу сервером, без новомодных приколов. А то нынче круто пошло, загрузить данные, преобразовать в объекты, построить копию дом дерева, воткнуть на страницу, навешать хендлеров и вуаля, табличка на 20 строк работает медленней чем моя на 1000+

illitaexperta 23 сентября 2018 в 00:41

+6

Вы тут часто обвиняете капитализм, но это не корень проблем. Корень проблем нынешнего IT-сообщества — **повально низкая квалификация разработчиков**.

Разработчиков низкой квалификации стало так много, что произошел переход из количественного в качественное. Только вот, это значит не то что качество новых разработчиков повысилось, а то что квалификация среднего разработчика, опустилась на качественно более низкий уровень. Процесс зашел так далеко, что само понятие квалификации ощутимо размылось.

Разделение на Junior, Middle, Senior больше ничего не значит. Сейчас можно совершенно не уметь проектировать, придумывать собственные решения и подходы, решать нестандартные практические задачи, выбирать инструменты и технолгии под задачу или писать свои, да и просто думать головой а не догмами и не паттернами, но при этом зваться Senior Android Developer или Senior Full Stack Developer. Вариться в тусовке таких же фейковых сеньеров, а тысячи еще более фейковых мидлов и джунов будут вам подражать и ориентироваться на вас, затягивая сообщество еще дальше в воронку деградации.

То что такой Senior Android Developer посыпется при попытке написать приложение хотябы для iOS, *ведь это не его же профиль, Swift он не знает, а Objective-C вообще почти Си, а это для него уже оверскилл*, никого не смущает. Ни коллег, потому что они сами такого же уровня, ни менеджмент, ведь менеджмент ориентируется на мнение ведущих разрабов в своей конторе. А кто эти ведущие разрабы? Такие же фейковые синьеры. Это значит что стало намного меньше разрабов, умеющих хотя бы правильно эту квалификацию оценить.

Умение решать практические задачи подменилось умением набрать побольше фреймворов с модными названиями себе в проект. Умение проектировать архитектуру подменилось заучиванием паттернов и аббриватур вроде MVP и спорами про ~~количество ангелов на кончике иглы~~ про что должна знать Model, про что View, а про что Interactor. Умение выбрать язык и технологию под задачу стало такой редкостью, что HR искренне удивляются узнав что такое вообще возможно. А самое главное в работе программиста умение — думать головой давно подменилось банальным следованием догмам от лидеров мнений и выбором из двух неуместных крайностей.

illitaexperta 23 сентября 2018 в 01:16

+11

Если же задаться целью понять почему так получилось, то нужно обратиться к истории.

Причина деградации ровно одна — **грубые ошибки в проектировании популяных технологий**. Не очевидно, но я объясню.

Например, возьмем хороший пример — Delphi.

У Дельфи очень низкий порог входа — т.е. научиться клепать на ней формы мог почти кто угодно. Дельфи имела в свое время очень обширное коммерческое применение — т.е. под нее было много вакансий, на этом рынке крутилось много денег. Но при этом она была прекрасно спроектирована. Благодаря этому, даже у криворуких разрабов получались более менее качественные приложения, они не занимали сотни мегабайт, не требовали жирных фреймворков и виртуальных машин и быстро работали на любой системе. При этом новоприбывшие разрабы учились более менее хорошим подходам и дальше либо становились нормальными разработчиками, либо оставались плавать в собственном соку в рамках Delphi-экосистемы и никому не мешали.

А теперь возьмем плохой пример — **клиентский веб-стек**. С него все и началось.

Тут все похоже. Низкий порог входа — куча джунов. Сайты — это новый огромный рынок — востребованность разрабов, еще больше джунов. Но, клиентский веб-стек это набор грубых ошибок и костылей, описывать их по двухсотому кругу смысла нет. Что получилось в результате — прорва разрабов, которые не умеют нормально программировать, и при этом даже этого не понимают и считают себя Senior Full Stack Developerами. Которые впитали костыли и кривые принципы разработки и думают что так и надо. Которые думают что прелюбуный margin: 0 auto это такой прием разработки, а не ужасающий костыль. Но хуже всего то, что востребованность таких разрабов привела к тому, что они еще и не хотят учиться другим стекам и технологиям.

Пока фронтенд-js-комьюнити варилось в собственном соку, все было более менее хорошо. Все уже привыкли что браузеры тормозят, а сайты глючат. Но в какойто момент js-комьюнити начало переливаться в другие сферы. Сначала на бекенд — node.js, затем на мобилки — всякие fonegap, а теперь и на декстоп — Electron.

> Кстати, кто тут утверждают про что плохого в Electron? Это же просто абстракции!

Далее, эта волна вышла на новый уровень, разрабы с js-бекграундом начали лезть в другие языки со своим костыльными практиками, в Android, в Go. Вы только почитайте статьи про Go, сколько там слепого догматического обожания от бывших джиесеров.

Android тоже добавляет масла в огонь. В конце нулевых под Android восновном писали разрабы старой школы, которые мирились и обходили откровенно ужасные косяки в проектировании системы. Но потом пришли новоприбывшие бывшие школьники, смешались с выходцами из js-комьюнити, и решили что все то говно, что есть в Android SDK это манна небесная.

К нынешнему моменту уже выросло целое поколение андроид синьеров, программирующих исключительно добавлением фреймворков и рассуждающее неадекватными паттернами вроде MVP и не понимающих что уничтожение Activity при повороте экрана это кромешный писец, а не нормальное решение.

И пути назад нет. Бизнесу нужно решать свои задачи, и ему под это дело нужны разрабы. Маховик продолжит крутиться и будет плодить все новых и новых фейковых синьеров. И эти синьеры будут дальше развивать и создавать новые технологии, аналогичного хреновго качества. А эти технологии уже будут плодить еще более криворуких разрабов. И так далее.
Круг замкнулся, мы сейчас имеем фактически новую школу разработчиков — которая раз за разом воспроизводит деградировавшие кадры.

Остается надеяться что этих фейковых сеньеров хватит лишь на создание очередного безполезного фреймворка с модной иконкой, и этот порочный круг разорвется

Kwisatz 23 сентября 2018 в 16:48

+1

Дело не только в Js. Процесс глобален. У меня вот не сложились отношения с ORM, я десятки раз видел ка кради пары запросов люди подцепляют здоровый фреймворх, хрен кладут на индексы, а потом сражаются с тем что получилось. Я так не смог, не могу я использовать технологию, с которой нужно постоянно сражаться. Но теперь я себя чувствую белым мамонтом в пустыне, потому что вокруг сплошняком в базах какая то дичь, поверх этой дичи жуткий оверход слоев абстракции и куча всевозможных умных гайдов о необходимости кеширования и «глубокого понимания устройства памяти и файлового кеша».

Раньше когда мне надо было на яве написать простенькую утилитку, то все гуглилось довольно легко и просто и можно было почитать обсуждения интересных моментов. Сейчас если окунуться в непопулярные вещи типо поиска нечетких дубликатов с попутным определением кодировки, то искать нюансы можно бесконечно: поисковик забит «мини-гайдами» о том как надо взять спринг, обязательно в полной его иностаси, наклепать быстро пару классов, воткнуть @магическиеШтуки и все сразу будет работать. Причем по статьям, с помощью прямых вопросов, а иногда даже и с помощью документации, не всегда можно выяснить что делают эти инжекторы.

Вообще довольно показателен тот факт, что бОльшая часть php-фреймворков просто не умеет адекватно работать с prepared запросами. Я уже раз 10 сталкивался с тем, что людям prepared сэкономит до трети расходов на запросов, но фреймворк надо допиливать, потому что он не в курсе что у pg можно и нужно получить список подготовленных запросов для соединения.

sentyaev 23 сентября 2018 в 18:11

0

До сих пор количество вакансий больше чем программистов, от этого мы имеем такие высокие зарплаты и их постоянный рост. На рост зарплат дополнительно оказывает влияние глобальный рынок, т.е. границы между странами размыты и все хантят всех. Из-за этого программисты прожигают бюджеты с космической скоростью.
С другой стороны производительность железа все еще растет, не такими темпами как 10 лет назад, но все же.
Поэтому главное это выпустить продукт как можно скорее и начать получать доход, чтобы либо платить было чем, либо поднять инвестиций для следующей пачки фич.
В этом смысле Electron — идеальная технология, быстро и относительно дешево.

Нет смысла расстраиваться по этому поводу, рынок все расставит по местам.

Да, нужно заметить, что тренд на веб-приложения заканчивается, все уходит в мобильные приложения. Тренд на увеличение производительности тоже проходит, а приходит тренд на увеличение количества ядер и рост памяти.

kpakozz96pyc 28 сентября 2018 в 17:16

0

Можно я вас обижу, хотябы удаленно? Специально вспомнил пароль от хабра, чтобы плюснуть ваш комментарий. Только сегодня у меня горело от VS Code, которая вальяжно раскинулась на 2ГБ оперативки в простом проекте, в то время как Sublime на нем же не тормозил, занимая 400мб со всеми плагинами. А рядом был Slack в котором 5 каналов чата тупили занимая 350 мб и Telegram который не потел, занимая меньше 30 мб с 50+ каналами.
Да даже VS2017 занимает меньше гига в памяти! А Code — которая позиционируется как её легковесная версия, как быстрый и легкий аналог, запускает почти 20 процессов и занимает в 2 раза больше.
Мне кажется JS — это царь Мидас современной разработки. Я никогда не пойму, как можно на языке, который изначально создавался для домохозяек писать что-то сложнее прототипа.

popov654 18 октября 2018 в 09:31

0

Имхо, сам JS не виноват. Можно писать быстрый и эффективный код на JS, который не будет тормозить в целевом браузере. Проблема в другом — современные версии браузеров слишком прожорливые и слишком дофига весят (Electron туда же).

Druu 18 октября 2018 в 11:35

-2

> Только сегодня у меня горело от VS Code, которая вальяжно раскинулась на 2ГБ оперативки в простом проекте,

Позволю себе посоветовать вам лайфхак: вытаскиваете лишнюю планку памяти и кладете на полку. Тогда ее никто не будет занимать.

arieff 24 сентября 2018 в 17:37

0

Прекрасно написано! А вы бы не вынесли ваши мысли в независимую публикацию? У меня была похожая, но с немного иным упором. habrahabr.ru/post/225623 И новая версия на английском. mystudentvoices.com/on-higher-education-programmers-and-blue-collar-job-383ce12f64ba

botyaslonim 28 сентября 2018 в 14:02

0

Общественное устройство именно что корень проблем. Именно оно требует в том числе и определённых специфических подходов к образованию, овладеванию профессий. Если рыночку требуются миллионы разработчиков, чтобы клепать примерно одно и то же, клепать, используя 100500 разных инструментов, то нас появляется миллион разработчиков низкой базовой квалификации, способных быстро, на лету осваивать все эти тысячи фреймворков. Только и всего. Люди сами не выбирают быть плохими. От них

просто требуется бежать, не оглядываясь и ни о чём не думая. Иначе работу получит конкурент

popov654 18 октября 2018 в 09:32

0

А кто выигрывает от наличия 10500 разных не совместимых друг с другом инструментов? Даже если они просты и интуитивны — зачем так много? Я уже не говорю о том, что в половине случаев они вообще не нужны, и всё без них нормально реализуется.

botyaslonim 18 октября 2018 в 15:15

0

Выигрывает тот, кто успешно продаёт свой стартап

popov654 23 октября 2018 в 03:20

0

Каким образом это ваше утверждение связано с наличием 100500 инструментов и фреймворков?..

Lucky_Student 23 сентября 2018 в 01:09

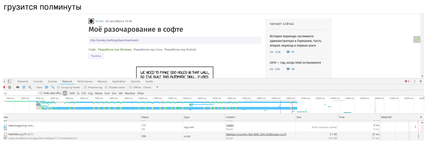
+2

Не логинился на Хабре/Гиктаймсе около двух лет. Просто читал и не комментил последнее время. Ради того, чтобы поставить «плюсик» — залогинился. Полностью согласен с автором, правильный пост.

PsyHaSTe 23 сентября 2018 в 04:32

+3

Тем временем статья про падение нравов разработки и написание тормозного и плохого софта лагает и грузится полминуты



Crandel 23 сентября 2018 в 12:58

0

В огнелисе загрузилась за 2,5 секунды

Заголовок спойлера

ilya81 23 сентября 2018 в 07:28

0

Местами есть преувеличения, например, на счёт загрузки телефона за 1 секунду. Я вообще не знаю ни одной GUI ОСи, загружающейся за 1 секунду, а CLI и псевдографика по мне слишком большая жертва ради быстродействия. Да и GUI бывает разным — OpenBox загружается менее 5 секунд, KDE — всё ж подольше, и я без крайней необходимости обычно выбираю более совершенный GUI в KDE. С играми вообще особая история, специализированные процессоры, в т. ч. GPU, всегда эффективнее, но для прочих целей специализированные процессоры на пользовательских компах вряд ли приживутся. В Windows, впрочем, есть проблема стандартизации — скажем, WPF криво работает с видеофермом, а особой альтернативы использования GPU вне игр не прижилось, не будет ж каждай писать свою прослойку, а GDI для графики использует CPU. Про Windows 95 вообще в 30 Мб отдельная история, Windows так и тянет тяжёлое наследие оптимизации с разделяемыми DLL и т. п., в итоге для их обновления теперь пришлось использовать транзакции, а в этом случае 30 минутам на обновление удивляться не стоит. Автор знает, сколько занимает закрытие банковского дня? А знает почему? Вот они транзакции. Хотя, если перейти к теме приложений, кривое использование ORM порой и впрямь создаёт транзакции почём зря. А всякие Docker'ы и прочие Kubernetes — это решения для создания массового создания крупных систем, что раньше не применялись.

khim 24 сентября 2018 в 05:11

+1

> Я вообще не знаю ни одной GUI ОСи, загружающейся за 1 секунду

RISCOS на Raspberry PI примерно столько грузится с момента когда ещё загрузили в память. Считать нужно именно это время, так как изначально RISCOS (как и большинство OS, разработанных в 80е) была рассчитана на то, что её будут зашивать в ROM. Соответственно загрузка занимала секунды три — а с тех пор компьютеры стали в 1000 раз быстрее. Наверное и сейчас её можно было бы прошить в ROM на Raspberry PI — но с этим никто не хочет заморачиваться.

> Автор знает, сколько занимает закрытие банковского дня?

Да, знает, я думаю.

> А знает почему?

Потому что для того, чтобы исправить хоть что-то нужно собрать 100500 бумажек и никто не будет менять архитектуру, даже если она безбожно транжирит ресурсы? Так об этом и статья.

> Вот они транзакции.

О каких, блин, транзакциях мы говорим? Зачем вам транзакции **внутри** процесса обновления? Делаем снапшот, делаем обновление, если не получилось — возвращаемся назад. Нафига вам в этом процессе ещё внутри какие-то транзакции?

> А всякие Docker'ы и прочие Kubernetes — это решения для создания массового создания крупных систем, что раньше не применялись.

Да применялось это. Начиная с 70х годов OS/360 могла вкладываться внутрь OS/360. А туда — можно было ещё одну OS/360 вложить. И там — хоть 100 раз, пока памяти хватит. System/370 тем и отличалась от System/360.

Но для этого нужен правильный дизайн, да. Тот факт, что, когда проблема была диагностирована, Intel и AMD не исправили архитектуру (как сделала IBM в System/370), а наворотили ещё один уровень костылей вызывает, если честно, «чувство глубокого недоумения» — но очень хорошо ложится в канву обсуждаемой статьи.

ilya81 24 сентября 2018 в 07:47

0

> О каких, блин, транзакциях мы говорим? Зачем вам транзакции внутри процесса обновления? Делаем снапшот, делаем обновление, если не получилось — возвращаемся назад. Нафига вам в этом процессе ещё внутри какие-то транзакции?

Я не о восстановлении. Если ничего не путаю, в MSI оно вообще отдельно реализуется и опыт показывает, что его полноценной реализацией обычно никто не заморачивается. Но есть такие специфические двоичные файлы, как system.dat и user.dat. И Windows построен так, что информация о DLL должна быть записана ещё и в них. Если запись в них осталась, а файл удалён — потом все использующие приложения будут падать, поэтому нужно обеспечивать атомарность таких действий с учётом того, что электричество могут отключить в любой момент.

> Да применялось это. Начиная с 70х годов OS/360 могла вкладываться внутрь OS/360. А туда — можно было ещё одну OS/360 вложить. И там — хоть 100 раз, пока памяти хватит.

Я вообще не об этом. В те годы применялись ERP? Не в теории, а на практике? Реально внедрённых их во всём мире было больше, чем пальцев на одной руке?

khim 24 сентября 2018 в 08:20

+1

> Если запись в них осталась, а файл удалён — потом все использующие приложения будут падать, поэтому нужно обеспечивать атомарность таких действий с учётом того, что электричество могут отключить в любой момент.

Для этого совершенно не обязательно устраивать транзакции внутри транзакций. Достаточно точки восстановления перед началом процесса и после.

> > Да применялось это. Начиная с 70х годов OS/360 могла вкладываться внутрь OS/360. А туда — можно было ещё одну OS/360 вложить. И там — хоть 100 раз, пока памяти хватит.
>
> Я вообще не об этом. В те годы применялись ERP? Не в теории, а на практике? Реально внедрённых их во всём мире было больше, чем пальцев на одной руке?

Вопрос под тем, что вы понимаете под ERP. Сам по себе ERP — это новый термин, до него был MRP, который к 70м уже использовали, как гласит Wikipedia «большинство фирм США с годовым объёмом продаж готовой продукции свыше 15 млн долларов; в Великобритании — каждое третье производственное предприятие».

Собственно с подобных систем IBM и начиналась (в 1911 году, напомню). Конечно тогда гигабайтов и гигагерц не было, выглядело всё немного по другому, но… считало почти то, что умеют современные системы… просто навыки у обслуживающего персонала нужны были определённые…

popov654 18 октября 2018 в 09:40

0

> Если запись в них осталась, а файл удалён — потом все использующие приложения будут падать, поэтому нужно обеспечивать атомарность таких действий с учётом того, что электричество могут отключить в любой момент.

А разве не проще написать код, который во время запуска в случае обнаружения сбоя проверит эти файлы и пофиксит их? Или даже встроить это в мастер диагностики ошибок, и предлагать пользователям использовать его, если сбоят программы — можно даже иконку в трей поместить для этого, чтобы точно все нашли?

В итоге получим прирост скорости обновления, а стабильность не упадёт.

Akon32 19 октября 2018 в 13:26

0

Пользователю нужно решать свои задачи. Пользователю нужно, чтобы его данные не повреждались. Запуск какого-то *"мастера диагностики ошибок"* после каждой неожиданной перезагрузки — совершенно не то, что хочет делать пользователь. Проблемы при сбоях — это проблемы ОС, а не пользователя.

Текущий подход решает проблемы при сбоях автоматически, и в основном корректно. Подобный подход работает в большинстве СУБД. Зачем что-то придумывать, что добавит проблем пользователям?
О каком приросте скорости идёт речь, когда пользователя заставляют вручную исправлять реестр??

popov654 23 октября 2018 в 03:24

0

Я не говорил про «вручную», вы что-то придумали.

Вы правда не видите разницы между «подождать 1-2 минуты работу мастера, если его запуск реально будет нужен из-за повреждения файлов после внезапной перезагрузки», и «ждать лишние 10-12 минут при каждом обновлении»? Знаете, не все ставят обновления в конце дня, когда выключают компьютер и уходят из офиса. Я вот люблю перезагрузить и смотреть, как они ставятся при мне, чтобы потом оценить, что изменилось. Не говоря уже о том, что у меня виртуалка, и там вообще всё тормозит как не знаю что — а тут ещё этот дополнительный слой транзакций. Рукалицо просто.

В Windows 7 как-то жили без этого, ничего после внезапной перезагрузки не падало? Ну окей, допустим у меня один раз падало. Но велики ли шансы, что прямо во время обновления в доме пропадёт свет, если само обновление будет быстрым? Мне кажется, крайней малы. А такой мастер свёл бы проблемы вообще к минимуму — не нужно стартовать из рекавери, не нужно делать образы ОС сторонним софтом.

BigBeaver 23 октября 2018 в 09:25

0

Какому-нибудь Дебиану вообще пофиг будет, если перед «случайной» перезагрузкой не удалить ядро, менеджер пакетов и еще что-то важное.

Akon32 23 октября 2018 в 13:25

0

> Я не говорил про «вручную», вы что-то придумали.

Да вот же:

> встроить это в мастер диагностики ошибок, и предлагать пользователям использовать его

Т.е. пользователь должен будет (руками) какие-то непонятные кнопки нажимать, надеясь, что они запустят магию. Зачем требовать этих действий? Лучше сделать, чтобы оно как-то само.

> Я вот люблю перезагрузить и смотреть, как они ставятся при мне, чтобы потом оценить, что изменилось.

Не все это любят. Для части пользователей общение с компьютером — лишь необходимое зло, и тратить своё личное время, глядя на компьютер, они не будут.

> Но велики ли шансы, что прямо во время обновления в доме пропадёт свет, если само обновление будет быстрым? Мне кажется, крайней малы.

Число установок windows — сотни миллионов. Шанс того, что у кого-то свет таки отключится, этот кто-то потеряет данные и настрочит в бложик, какая плохая винда, довольно высок (примерно равен 1). Предполагаю, таких пользователей будут как минимум сотни. Во времена использования fat для системного диска примерно так и было. Сбой, переустановка, сбой, переустановка.
Обычно сбои всё-таки предотвращают, как могут, а не пишут костыли для их исправления.

popov654 23 октября 2018 в 17:28

0

> Т.е. пользователь должен будет (руками) какие-то непонятные кнопки нажимать

Окей, и правда я писал это как один из возможных вариантов. Ну пусть оно само тогда эти файлы проверяет при старте :)

> и тратить своё личное время, глядя на компьютер, они не будут

Пользователи — пожалуй, а как насчёт сисадминов в компаниях, задача которых — проверить работоспособность новой сборки, и что ничего не сломалось (например, не отвалился какой-то софт)? Пожалейте хоть их время…

> и настрочит в бложик, какая плохая винда

Так если при старте мастер исправления автоматически пофиксит те файлы, винда запустится. С чего тогда пользователю строчить что-то в бложик?

> Обычно сбои всё-таки предотвращают, как могут, а не пишут костыли для их исправления.

Ну знаете, механизм транзакций — такой же костыль для исправления сбоев. Только блестящий, красивый и модный :) И много где применяемый.

Alexeyslav 29 октября 2018 в 11:07

0

Транзакции это всё-таки не костыль а вполне логичное и надёжное решение — если что происходит непонятное, система остается в первоначальном виде. Костылём как раз является проверка неких файлов(кто будет держать актуальным этот список?) и их восстановление(откуда? где держать актуальные копии?). А если файл неучтённый, копии нет или она тоже повреждена/неактуальна/является причиной сбоя?
В винде есть давно уже ассистент по восстановлению после сбоев, только на моей памяти(одна неответственная железка очень часто падает) он НИ ОДНОГО РАЗУ не помог, только время занимает на старте часто требуя активных действий от пользователя(ага, на системе где ни клавиатуры ни монитора нет).

popov654 31 октября 2018 в 09:42

0

> кто будет держать актуальным этот список?

Очевидно, сама ОС. Актуальные копии можно делать периодически во время работы пользователя (когда ничего не устанавливается).

> он НИ ОДНОГО РАЗУ не помог

Так это не проблема концепции, а ужасная реализация :)

Alexeyslav 31 октября 2018 в 10:18

0

Сама ОС не знает где её файлы а где чужие, чтобы она могла отличить их кто-то должен пометить. Драйвера от сторонних производителей и их файлы тоже должны считаться частью ОС и она должна за ними следить? Не уследит за драйвером от материнки и всё…

BigBeaver 31 октября 2018 в 10:59

0

Как-то вы все усложняете. Это давно придумано и работает со времен XP, если не раньше.

Alexeyslav 1 ноября 2018 в 09:40

0

Да, и там используются списки файлов входящих в дистрибутив в том или ином виде. Причем на рабочей системе этот список не хранится и для «починки» нужно присутствие инсталляционного диска.

BigBeaver 1 ноября 2018 в 11:05

0

Список пакетов хранится локально. Кроме того, есть локальный кэш всех скаченных пакетов, если он не был очищен. А с диска никто уже со лет не ставит. Если есть интернет, то ест и все остальное. И если мы говорим о поломке (сбоях питания или типа того с последующим нарушением транзакционной целостности [или как мы это назовем?]) в момент обновления, то это автоматически означает наличие доступа к репозиторию (не важно, локальный он или какой).

popov654 1 ноября 2018 в 15:30

0

Не с диска, так с флэшки, не всё ли равно. Интернета во время установки обычно ещё нет, потому что на первом этапе нет драйверов на оборудование, включая сетевую карту. Во время обновления — технически он есть, но вряд ли используется, имхо.

staticlab 1 ноября 2018 в 15:41

0

А если к компьютеру физически нельзя подключить флешку?

BigBeaver 1 ноября 2018 в 15:54

0

Вы шутите. Конечно же интернет есть всегда.

Но я еще раз повторяю, что это не важно. Если мы говорим об установке, значит, есть доступ к пакетам. Точка.

Alexeyslav 1 ноября 2018 в 16:42

0

Если это оффлайн-установка, то доступ есть разве что только к одному пакету, который и устанавливается в данный момент.

BigBeaver 1 ноября 2018 в 20:04

0

> Только если вы его отдельно скачали. Но если вы устанавливаете один пакет, то и нарушиться целостность может только у этого одного пакета. В чем проблема-то?



popov654 31 октября 2018 в 11:16

0

> Драйвера от сторонних производителей и их файлы тоже должны считаться частью ОС

Должны. Если один из драйверов окажется повреждён (подпись или контрольная сумма), Windows скорее всего не запустится. Да и смысл запускаться в этой ситуации? Даже если она бы это сделала, был бы 99-процентный шанс получить синий экран.

> Сама ОС не знает где её файлы а где чужие

Знает, файлы ОС имеют специальную цифровую подпись, плюс ОС знает места их хранения.

И я возвращаюсь к тому, что предлагалось просто при старте добавить проверку корректности файлов system.dat и user.dat, эти файлы понятно как называются и понятно где лежат.

DrPass 31 октября 2018 в 14:29

0

> И я возвращаюсь к тому, что предлагалось просто при старте добавить проверку корректности файлов system.dat и user.dat, эти файлы понятно как называются и понятно где лежат.

Немного не понял вашу идею. Вы имеете в виду файлы реестра Windows 9x? Она их и так проверяла при старте, и запускала исправление, если что-то не так. Реестр в Windows 10 хранится в других файлах, но проверка при старте и восстановление при необходимости также есть.



popov654 31 октября 2018 в 16:39

0

Там выше написали, что якобы если в этих файлах что-то будет испорчено, то и в новых версиях будут проблемы. Не я автор этого утверждения, я просто цитирую)

DrPass 31 октября 2018 в 17:05

0

А, увидел. Ну, там проблема немного не такая, о которой писал автор комментария. Система действительно проверяет целостность реестра, но проверять, насколько корректны записанные в него данные, и не всегда возможно, и не всегда целесообразно. Например, зарегистрированных COM-объектов может быть тысяча. Если при старте каждую запись сверять, есть ли на диске соответствующая библиотека, то компьютер будет очень долго стартовать. Да и смысла в этом нет. Ну вот не нашлась какая-то DLL, которая используется в приложениях СуперОфис 2015, Переводчик 17 и МойЧатик. Что дальше системе делать, просить пользователя найти этот файл и положить в нужную папку? Наверное, нет. Поэтому она отдаёт принятие решения приложениям и пользователю. Если оно чего-то не нашло, то может или обработать ошибку и сообщить, или продолжить работу без этой компоненты (если такое возможно), или упасть и тем самым намекнуть пользователю, что неплохо было бы его переустановить.

popov654 31 октября 2018 в 17:59

0

Можно кстати задать вам вопрос из области практики (гуглить пробовал, беглый анализ за 2 минуты ответа не дал)?

Есть компьютер (не мой, у меня на работе в небольшой фирме). ОС — Windows 7 x64. На нём что-то случилось, и ряд программ не может запуститься, выдавая ошибку 0x000007b (вылета в синий экран нет, есть окошко). При этом пишет, что файл не является приложением, или что-то вроде того. Но Проводник при этом работает, ещё работает как минимум Хром.

Ещё я знаю, что в том помещении на всех компах обновляли версию Office, и при онлайн-установке временами отваливался интернет (правда, руководство говорит, что тот сбой пошёл несколько раньше момента, когда её стали обновлять). Я просто к тому, что есть некоторый специальный хак (его применяют многие антивирусы, включая Dr. Web), который временно делает все exe файлы «не приложениями», и их нельзя запустить. Возможно, онлайн-установщик Office в какие-то моменты его использует на короткое время — и возможно, установку аварийно прервали.

Как думаете, что это может быть? Косяк настройки групповых политик на запуск приложений? Под админом эта проблема тоже есть, не только под обычными пользователями.

DrPass 31 октября 2018 в 23:15

0

Это точно не политики, и очень маловероятно, что какой-то вирусный монитор (или вирус:-). Дело в том, что эта ошибка выбрасывается загрузчиком приложений не только если exe-файл модифицирован/поврежден, но и если повреждена любая из статически скомпонованных с ним dll'ок.
Вполне вероятно, что повреждена какая-либо из системных библиотек, которая используется рядом приложений. Кто не использует её, тот работает.
Я бы в данном случае начал с запуска утилиты проверки и восстановления системных файлов Windows, или просто бы откатился к предыдущей точке восстановления, если она есть.

popov654 1 ноября 2018 в 15:32

0

Точек увы нет почему-то (хотя комп включался регулярно, это учебный класс). Утилиту надо попробовать… Она в составе ОС, или надо диск ставить?

DrPass 1 ноября 2018 в 16:54

0

В составе ОС, hetmanrecovery.com/ru/recovery_news/how-to-recover-windows-system-files-sfc-dism.htm

Alexeyslav 1 ноября 2018 в 16:59

0

Сама утилита в составе ОС, но данные которые её требуются обычно на диске. А вообще это очень интересный вопрос, это раньше винда годами не обновлялась и установочный диск имел какую-то актуальность, а сейчас когда патчи ставятся каждую неделю, за несколько месяцев актуальность дистрибутива с которого была установлена система ставится под сомнение. Не сможет эта утилита восстановить системный файл если он был обновлён ранее каким-то из апдейтов иначе может сделать только хуже. В итоге что? Надо иметь где-то дополнительно копию всех обновлённых файлов… и после удивляемся почему свободного места становится всё меньше и меньше? А если и эти файлы повреждены? Получить ошибку считывания данных не обнаруженных схемами коррекции ошибок очень маловероятно но вполне возможно, на современных-то объемах данных. Я с таким за жизнь уже сталкивался 2-3 раза. Поэтому повредить файл просто его копируя с диска на диск маловероятно но МОЖНО(кто из вас сверяет контрольные суммы файлов до и после копирования?). А может, у вас ещё есть информация о происходящих ошибках чтения в RAM? а там они между прочим происходят гораздо чаще чем мы об этом можем подумать… а она используется в качестве буфера при передаче файлов с диска на диск, что увеличивает вероятность повреждения файла.

Если нет точек восстановления, то либо это зловред их отключил и уничтожил или сделано было из экономии на ресурсах сознательно.

DrPass 1 ноября 2018 в 23:24

0

> Не сможет эта утилита восстановить системный файл если он был обновлён ранее каким-то из апдейтов иначе может сделать только хуже.

Справедливости ради, сейчас уже не так, как было во времена WinXP. Нынче в 99% случаев она нормально справляется, и если ей вдруг чего-то не хватает, то сообщает об этом в логе, а не ломает систему.

popov654 2 ноября 2018 в 01:57

0

Была точка, одна. За 2 дня до текущего. А проблема возникла на две недели раньше (как минимум).

Alexeyslav 1 ноября 2018 в 09:47

0

Далеко не все файлы подписаны. К примеру *.INF хотя для системы они не менее важны чем сами драйвера. Ещё раз повторюсь — ОС как таковая НЕ ЗНАЕТ где находятся её файлы и места их хранения. В случае повреждения реестра где прописаны пути к файлам и системным папкам операционка будет в полной прострации даже если все файлы будут целёхоньки и лежать на своих прежних местах. Такую ситуацию может исправить только установщик у КОТОРОГО есть точный список всех файлов дистрибутива, и который может их восстановить и

прописать пути заново. При этом все сторонние файлы которые не входили в дистрибутив будут считаться посторонними не смотря даже на подписи, ибо что с ними может сделать установщик операционки? Он даже не знает где взять оригиналы для починки. Но это знает установщик соответствующего драйвера и т.п. Точно так же с обновлениями — каждый пакет знает только про те файлы которые подлежат обновлению и не более того.

sumanai 1 ноября 2018 в 20:23

0

> Далеко не все файлы подписаны. К примеру *.INF хотя для системы они не менее важны чем сами драйвера.

Но INF файлы должны быть подписаны для того, чтобы с его помощью можно было установить драйвер.

> В случае повреждения реестра

Есть его резервные копии, плюс соответствующие ветки реестра продублированы несколько раз для возможности быстрого отката, та самая «Последняя удачная конфигурация».



Alexeyslav 2 ноября 2018 в 02:01

–1

> Подписаны драйвера, а INF-фалы несут только информационную нагрузку, но очень важную — если они укажут на неправильный драйвер, хоть он и будет подписан но ничего не заработает.

DrPass 2 ноября 2018 в 02:44

+1

> Подписаны драйвера, а INF-фалы несут только информационную нагрузку,

Вы ошибаетесь. Во-первых, вендорская подпись драйвера Windows накладывается на все файлы в пакете, в том числе и на inf. Во-вторых, inf используется только во время установки драйвера, и если с ними что-то произойдет после установки, на работу ОС это уже не повлияет.

BigBeaver 31 октября 2018 в 10:40

0

У нас уже есть журналируемые файловые системы. И что-то не заметно чтобы они тормозили комп. У нас уже есть менеджеры пакетов, которые тоже умеют в журналирование и доустановку/донастройку пакетов, во время работы с которыми были проблемы и тд. И я не вижу, что они тормозят.

Не знаю, как у вас на винде, но наверняка тоже есть средства для реализации всех нужных фич малой кровью.



popov654 31 октября 2018 в 11:17

0

> И что-то не заметно чтобы они тормозили комп

Просто для современного железа это «тьфу». А так-то по логике они и правда «тормозят» операции по сравнению с не журналируемыми фс.

> И я не вижу, что они тормозят

Так тут многое зависит от реализации. Microsoft при реализации движка обновления в Windows 10, имхо, здорово налажала. Хотя я могу быть не объективен, поскольку тестил только в виртуальной машине, а она уже сама по себе создаёт огромные тормоза, в реальности всё может быть на порядок быстрее.

BigBeaver 31 октября 2018 в 11:32

0

Это требует доказательств. Есть мнение, что ext4 быстрее, чем vfat.

sumanai 31 октября 2018 в 21:01

0

> У нас уже есть журналируемые файловые системы.

Обычные журналируемые ФС имеют журнал метаданных, но не данных. Плюс она гарантирует целостность на уровне файла, а не нескольких файлов. Именно для избежания ситуации, когда половина файлов обновлена, а вторая нет, MS пришлось делать поверх NTFS дополнительные транзакции TxF.

BigBeaver 31 октября 2018 в 23:40

0

А журнал данных и не нужен. У нас есть знание о том, что трансакция с таким-то файлом зафайлена и есть репозиторий со всеми эталонными для каждого пакета. У нас есть логи конфигурирующих скриптов и тд. В чем проблема-то?

sumanai 1 ноября 2018 в 20:26

+1

Так суть в том, что транзакция с одним файлом прошла успешно, а с другим не успела начаться. С точки зрения обычной журналируемой ФС всё в порядке, целостность структур не повреждена, данные (опять с её точки зрения) консистентные. А вот про то, что в новой DLL, которая обновилась первой, есть вызов функции, отсутствующей в старой версии другой DLL, которая обновится не успела, файловая система не в курсе. Именно по-этому и нужны транзакции уровнем выше, которые в винде затрагивают ещё и реестр.

BigBeaver 1 ноября 2018 в 20:39

0

Так за этим менеджер пакетов следит.



Ogra 1 ноября 2018 в 21:25

0

Вам напомнить, как обновление одного пакета удаляло сразу весь /usr?

BigBeaver 1 ноября 2018 в 21:38

0

Ну так мы знаем, кто что удалил, как это относится к трансакционной целостности? Система примет это за нормальное поведение — такое лечится только бэкапами.



Ogra 1 ноября 2018 в 22:31

0

Да никак не относится — нет никакой транзакции. Пакетный менеджер просто взял и запустил скрипт. Даже если скрипт упадет на полпути, /usr не восстановить. Откатить? Не, только бэкапы.

BigBeaver 1 ноября 2018 в 22:42

0

И?



popov654 2 ноября 2018 в 02:00

+1

Только зачем для этого было TxF городить? Можно просто вести лог каждого записанного файла (добавлять запись о его обновлении). При старте проверяем по логу, что обновилось абсолютно всё, если нет — продолжаем обновление с места остановки (плюс перед началом обновления делается снимок текущих версий файлов для отката назад, если что). Чем плох такой вариант?

sumanai 2 ноября 2018 в 18:23

0

Вы забыли про реестр и унификацию. TxF имеет открытый API, и им может пользоваться любая программа.



popov654 3 ноября 2018 в 02:39

0

Тогда в данном случае это просто как решение для сторонних программ, которым нужна целостность, чтобы авторы не писали свой велосипед (ну и да, с реестром они тоже работают, как вы верно заметили).

sumanai 3 ноября 2018 в 14:03

0

Не только сторонних, но и встроенных, то есть Windows Update тут самый что ни на есть рядовой пользователь этого API, просто самый первый.
Единственная проблема в том, что TxF не самая быстрая вещь.



Alexeyslav 29 октября 2018 в 10:57

0

А если какой-то файл помешает запустится самому мастеру? Как не так давно в моём телефоне на андроиде стал крешится сервис гугл прямо на старте который собственно следит за всеми остальными сервисами, их живучестью и настроен на автоматическую перезагрузку… и вот он упал, поднялся и снова упал… и так до бесконечности пока через 6 часов не разрядил батарею.
Кроме того восстанавливать какие-то удалённые файлы… а на что именно их восстанавливать? создавать пустыми просто чтобы были? Это такой очень сомнительный костыль… подобный мастер по восстановлению тогда должен держать в себе копии всех нужных файлов, причем в актуальных версиях! Что-то мне это напоминает…



popov654 31 октября 2018 в 09:44

0

> А если какой-то файл помешает запустится самому мастеру

Вероятность этого столь мала, что можно смело ею пренебречь. У вас то же самое случится, если повредится файл ядра, например.

> Кроме того восстанавливать какие-то удалённые файлы… а на что именно их восстанавливать? создавать пустыми просто чтобы были?

Так в том-то и дело, что не пустыми. Там выше в комментарии написали: нельзя, чтобы в этих файлах оставались недействительные записи, ведущие «в никуда». Вот это нужно и проверять.

Спойлер



Alexeyslav 31 октября 2018 в 10:20

0

Откуда тогда содержимое брать?



popov654 31 октября 2018 в 11:20

0

Либо делать «последнюю удачную копию», как с конфигурацией оборудования (снимок одной из ветвей реестра по сути), либо брать текущее содержимое, и удалять то, что не должно там быть (что может нарушить или уже нарушает работу программ). Первый подход и надёжнее, и быстрее, так что логичнее пойти по нему.

Но тут надо понимать, что в значительном числе случаев, если я правильно понял, проблемы с этими файлами будут приводить к сбоям программ. То есть это не фактор, мешающий запуститься самой ОС. В общем, тут не тот случай, что «файл был, и вдруг внутри него потерялась часть данных, надо брать предыдущую рабочую копию». Это случай, когда «файл есть, ничего не тёрлось, но внутри него в какой-то момент случилась логическая ошибка, ОС это запускаться не мешает». А полное сканирование системы с целью поправить этот файл может быть очень долгим процессом. Может, и не стоит делать такой сканер частью ОС, для этого уже есть софт от того же Norton в (типа WinDoctor в пакете System Works).

sumanai 31 октября 2018 в 21:02

0

> Может, и не стоит делать такой сканер частью ОС

Поздно, такой сканер есть со времён XP.



popov654 1 ноября 2018 в 15:34

0

А как он называется? Я не очень понимаю, про что сейчас речь)

XP (и даже 7/8/10) вряд ли запустится, если, например, серьёзно повредить реестр (простой пример — отредактировать реестр WinXP редактором реестра от Win7 и сохранить куст).

sumanai 1 ноября 2018 в 20:41

0

> А как он называется?

Старый как мир SFC /SCANNOW, или, в новых версиях, обслуживание через dism.exe



SirEdvin 24 сентября 2018 в 07:50

0

> Потому что для того, чтобы исправить хоть что-то нужно собрать 100500 бумажек и никто не будет менять архитектуру, даже если она безбожно транжирит ресурсы? Так об этом и статья.

Причем тут архитектура к бумажкам? Сложность банковского процесса в ПО — это отображение сложности банковского процесса в реальности, вот и все.

khim 24 сентября 2018 в 08:23

0

> Сложность банковского процесса в ПО — это отображение сложности банковского процесса в реальности, вот и все.

Дык «банковский процесс в реальности» — это как раз порождение сорта, подобного тому, что происходит в софте, только в этот раз не в операционной системе, а в оргпроцессах.



popov654 18 октября 2018 в 09:35

0

> в итоге для их обновления теперь пришлось использовать транзакции, а в этом случае 30 минутам на обновление удивляться не стоит

Но во времена Win7 всё обновлялось реально быстрее (правда, не очень корректно сравнивать мелкие патчи, которые и на десятку ставятся довольно быстро, и функциональное обновление сборки, которое приносит функций и внутренних изменений даже больше, чем в своё время приносили сервис-паки).



ladutsko 23 сентября 2018 в 14:22

0

ППКС



sboronin 23 сентября 2018 в 15:35

0

Согласен с автором, более того, процесс распухания софта я начал наблюдать еще в 90-х. Тогда еще мало кто соглашался с этим, находя оправдания в росте сложности ПО.
Сейчас это стало настолько очевидно, что есть шанс на то, что появятся те, кто будет готов что-то сделать в отношении данной проблемы.



sboronin 23 сентября 2018 в 15:38

0

Автор статьи пишет «Linux намеренно убивает случайные процессы.»
Можно ли привести примеры, когда Linux так делает.
Работаю с Linux-ом с 2000-го года, но не сталкивался с данным явлением.

khim 24 сентября 2018 в 05:16

0

> Можно ли привести примеры, когда Linux так делает.

OOM-Killer. Мерзкая вещь — но суть следствие неправильного дизайна. Где posix_spawn появился сильно позже сладкой/парочки fork/exec, часто реализован поверх них и, соответственно, требуется в подпорках, каковыми являются COW-fork и overcommit. А дальше — уже и OOM-killer, как следствие…



norguhtar 24 сентября 2018 в 09:16

0

Память никогда не заканчивалась? Это только в этом случае бывает.

Londoner 23 сентября 2018 в 16:27

+1

На самом деле, во фронтенде проблему исправить довольно легко. Достаточно создателям Хрома перестать загружать страницы длинной, скажем, больше мегабайта или джаваскрипт, требующий память больше десятка мегабайт. Сначала показывать предупреждение, а потом просто крешить эти страницы, оставив глубоко в настройках опцию «allow bloatware from the following domains»

Druu 23 сентября 2018 в 16:28

0

> Достаточно создателям Хрома перестать загружать страницы длинной, скажем, больше мегабайта или джаваскрипт, требующий память больше десятка мегабайт.

А зачем им это делать, если пользователи хотят такие страницы загружать?

Londoner 23 сентября 2018 в 17:00

0

Вряд ли они захотят их загружать будь у них альтернатива. А альтернатива не появится по чисто экономическим причинам, пока гавнокод остаётся безнаказанным.

PsyHaSTe 23 сентября 2018 в 17:02

+1

Вспоминается, как люди хакали свои драйвера, чтобы не проходить сертификацию майкрософта — писали скрипт, который за пределами экрана открывал нужное окошко «разрешить установку небезопасного драйвера» и программно жали «ок».

Druu 24 сентября 2018 в 04:03

+3

> А альтернатива не появится по чисто экономическим причинам, пока гавнокод остаётся безнаказанным.

А безнаказанным он остается потому что, еще раз, пользователей все устраивает.
Нету у пользователя такой проблемы. Она есть у разработчиков, у вас, у автора статьи там. А у пользователей — нет.

Londoner 24 сентября 2018 в 22:25

+1

Т.е. Вы всерьёз утверждаете, что у пользователей есть реальная альтернатива, где странички в сотню килобайт открываются за считанные миллисекунды, а пользователи по-прежнему продолжают выбирать тормозной и глючный кактус необъятных размеров? Тогда, с этого момента, пожалуйста, поподробнее.

0xd34df00d 25 сентября 2018 в 01:44

+1

lynx/links/elinks/w3m

Londoner 25 сентября 2018 в 12:28

0

Вы серьёзно? А facebook-ом или youtube-ом можно полноценно пользоваться хоть в одном из них? А какой процент веб-разработчиков тестирует на них свои продукты?

0xd34df00d 25 сентября 2018 в 20:32

+1

Описанному вами в прошлом комментарии критерию они удовлетворяют, тем не менее.

А ютуб можно через youtube-dl пополам с mpv смотреть.

Londoner 25 сентября 2018 в 22:32

0

Вы мне напомнили двух юристов с бутербродами из анекдота. Тогда, следуя Вашей логике, заказ авиабилетов — через турагентство по телефону, а фейсбук — это бабушки у подъезда, да?

Druu 26 сентября 2018 в 14:05

0

> Т.е. Вы всерьёз утверждаете, что у пользователей есть реальная альтернатива, где странички в сотню килобайт открываются за считанные миллисекунды, а пользователи по-прежнему продолжают выбирать тормозной и глючный кактус необъятных размеров?

А это не важно, есть или нет. Важно, что то, что есть, пользователей устраивает.
Если вы считаете что страничка, открывающаяся за миллисекунды — это серьезное конкурентное преимущество, ну делайте свой бизнес, под это заточенный.
Вот дяди с деньгами считают иначе, не платят программистам за оптимизацию, и, вобщем-то, конец истории.

Londoner 27 сентября 2018 в 01:31

0

Если Вы считаете что проблемы не существует, как Вы объясните такое количество комментариев к этой статье? Как Вы думаете, почему куча народу пересела с PC на Mac? Подсказка — из-за чуть большего (не намного, кстати) качества железа и софта. А дядям с деньгами надо чтобы Вы каждый год покупали у них новый девайс, вот их — да — всё устраивает. Следуя Вашей логике, если Вам не нравится обслуживание в сбере, никто не мешает Вам открыть свой банк. А ещё почитайте вот тут.

popov654 18 октября 2018 в 09:43

0

Меня не устраивает. Как пользователя в том числе. Не говорите за всех

Druu 18 октября 2018 в 11:41

−1

Я и не говорю за всех, я говорю про подавляющее большинство, в рамках которого вы — где-то за пределами стат. погрешности.

popov654 18 октября 2018 в 13:06

0

Судя по количеству комментариев в данной ветке (а также зная своих друзей не-айтишников, и то, чтобы они любят, чтобы всё «работало быстро») — вы не правы.

Druu 18 октября 2018 в 15:13

0

> Судя по количеству комментариев в данной ветке (а также зная своих друзей не-айтишников, и то, чтобы они любят, чтобы всё «работало быстро») — вы не правы.

Даже если все комментаторы данной статьи вас поддерживают — это все еще не выйдет за пределы стат. погрешности.
А то, что я прав, подтверждается ситуацией на рынке. Если бы пользователям действительно была бы нужна оптимизация — то оптимизированные решения бы побеждали в конкурентной борьбе. Но они не побеждают.
То есть, факт, наличия данной статьи и ее обсуждения — как раз говорить в пользу моей правоты. Если бы правы были вы — никакой бы статьи в принципе *не было*, т.к. софт в своем большинстве был бы быстрым и оптимизированным.

popov654 23 октября 2018 в 03:26

0

> Если бы пользователям действительно была бы нужна оптимизация

Но энному числу людей, которых я знаю, она бы не помешала (их бесят тормоза). И это люди не сидят на Хабре, они даже не программисты. Просто не любят ждать)

Вы правда не видите разницы между «не хотят ничего менять» и «жрут кактус, потому что не видят возможности ничего изменить»?

Druu 24 октября 2018 в 09:31

0

> Но энному числу людей, которых я знаю, она бы не помешала (их бесят тормоза).

Так возвращаемся выше — этих людей слишком *мало*, чтобы скинуться и оплатить быстрый софт.

BigBeaver 24 октября 2018 в 13:05

0

Это требует отдельного доказательства.

popov654 25 октября 2018 в 01:44

0

Окей, допустим вы правы. Но если бы даже было не так мало — как и куда предлагаете скидываться?)

Druu 25 октября 2018 в 01:53

0

> Но если бы даже было не так мало — как и куда предлагаете скидываться?)

Ну например на ежемесячную подписку для "быстрого браузера без

рекламы и смс".

popov654 25 октября 2018 в 02:02

0

Так надо найти ещё людей, которые возьмутся его делать. Есть кикстартер, но там это в обратную сторону работает.

Druu 25 октября 2018 в 13:45

0

> Так надо найти ещё людей, которые возьмутся его делать.

Если будете платить деньги — люди найдутся :)
Задача-то сама по себе хорошая, правильная.

InterceptorTSK 23 сентября 2018 в 16:33

0

Когда я начал прозревать, ну это давно уже было — мне попался забавный клиент. Он компьютер вообще в глаза не видел. Оно ему не нужно было полностью. Впрочем 99% и без компьютера прожить могут прекрасно, но суть не в этом. Значит это был директор достаточно крутой канторы, но вообще без компьютера. Так вот, обсуждали мы что делать будем, а рядом сидел главный инженер этой канторы. Т.е. директор сидел рядом исключительно справочно, собственно он этого и не скрывал, и заранее так и сказал.
Обсуждение свелось к банальностям. Нахрен все это надо? Если этот ваш кампутер нихрена не может? На том и разошлись. А ведь етот директор абсолютно прав.
И чем дальше оно двигается в каких-то непонятных направлениях — тем хуже оно становится.

И да, бухгалтер у него считал на счетах. Давно дело было. Кантора приличная, человек 200.
Рассчитывала все одна женщина. Ошибок ровно ноль. Я специально спросил. НОЛЬ. Проблем тоже НОЛЬ.

kinazarov 25 сентября 2018 в 16:41

+1

Наверное у вас идеальные компетентные никогда не болеющие сферические люди в вакууме в «етой канторе»
Ознакомьтесь, грамотяй:
dtkt.com.ua/show/3cid11625.html

InterceptorTSK 25 сентября 2018 в 17:02

−2

> Это к чему?) Ну увидели вы ошибку, и что?))) А если это не ошибка? Если оно преднамеренно так пишется?
> И да, если вы пытаетесь неумело ловить ошибки, которые вовсе ими не являются, так ловите их все)
> Или опять слабо?
> Какой-нить русовед закроет лицо рукой тут. Причем вовсе не над моим текстом. Ага?
>
> После слова «наверное», вам необходимо запятую поставить, если ЧЁ) А в конце предложения ставится точка, а у вас нету точьки. Но я бы этого никогда указывать не стал бы. Это уже исключительно «пАржать». Вечер же. чО бы нет?)

kinazarov 25 сентября 2018 в 17:28

+2

>А если это не ошибка?
А что же это? Авторский «высокей штиль»?
>так ловите их все
Вы мне пояказывайте что и как мне делать. А еще лучше смотрите рисунок №1.

Вы неправы и по форме и по сути. Ошибкой является писать слово «контора» через «а».
Также ошибкой является держать единственного и незаменимого бухгалтера со счётами и верить что она допускает ноль ошибок, ведь нет коллег, которые могут проверить работу этого бухгалтера.
И хамство как правило — тоже ошибка. Да, даже вечером. Да, и в интернете тоже.

InterceptorTSK 25 сентября 2018 в 18:33

−2

> Давайте еще разок. Прежде чем мое бревно искать, которое вовсе не бревно — вы свое вытащите. Ну бредово же тыкать в чужие ошибки тут же допуская собственные ошибки)
> Смысл про бухгалтера был же изначально, но вы съехали на ошибки в словаре, это от безнадежности совсем?)
> Причем тут вера? Я вообще не знаю что это и никакой надобности знать что это — мне нет.
>
> И причем тут болезнь?)) Бухгалтер болеет — процессы стоят. Хоть на чем бухгалтер сидит — хоть на счетах, хоть на кампутере. Если нет бухгалтера, т.е. он болеет — вам кампутер поможет что ли? А если кампутер помогает без бухгалтера, тогда зачем бухгалтер?
> Вы объяснитесь хоть как-то)))) Действительно смешно)

kinazarov 2 октября 2018 в 11:00

0

>И причем тут болезнь
Притом, что если бухгалтер один единственный на предприятие из 200+ людей, то процессы будут из-за его болезни стоять. А если бухгалтеров больше одного — не будут. Они скорее всего будут протекать со сниженной скоростью или эффективностью. А может и не будут, зависит от числа бухгалтеров и организации процесса.
Это слишком сложно для вашего восприятия?
Или для вас является нормой когда 199 человек ждут пока перестанет болеть один чтобы, например, получить зарплату?

Для меня очевидно что так быть не должно и на 200 сотрудников должно быть больше одного бухгалтера. Дополнительно: если бухгалтеров больше одного, они могут проверить работу друг друга и предотвратить ошибку в регламентной отчётности. И они, знаете ли, так делают. Просто потому что такая ошибка чревата крупными штрафами и потерей работы для всех этих бухгалтеров.

А вот когда бухгалтер один и проверить его работу квалифицированно никто не может, то остаётся только верить что ошибок этот единственный бухгалтер со счётами не допустит. Как верят в бога, существование которого невозможно проверить.

Так что ошибки не только в словаре — ошибка в вашем представлении о том как правильно организовать процессы.
См рис. №1

DMGarikk 2 октября 2018 в 12:17

0

> бухгалтер у него считал на счетах. Давно дело было. Кантора приличная, человек 200.

Сейчас (да и лет 10 назад уже), физически невозможно посчитать зарплату в одиночку на счетах в фирме со штатом 200+ел
Все остальное... с натяжкой, с исключениями для налоговых режимов... да, но зарплату нереально просто по физическим причинам... достаточно выписал паре человек больничные, а потом еще их и продлить когда человек принесет дополнительные бумажки
пересчет средних заработков пару суток занять может

Я в далеком уже 2007 году видел похожую фирму где был главбух не признающий компьютеры, так там был штат человек 20 бухгалтеров которые на компьютерах печатали огромные моногострачные обороткы и ведомости и шахматки чтобы главбух на счетах все мог пересчитать и проверить, бумагу контора тоннами изводила на это

DrPass 2 октября 2018 в 14:34

0

Вот то же самое хотел сказать, без компьютеров в компании на 200 человек нужно будет кучу участков по бухгалтерии — на склад, на ОСы, на закупки/продажи, на кассу и банк, минимум два человека на зарплаты, и главбух.

tvr 2 октября 2018 в 15:18

0

> нужно будет кучу участков по бухгалтерии — на склад, на ОСы, на закупки/продажи, на кассу и банк, минимум два человека на зарплаты, и главбух.

Класс, куча народа и все при деле.

popov654 18 октября 2018 в 09:47

0

Там такая сложная формула расчёта больничных что ли?)

Alexeyslav 29 октября 2018 в 11:18

0

Если бы была какая-то фиксированная формула, одна и на всех то небыло бы проблем от слова вообще — один формулу вывел остальные пользуются и в шоколаде. Но нет. Каждый интерпретирует законодательство по своему, у каждого своя методика расчета... на закон ровняются, но нигде полной ясности и однозначности нет как и во многих законах.
Не жалуются — значит всех всё устраивает (с).
Помню, у нас был случай работник посчитал свою зарплату и она не сошлась на 1 копейку(недоплатили) — суд выиграл, компенсация...
Полагаю, если бы каждый проверил расчет своей зарплаты, таких несоответствий(в особенности из-за неоднозначности интерпретации законов) было бы очень много. Народ считает за благо что больничные вообще насчитывают и платят...

DrPass 29 октября 2018 в 16:00

0

Формула там как раз вполне понятная и без двояких трактовок, но вычислений слишком много, чтобы было возможно их проводить вручную. Например, для начисления отпускных бухгалтер должен по сотруднику посчитать средний дневной заработок за предшествующий год. Т.е. ему нужно поднять все начисления, сделанные работнику за год, просуммировать, и разделить на количество фактически отработанных дней, согласно табелю выходов. Который тоже нужно скрупулёзно перебрать за год. Т.к. где-то отгул, где-то больничный, где-то другой отпуск.
Это только по одному сотруднику. Если их двести, да ходят в отпуска по два раза в год, то такую процедуру надо проделывать каждый месяц раз сорок. Как вы думаете, это реально без компьютера?

popov654 31 октября 2018 в 09:47

0

> и разделить на количество фактически отработанных дней, согласно табелю выходов.
> Который тоже нужно скрупулёзно перебрать за год

> Как вы думаете, это реально без компьютера?

Ещё как реально, простейшая арифметика. Калькулятор вполне сгодится.
Только вот табель выходов должен быть в наличии, и храниться под рукой и в удобном формате.

Кстати, ещё это всё можно неплохо в Excel обсчитывать через формулы, имхо.

DrPass 31 октября 2018 в 14:41

0

> Ещё как реально, простейшая арифметика

Я под «реально» имел в виду исходные данные, т.е. «в одиночку и на работе», а не «в принципе, если время неограничено». Вам же не просто колонку цифр просуммировать. Вам надо сначала собрать папки с журнал-ордерами по начислениям сотрудникам за предыдущий год, из каждой просмотреть журналы, выбрать строки с нужными начислениями по вашему сотруднику, просуммировать.
То же самое проделать с табелем выходов (там, правда, всего одна папка, если 200 человек). Это не хранится под рукой и в удобном формате. Для компании на 200 человек это как минимум полка в шкафу, которую всю надо пролистать.
Нужен хотя бы Эксель, в который в таблицы по каждому сотруднику заносить предыдущие начисления и выходы. Но это уже как раз будет расчет на компьютере.

Ogra 1 ноября 2018 в 09:55

0

> Это не хранится под рукой и в удобном формате.

А я бы хранил, если бы мне надо было 40 раз в месяц (это 2 раза в день) считать.

DrPass 1 ноября 2018 в 14:10

0

Так объем исходных данных на компанию из 200 человек, повторюсь, без компьютера занимает как минимум одну полку шкафа. Не влезет оно вам под руку :)

popov654 1 ноября 2018 в 15:40

0

Я писал больше про подсчёт. Хранить это всё можно хоть в MySQL, если очень хочется. А потом выбирать по запросам)

DrPass 1 ноября 2018 в 17:00

0

Да подсчет нет смысла рассматривать в отдельности от остального процесса. Что вам даст тот факт, что сложить сотню чисел даже на счетах можно за десять минут, если перед этим нужно потратить несколько часов на то, чтобы их найти в бумагах, если нет компьютера?

popov654 2 ноября 2018 в 02:10

0

И то правда)

BigBeaver 31 октября 2018 в 11:00

0

Раньше же считали как-то.

DMGarikk 31 октября 2018 в 11:12

0

раньше были отделы расчетчиков, человек на 20, которые только расчетами зарплаты занимались, причем в зависимости от размера конторы были отельные отделы расчетчиков для разных подразделений, а также огромные журналы и регистры для записей.
И это при том что раньше методики расчетов были гораздо проще и очень много аналитики округлялось и отбрасывалось из-за сложности

BigBeaver 31 октября 2018 в 11:16

0

Вы спросили, реально ли. Очевидно, реально. Насколько затратно — другой вопрос.

popov654 31 октября 2018 в 11:28

0

> раньше были отделы расчетчиков, человек на 20, которые только расчетами зарплаты занимались

Это было уже после внедрения компьютеров? Я так понимаю, если использовать компьютер даже просто для хранения, это уже даёт бонус в виде более быстрого обмена документацией между сотрудниками такого отдела и почти мгновенного поиска данных за нужные периоды.

А считать можно хоть в программе «Калькулятор», хоть в Excel, что на физическом калькуляторе, если формула проста.

Выше просто пытались представить ситуацию так, как будто там какая-то дикая многоступенчатая формула в 20 этажей, которую без специального софта не посчитать. Теперь, когда я увидел формулу — она не кажется мне настолько нереально сложной (хотя наверное, просто пример с расчётом зарплаты был не из самых сложных, есть хуже).

DMGarikk 31 октября 2018 в 11:56

0

Это было до внедрения компьютеров, в РФ это вполне встречалось до середины 2000х годов
==
формула там простая, там способ расчета сложный
например вы посчитали зарплату и отпускные, а сотрудник взял да принес вам еще один больничный который «забыл»… а потом еще справку какуюнить дающую право на вычеты/пересчеты… а потом этих сотрудников много… а тут раз ктото ошибку нашел опять все пересчитывать. и это нарастает как снежный ком с увеличением штата
==
да и формула… там есть такие понятия как Календарные даты, Рабочие даты… причем бывает что некоторые дни в одном случае считаются рабочими, а другие нет… а потом еще бывает ночные смены которые например первой половиной выпадают на обычный день, второй на праздничный… а еще всякие фокусы типа что какието часы в рабочем дне оплачиваются на 1 ставки, некоторые по 1,5 ставки, некоторые по 2 ставки… и это в течении 2-3 дней в месяце… вобщем нюансов очень много и одной формулой в экселе это очень трудоемко посчитать

popov654 31 октября 2018 в 13:25

0

Спасибо, теперь относительно понятно, в чём проблема.

DMGarikk 29 октября 2018 в 19:35

0

что реально из-за копейки?
там на самом деле с копейками интересная штука, причем даже в НДФЛ есть, который считается нарастающим итогом в рамках года… и в законе не написано как округлять эти суммы… каждый месяц (когда платится зарплата или налог) или раз в год когда подается декларация?
при этом налоговая может иметь совершенно отличное от вас (как бухгалтера) так и от самого работника, мнение как именно это надо делать… и разница может нескольких десятков рублей достигать в рамках года
тоесть если каждый месяц округлять в меньшую сторону (в пользу работника), а потом в конце года посчитать все скопом (для декларации)… то получатся веселые цифры которые с работника удержат потому что общими цифрами недоимка получится… помню долго в одной конторе собачились с налоговой по этому вопросу… все по закону же… но работник обижен

popov654 31 октября 2018 в 09:49

0

Имхо, несколько десятков рублей — тоже копейки, чтобы из-за этого судиться или обижаться. Тем более если работник понимает, что это технические нюансы, а не жмотство (можно же объяснить).

Alexeyslav 31 октября 2018 в 10:22

0

Реально. Там вопрос был принципиальный, человек тоже принципиальный и шла война с бухгалтерией.

Wesha 2 октября 2018 в 21:33

0

> Кантора приличная

И как кантор поживает?

lxsimkv 23 сентября 2018 в 18:30

+1

Одной из причин разрастания (а может и основной) является, по моему наблюдению, поддержка обратной совместимость и нескольких платформ.



Infra_HDC 23 сентября 2018 в 21:10

0

> Моё разочарование в софте

Вы были очарованы софтом? И если да, то можно поподробней об этом?

springimport 24 сентября 2018 в 18:24

0

Да ладно, никто не разговаривает на чистом языке, так еще и смысл слов постоянно меняется.



norguhtar 24 сентября 2018 в 09:25

0

Очень долгое время компьютеры каждый год становились все быстрее и быстрее. Даже тот же Спозльски писал. Вы можете заняться оптимизацией своего ПО несколько месяцев. А можете просто играть в рок-группе. В результате ПО будет работать одинаково.

Ну и да. Не так давно java была прям эталоном работает медленно, жрет много памяти. А что же сейчас? Да на фоне node.js она жрет мало памяти и ультрабыстрая.



SirEdvin 24 сентября 2018 в 09:37

0

> Ну и да. Не так давно java была прям эталоном работает медленно, жрет много памяти. А что же сейчас? Да на фоне node.js она жрет мало памяти и ультрабыстрая.

Не так давно — это лет 10-15 назад? Сейчас она жрет много памяти только в мемчиках из интернетов, потому что люди не знают, как она работает.



norguhtar 24 сентября 2018 в 10:35

0

Ну да Ж)

VolCh 24 сентября 2018 в 19:37

0

А знания как-то влияют на показатели счётчиков ОС?



SirEdvin 24 сентября 2018 в 19:51

0

Приходит умение менять значения памяти, которая отъедается. Даже еластик можно зажать в 256 мб и оно даже как-то будет работать.

stanislavskijvlad 24 сентября 2018 в 09:56

0

Автор в статье указал, что Windows 95 была функциональна. Товарищ предложил поставить W98, и от этого нам захочется вернуться в будущее. Вы поймите, 98 была крута и функциональна для Того времени. Автор хочет, чтобы при сегодняшних возможностях скорость работы была пропорциональна, и современные системы были на уровне Своего времени.

DMGarikk 24 сентября 2018 в 10:19

0

Если бы Win9x были стабильны также как NT4, то не особо и хотелось бы возвращаться в будущее, разве что w2k был лучше… но начиная с XP винда погрязла в рюшечках окончательно и бесповоротно тоесть в ТО время были системы сравнимые по стабильности с сегодняшними но по разным причинам все юзали этот 9х кошмар

Roman_Popov 24 сентября 2018 в 10:40

0

Выше приводили в пример авто и самолеты как образец правильного подхода. Так вот, у меня плохие новости:
denokan.livejournal.com/178370.html — самолеты
habr.com/company/pvs-studio/blog/310862 — атомобили
В общем и там Agile рулит и педалит, часто в прямом смысле.

Купил билет = записался в бетатестеры боинга? :)



Matohin 24 сентября 2018 в 11:55

0

Прочитал до середины и понял, что мысль уже выражена и автор ходит по кругу, поэтому остаток просмотрел по диагонали. В заключение инженерам предлагают решить все проблемы но… призыв не по адресу.
То каким образом организована софтверная индустрия и решение каких проблем востребовано — не технический вопрос. Его решают пользователи и крупные институциональное игроки за пределами индустрии. Если они не видят проблем с «раздутостью» приложений, то их нет.
Корректный способ считать «производительность» — это сравнить объем IT как индустрии с средней прибавочной стоимостью, создаваемую средним приложением в час, в долларах по ППС.
Хорошим прокси для первого показателя является доля IT в мировом ВВП. Второй показатель посчитать крайне сложно, поскольку отделить ту прибавку в производительности труда, которую дает именно IT составляющая можно только по косвенным признакам. Да и посчитать запущенные приложения. Но про это можно было бы подумать и написать статью. Я бы почитал. Только писать ее должен экономист, а не программист.



popov654 18 октября 2018 в 09:50

0

> Его решают пользователи и крупные институциональное игроки за пределами индустрии. Если они не видят проблем с «раздутостью» приложений, то их нет.

Так пользователи-то видят. Это менеджерам компаний пофиг — а программистам лень решать этот вопрос бесплатно (а кое-где и запрещают сверху, потому что задачи себе программист ставит не сам, а решения спускаются через менеджера проекта и тимлида группы). То есть программисту не позволят заниматься задачей, отличной от той, что была поставлена.

Druu 18 октября 2018 в 11:44

0

> То есть программисту не позволят заниматься задачей, отличной от той, что была поставлена.

А с чего бы должны позволять? Ему же деньги платят за то, чтобы он делал что-то, что повысит конкурентные свойства продукта, то есть, в итоге, то, что принесет больше денег, чем вы программисту заплатили, иначе, с точки зрения заказчика, смысла в работе программиста нет. Раз оптимизация не повышает конкурентные свойства продукта, то есть не приносит больше денег, чем за нее уплочено — ну так никто ее и не заказывает.

Если вы, как пользователь, хотите оптимизации — ну так возьмите и профинансируйте разработку быстрого браузера или быстрые аналоги каких-то других медленных, по-вашему, приложений. Или соберитесь и краудфандинговую кампанию организуйте. Почему вы ждете, что кто-то за вас это сделает? Это же ВАМ нужно. Вы и делайте.

popov654 18 октября 2018 в 13:07

0

> Раз оптимизация не повышает конкурентные свойства продукта

Вот в этом и есть фундаментальная ошибка. Ещё как повышает (особенно при прочих равных).

Druu 18 октября 2018 в 15:14

0

> Ещё как повышает (особенно при прочих равных).

А прочих равных не бывает, т.к. количество ресурсов ограничено. Если вы их потратили на оптимизацию — значит, не потратили на что-то другое. Так уж выходит, что, оказывается, "что-то другое" для пользователей важнее. Как минимум — ваш конкурент может вывести свой продут на рынок раньше вас, т.к. не будет на оптимизацию тратить время. И пользователи в итоге голосуют за то, чтобы иметь пусть и неоптимальное решение, но сегодня, чем оптимальное, но когда-то потом. Потому что задачи этим пользователям надо решать сегодня, а не потом.

popov654 23 октября 2018 в 03:53

0

Если он выведет продукт раньше, но он будет глючить и тормозить, в итоге он потеряет большую часть людей, когда выйдет лучшая альтернатива (особенно если она будет стоить дешевле). Посмотрите на PUBG, который проиграл Fortnite (из-за тормозов, высокой цены, менее продвинутой модели монетизации и более позднего внедрения кроссплатформы).

Ну или давайте дойдём до абсурда — и посмотрим на Firefox, который вышел на рынок раньше Chrome, но чем всё кончилось в итоге))

DMGarikk 23 октября 2018 в 09:00

+1

> который вышел на рынок раньше Chrome, но чем всё кончилось в итоге

кончилось тем что на каждом сайте гугла вам насильно заставляют поставить хром… у кого рекламный бюджет и охват больше, тот и круче

Druu 24 октября 2018 в 09:34

+1

> Если он выведет продукт раньше, но он будет глючить и тормозить, в итоге он потеряет большую часть людей, когда выйдет лучшая альтернатива (особенно если она будет стоить дешевле). Посмотрите на PUBG, который проиграл Fortnite (из-за тормозов, высокой цены, менее продвинутой модели монетизации и более позднего внедрения кроссплатформы).

Вы в своем примере назвали четыре причины, из которых три — с оптимизацией никак не связаны :)

popov654 25 октября 2018 в 01:47

0

Верно. Просто по ютьюбу ходили жуткие истории про то, как лагает PUBG, Репутация — она такая. Уже и лагать перестанет, а слухи о лагах всё ползут с диким лагом (сорри за тавтологию)… Вспомнить вот историю с Вистой. Висту починили, а люди на неё переходить всё ещё боялись :)

В целом — да, понимаю, что значительная часть всё равно может остаться, мой аргумент очень шаткий. Просто я бы на месте автора стартапа не спешил выходить первым, и предпочёл качество. А уж чем бы оно для меня при этом закончилось — не знаю…

BigBeaver 18 октября 2018 в 15:39

0

Проблема в том, что конкуренция, как минимум, трехсторонняя. Фреймворки конкурируют за программистов, а программисты за пользователей. Браузеры конкурировали за поддержку сайтов, а сайты за пользователей. То есть, пользователю пофиг, как быстро работает браузер О, если в ней не работает что-то нужное. А не работает оно потому, что браурер Х реализовал фичу У, которая позволяет на пару месяцев быстрее запустить это «нужное», тем самым выдавив конкурентов, которые бы нормально работали в браузере О. Ну и тд

popov654 23 октября 2018 в 03:30

0

Жаль только, что это «нужное» часто является нужным с большой натяжкой…)) А так абсолютно согласен.

Кстати, я буду прав, если скажу, что данный принцип имеет исключения?

Возьмём ситуацию: Firefox реализует некую фичу, которая позволит запустить что-то крутое компании Х. Х выкатывает версию, которая работает только в Firefox (или работает везде, но плохо, а в Firefox хорошо). Сильно ли это поможет популярности Firefox? Много ли людей придут в первые месяцы в такой проект, учитывая повсеместное засилье Chrome?

BigBeaver 23 октября 2018 в 09:32

0

Раньше помогало. Многие ставили ФФ ради плагинов (включая адблок). Многие ставили оперу ради умного управления контентом (загрзка медиа по запросу вместо мгновенного, сжатие трафика, банерорезка). Год назад я поставил ФФ, когда очередная версия nordic-drv-zone перестала работать в опере. При этом опера не так визуально ужасна как ФФ и не так роняет систему как хром.

Другое дело, что против хромиума (как движка) уже нет смысла бороться. Нет шансов. Хотя мне до сих пор не хватает фич 8 оперы (дожили до того, что скучаем по функциям 10-летней давности, которых нет в типа прогрессивном современном софте, зато, тормоза есть).

popov654 23 октября 2018 в 17:33

0

Вот с движком не согласен. Чем плох Gecko? Чем не годятся его форки типа Goanna? Про движок Edge не будем, у него, наверное, и правда всё не так гладко по скорости и корректности отрисовки :)

> При этом опера не так визуально ужасна как ФФ

ФФ ужасен, серьёзно? Вы про старый Australis интерфейс, или про новый? Если про новый — проблема полностью решается темами и кастомными стилями :)

Я себе очень красиво всё настроил, прямо не нарадуюсь.

sumanai 23 октября 2018 в 19:46

0

> Если про новый — проблема полностью решается темами и кастомными стилями :)

А в старом я себе накликал прекрасный интерфейс в Classic Theme Restorer, а в новом нашёл только пару негибких решений, которые не позволяют сделать как мне нужно.

popov654 25 октября 2018 в 01:51

0

Вам сложнее: вы знаете, как вам нужно. Мне вот всё равно, лишь бы было симпатично, не слишком ярко и глаза не мозолило (ну и не тянуло тошнить при взгляде наверх от безвкусицы).

Сначала мне очень нравилась Opera 11.64. Потом пересел в 2015-ом на Хром, очень не нравилось, что он весь такой светлый, почти белый, и полоска окна синяя. Поставил тему «Тёмный графит», привык за пару недель, теперь с ней не расстаюсь.

На Win7 не нравился шрифт в Хроме (и тормоза при старте), но нравился Firefox по скорости и в плане рендера текста. Но бесил новый контрастный скин. В итоге скачал тёмно-бирюзовую тему из стора и сам её довёл до ума, перекрасив заодно все меню, чуть поменяв цвет иконок на кнопках, и ещё по мелочам.

Alexeyslav 29 октября 2018 в 11:27

0

В новом ФФ кастомизация очень сильно ограничена. Нельзя сделать интерфейс как в старом браузере… это уже принципиально невозможно.

popov654 31 октября 2018 в 09:51

0

Как в старом — нельзя. Но зато можно очень гибко управлять цветом, шрифтом, тенями, менять иконки на свои (включая их через data-uri)… Там возможности по сути ограничены возможностями самого CSS (и ещё немного сверху движком UI). Можно скрывать лишние кнопки. Можно кое-что даже позиционировать в другое место, были какие-то статьи на эту тему… Не так гибко как раньше, но можно.

ComradePashka 24 сентября 2018 в 12:48

+2

Со многими тезисами и фактами сложно не согласиться, но все же местами, как мне кажется, материал достаточно бессистемно замешан в здоровый винегрет, в котором теплое сравнивают с мягким.
Если отталкиваться от той же аналогии с автомобилями, например, можно из урбанистического утиля при наличии молотка, зубила и достаточного количества задора собрать какой-нибудь «мандыбас» с паровым двигателем, который даже будет способен передвигаться и этим соответствовать ярлыку «транспортное средство», но никому ведь по-серьезному не придет в голову сравнивать такую поделку с флагманами автопрома, у которых, не смотря на отсутствие принципиально революционных идей в устройстве ДВС, процессы проектирования, производства и текущего обслуживания все-таки претерпели достаточно существенных изменений. Точно так же нелепо сравнивать легковой автомобиль, например, со спецтехникой или болидами формулы-1. И тем более скрещивать из них гибридов.
Для энтерпрайз разработки (запущенных случаев) в которой не то что один разработчик, а не одна команда, отдел или даже не один департамент, в купе с руководителями отделов, манагерами разного калибра, маркетологами и еще огромной горой народа, которые в принципе далеки от программирования, и при этом еще ТЗ там из разряда уже классических «семи перпендикулярных линий», конкретный отдельно взятый разраб вообще никак не влияет на качество продукта на выходе. И это — следствие современной цивилизации потребления: код больше на миллион строк? на 100 мегабайт толще дистрибутив? прекратили поддержку пользователей с ОС SuperOS-1999? — люди, которые занимаются продажей продукта явно такими вопросами не интересуются, по-крайней мере до тех пор пока корреляция этих факторов с ЦА на рынках сбыта не начнет превышать допустимые значения.
Если говорить об одиноких самураях кодинга, опять же всплывает классический микроскоп с орехами. Навешивание фуллстека на какой-нибудь одностраничный сайт-визитку ничем не лучше тотального отгораживания от фреймворков, постулирования, например, ванильного яваскрипта в чуть более крупных проектах, неумение и нежелание осваивать новые инструменты и вообще непонимание того для каких целей они создавались. Не удивительно что все это законно выливается в костыли, велосипеды и в итоге в спагетти-код.
Изучение и использование новых фреймворков помимо всего прочего может открывать не только новые конструкции языка, но и целые новые парадигмы, что в свою очередь с накоплением опыта прививает программисту еще и навыки правильного проектирования архитектуры. Т.е. как раз-таки «сон разума порождает чудовищ» и какой-нибудь закоренелый jQuery-ист придя, скажем, в react с пылу с жару начнет там сшивать по кускам и оживлять монстров Франкенштейна. Только это же не говорит, что фреймворки — зло.
С другой стороны обратная совместимость никого не насилует и не обременяет, например хотя бы утилиты из Sysinternals по сей день мастхэв и прекрасно работают под десяткой; для программирования на грани искусства есть демо-сцена; какие-нибудь Ардуины просто физически не позволят расточительно относиться к ресурсам и полностью наплевать на оптимизацию; а опенсорц при основательном погружении просто поражает воображение — на сколько эффективным может быть повторное использование кода и коллаборативная работа вообще.
В общем проблемы конечно есть, но автор сильно сгущает краски и бьет тревогу там где можно видеть нормальный эволюционный процесс во всем его многообразии мутаций.

Alexeyslav 29 октября 2018 в 11:37

0

Проблема ведь не в фреймворках как таковых — они и правда экономят время на разработке велосипедов, а в том что фреймворки стремяться стать слишком универсальными — один фреймворк на все случаи жизни и под все платформы. Ну дескать чтобы программистам не плодить лишние сущности. Где та тонкая грань между разумным разделением функциональности и универсальности? Что если делать не универсальный фреймворк, а линейку фреймворков по градации тяжести но с однообразным интерфейсом, и если не полностью совместимым то по крайней мере легко адаптируемым в случае необходимости перехода. А кому это надо… нам за это не платят! (с)

staticlab 29 октября 2018 в 12:26

0

Так приходят другие разработчики, и говорят: вот у нас один фреймворк, и всё хорошо. А у вас тут зоопарк, и их учить сложно. Давайте сделаем один уберфреймворк для всего.

ComradePashka 29 октября 2018 в 14:01

0

Ну вот в качестве примера, буквально месяца 3 назад, после первого знакомства с Laravel я сразу же начал делать проект и он уже больше месяца в работе — правда это вообще не веб-приложение, там ларовские таски и работа с почтой, причем частично используется SwiftMailer который часть Symfony, а для работы с imap вообще был взят сторонний пакет с packagist. И мне не пришлось гробить годы на изучение, потому что я использовал только то, что было нужно под задачу.
На другом ларовском проекте у меня в качестве пресета фронтенда выбран реакт и я его собираю/тестирую вебпаком, который суть приложение для ноды.
И вроде бы все нормальные фреймворки работают аналогично — есть ядро (как правило составное), есть куча расширений, есть возможность пилить полностью свое, и компоновать все это между собой — и в этом универсальность и заключается, а не в неком всепокрывающем API и безальтернативном призыве «делай как я!» (возможно, таким для меня в свое время стал ExtJS и может еще какие-то самобытные поделки, которые я благополучно бросил и забыл)
То что неудобно — то мы не используем, то и не приживается. Я бы сказал, что парадигмы программирования и отдельные концепции, подходы и, далее, по цепочке, фреймворки, инструменты и в целом код эволюционируют. И, логично предположить, что выживают сильнейшие и лучшие. А программисты в таком случае просто ноутели мемов, а те кто противится неизбежному — это уже «окаменелости древних организмов» :-)

popov654 31 октября 2018 в 09:58

0

Мне кажется, любой фреймворк будет в чём-то неудобен. Просто если фреймворк хороший, и «зацепил», то есть видение автора фреймворка созвучно твоему видению и стилю разработки — то ты быстро привыкнешь, и будешь пользоваться с удовольствием.

Но в общем и целом — фреймворк это не только часть реализованного функционала, это ещё и некий общий стиль написания кода (принцип «делай как я», как писали ниже), и ряд требований.

Если сам язык программирования — уже во многом определяет правила и требования, то есть ли смысл ограничивать себя сверху ещё больше, работая в рамках конкретного фреймворка? Ведь когда ты пишешь сам по себе — ты волен писать как угодно (в рамках разумного, конечно, твой код ещё другим потом читать), и экспериментировать со стилем и архитектурой. Всё, что в целом компилируется и не выдаёт ошибок и предупреждений — уже как минимум «нормально», а до «хорошо» и «отлично» уже дальше можно расти самому. Хороший фреймворк в этом росте помогает — но имхо, и творчество сильно сдерживает. И поиск оригинальных решений тоже.

rebus_x 24 сентября 2018 в 12:49

–4

Не программист (кто хочет — может, прочитав нижеследующее, бросать помидоры, если станет легче). Но т.к. статья по сути касается любого человека, пользующегося современной электроникой — тоже выскажусь.

Считаю, что непременным шагом к той цели, которая обозначена в статье — скорость, надёжность, простота технологий — является национализация теоретической базы программирования и компьютерных технологий вообще, а именно — чёткий, грамотный, точный, единообразный, понятный, утверждённый-согласованный перевод всего (по возможности) понятийного аппарата на родной / национальный язык программистов/инженеров данной страны (если говорить обо всём мире). Чего греха таить — это ключ к упрощению и улучшению технологий в данной стране в любой области. Просто потому, что инженерам в данной области становится в разы проще мыслить.

Стоит ли говорить, что сейчас более чем в половине случаев «перевод» (если он вообще предлагается) той или иной технологии или подхода дословный, и часто никак не раскрывающий смысл, не помогающий понять суть технологии или подхода.
(Пример: логические (двоичные) действия «AND, OR, NOT, XOR». Некогда их «перевели» как «И, ИЛИ, НЕ, исключающее/исключительно ИЛИ». Как слово «не» должно объяснить, что значение будет изменено на противоположное? Нет ли слово «и» хоть какой-то намёк на двойственность, на то что «два значения одинаковы»? Не несёт, потому что слово «и» обычно используется для перечисления или соединения чего бы то ни было. Почему надо ломать собственное воображение и понимать простейшие для понимания крохотные слова родного языка извращённо, говорить себе что «и — значит оба»? Почему сразу было не назвать «AND — ОБА», например?
Кстати, речь идёт о битах. Попробуйте ребёнку объяснить, что такое бит. Если я скажу, что бит — это лампочка, она может либо гореть либо не гореть — понятный пример? Почему было не перевести «bit» как «лампочка» или похожим образом?
Надеюсь, на этих двух примерах суть понятна.)

Ну а когда перевода не предлагается вообще — это самый тяжёлый случай, ибо человек вынужден переводить мозг на работу «в режиме иностранного языка». В лучшем случае, если человек вникает в то что делает, часть мощностей мозга по сути занята переводящей работой, вместо того чтобы заниматься работой полезной; разбираться в том, что непонятно, становится в несколько раз сложнее.
(Некогда изучал один курс по 3хмерной графике, где преподаватель на полном серьёзе пренебрегал словами «растягивать»-«плющить» и говорил «стретчить»-сквошить» (stretch, squash) — как пример «побочного действия» погружения в работу на чужом языке. С остальными «объяснениями» преподавателя всё было ещё печальнее.)

Чем всё это плохо?
Во-первых, вместо того чтобы заниматься полезным делом, вы учите иностранный язык, чтобы вообще заиметь возможность к делу приступить.
Во-вторых, вы постоянно напрягаетесь, чтобы понимать, что вы делаете и с чем работаете, потому что все названия, понятия, условия — всё это иностранное, и создано иностранцами. А это, помимо напряжения мозга для задачи постоянного перевода с одного языка на другой, загоняет вас в рамки мышления, которое вам чуждо. Иностранцы мыслят иначе, рассуждают иначе. Вы бы очень многое делали по-другому, чем они — но используя то, что вам досталось от них, вы вынуждены действовать только так, «как дают». Это приводит к тому, что вы вынуждены в вечную «позицию догоняющего», притом что часть ваших сил тратится на усилия вида «понимать, что я вообще делаю». Вы обречены не изобретать ничего своего, а только пользоваться тем, что изобретут «там».

Да, не спорю, что для тех же, условно говоря, «международных» разработок с открытым исходным кодом (или полностью готовых к «международной» разработке, к разработке «всем миром») необходим международный же язык кода. Но утверждать его, а также сопутствующие правила совместной работы — это решаемый вопрос. Всем будет лучше, если программист сможет мыслить на родном языке, и только писать и изъясняться — в случае необходимости — иногда придётся на иностранном.
Молчу, что двум русским / русскоговорящим людям обмениваться кодом на иностранном языке, не преследуя и не закладывая цели международной работы над этим кодом, просто глупо. Можно с тем же успехом «просто потому что» разговаривать друг с другом, работая над чем-то, на иностранном языке, исключив родной. Каков смысл?

Перевод заимствованных подходов на понятный родной язык — это только первый шаг. Дальше из национальной теоретической базы будут сами собой вытекать национальные же технологии, не придётся брать основную массу технологий из-за океана, исключается «позиция догоняющего». Люди получат возможность, условия, разрабатывать так, как работает их национальное мышление.
Нет ничего удивительного в том, что основная масса подходов и технологий в программировании и в ИТ приходит из США — просто американцы занимаются всем этим на родном языке. Поставьте остальных в равные условия с ними, и картина изменится.

Ну и до кучи просто красноречивое голосование в конце.

Путь ко всему этому тяжёл, через ломание привычек, ломание бюрократии и когорты, заинтересованной в том чтобы только заимствовать и ничего не изобретать, совершение гигантской работы над имеющейся и использующейся теоретической базой (после предварительного её сбора, кстати). Но начать можно с того, чтобы всё, что вы изучаете или кому-то объясняете, качественно переводить на родной язык (когда вы говорите об этом или обдумываете это), чтобы вы хорошо понимали и вас хорошо понимали. К счастью, я наблюдаю что чем более зрелым и опытным становится специалист, тем больше его тяга к простому и понятному языку — причём как у соотечественников, так и у иностранцев.

lxsmkv 24 сентября 2018 в 16:27

+1

Недавно смотрел видео Как Швеция стала лидером игростроения
Так вот, одной из предпосылок к этому стало внедренное государством изучение английского языка с первого класса. А также льготы фирмам которые обеспечивают своих сотрудников компьютером для личных нужд.

Не перенос мировых знаний в свою среду, а создание необходимых условий для интеграции трудового населения в мировую индустрию.
В Бельгии фильмы не дублируют — все фильмы на английском с бельгийским подстрочником. В Хорватии фильмы дублированы на хорватском с английским подстрочником.
Это кажется странно, но это правильно. Если ты не говоришь на языке которым пользуется подавляющая масса людей — ты всегда во вторых рядах. Всегда в роли догоняющего.
Я вам это из личного опыта интеграции в немецкую среду могу подтвердить. Отличных мозгов не достаточно. Чтобы тебя слушали, нужно владеть языком на высоком уровне. Выше разговорного.

rebus_x 24 сентября 2018 в 17:43

–1

Чем занимается эта самая «мировая индустрия» — чем-то хорошим? По сути этот вопрос проходит через всю статью, и ответ на него — нет, назрела необходимость перемен. Чтобы от кем-то там когда-то придуманных и неоправданных стандартов (по копированию готовенького) переходить к тому, что будет осмысленно и сделано на совесть.

Пусть автор этого и не пишет, но надо мысленно дополнить статью следующим: сейчас это (переход к работе «на совесть») никому, кроме простых честных трудяг, не интересно, а интересно организациям всех уровней только побыстрее скололотить денежек. И никаких предпосылок к тому, чтобы эти интересы поменялись, нет и не будет. Поэтому желанные перемены могут наступить только при условии настолько же значительных перемен в устройстве общества, в сознании людей — когда большинство людей будет регулярно себе напоминать, что деньги на тот свет не заберёшь, и потому резаная бумага перестанет быть первичной целью.

А в том, что вы пишете, я позволю себе вас поправить.

> Если ты не говоришь на языке которым пользуется подавляющая масса людей — ты всегда во вторых рядах.

Так сложилась история, что после падения одной из двух крупнейших сверхдержав — СССР — вторая сверхдержава смогла позволить себе распространить своё влияние на всю планету, поскольку ей больше никто не мешал. Однако до того, как это произошло, мир был поделён на два блока, которые успешно вели гонку технологий, пройдя несколько поколений этих самых технологий. И более того — небогатый в общем-то социалистический блок, не успевший к началу противостояния оправиться после Второй мировой, преуспел в отдельных технологических областях, и СССР оставил России задел, которым она пользуется до сих пор. Без того, чтобы «вливаться» куда бы то ни было.
(Я как бы молчу про то, что в России одарённейшие люди и огромный потенциал для того, чтобы в хорошем смысле переворачивать мир.)

Что касается обрисованного вами — вход в уже готовую индустрию, созданную не тобой, не является тем самым успехом, даже если ты занял первую строчку в списке. У индустрии есть хозяева, и ты не в их числе. Никто тебе не даст брать управление в свои руки.
Доказательство: выпускаемые той же Швецией (и кем угодно) игры (дорогие, ААА-уровня, от студий) «почему-то» соответствуют типичным голливудским шаблонам.
Так где же тут «роль ведущего», я не понял?

И я так тихонько ~~напомню, если вам с политических трибун мало было за 10 лет~~ добавлю, что всё это есть глобализм (игра по чужим правилам), время которого уже ушло. Сейчас разные центры силы в мире будут тянуть на себя одеяло.

OruSeUkG 1 октября 2018 в 11:08

+1

К сожалению, когда существует несколько «центров силы», происходит распыление этих самых сил, уж простите за тавтологию, и каждый из независимых центров может меньше, чем один совокупный. Именно поэтому люди с течением времени постепенно объединяются. Вроде бы это самоочевидно? И уж коль вам зашла за СССР, позволю себе напомнить, что в книгах столь почитаемых в СССР научных фантастов человечество всегда представлялось единым. Как будто бы никого тогда это особенно не задевало.

А сейчас слово «глобализм» — уже чуть ли не ругательное. Можно сколько угодно говорить о том, что время глобализма ушло, что глобализм загнивает. Во времена оно официально утверждалось, что и капитализм загнивает. Что загнило в действительности — все прекрасно видят.

Боюсь, однако, что осуждающие глобализм осуждают его вовсе не потому, что он плох сам по себе. Просто эти осуждающие живут за пределами доминирующего центра силы, а потому завидно. Вот и всё. Обидно людям, что жили они раньше в «одной из двух крупнейших

сверхдержав — СССР», и могла эта держава нагибать сателлитов, как хотела. И никто не считал это глобализмом, т. е. «игрой по чужим правилам». А теперь мы никого нагибать не можем, хотя имперские замашки остались — мы же все такие из себя огромные, широкие и вставшие с колен.

А насчет «простых честных трудяг» — вы уж меня простите… Как раз простые честные трудяги работают за денежки. Или вы, может быть, отказались в своей организации от зарплаты? «И никаких предпосылок к тому, чтобы эти интересы поменялись, нет и не будет» — вот тут я с вами полностью согласен. Что же до наступления «желанных перемен», то… для кого они желанны? Мы уже строили государство простых честных трудяг, и что из этого получилось — хорошо известно. А как выглядели бы государства, в которых всё устроено рационально, как положено, хорошо описано у Замятина («Мы») и Оруэлла («1984»), И человечество потому и не сгинуло до сих пор в пучине тотальной диктатуры, что устроено нерационально!

Человек в принципе не рационален. И слава богу! Люди-то ведь все разные. А всякий ревнитель рациональности понимает её по-своему. Добравшись до власти, он неизбежно начинает всех «нерациональных» или «неправильно рациональных» (в его понимании) в начале исправлять, а потом — искоренять. Тут же появляются внутренние враги (которые почему-то не хотят быть рациональными) и наймиты внешних врагов (которые почему-то опасаются, как бы и их рационализировать не взялись) и т. д. Я думаю, продолжать не имеет смысла.

А ещё добавок все рационализаторы почему-то так устроены, что как только они начинают претворять свои химерические замыслы в жизнь, они не могут ограничиться только одной, изначальной областью. Они начинают рационализировать ВСЁ вокруг. Почитаете Ефремова: во вселенной «Туманности Андромеды» и «Часа быка» рационализировано всё, что можно. Читать это, может быть, и интересно (хотя на самом деле скучно), а вот жить-то в таком мире вы бы хотели?

rebus_x 1 октября 2018 в 13:23

−1

> К сожалению, когда существует несколько «центров силы», происходит распыление этих самых сил, уж простите за тавтологию, и каждый из независимых центров может меньше, чем один совокупный. Именно поэтому люди с течением времени постепенно объединяются. Вроде бы это самоочевидно? И уж коль речь зашла за СССР, позволю себе напомнить, что в книгах столь почитаемых в СССР научных фантастов человечество всегда представлялось единым. Как будто бы никого тогда это особенно не задевало.

Верно. Но во-первых: объединение может выглядеть и выполняться по-разному. Предлагаемый глобализмом вариант — стереть границы, смешаться всем со всеми, сделать одно правительство и «новый мировой порядок» — это вариант паразитически-маразматический, и к тому же чисто расистский, потому что граждане, планирующие себя на роль правителей, про свою собственную «расу» (например) не забывают, и смешиваться ни с кем не хотят. Подробности читайте в древних иудейских источниках.

Правильный путь объединения — сохранение границ, сохранение национально-культурной самобытности всех народов и стран (не государств, а стран — территорий проживания народов). Единственное, что должно быть объединено — мировоззрение и цели. А именно: необходимо обеспечить мирное сосуществование народов (стран); необходимо перестать гадить в природу; необходимо создать все условия, чтобы человечество вышло в Космос.

Во-вторых, очень глупо и самоубийственно объединяться в «единый мировой порядок» в предлагаемом виде, когда строится общемировой электронный концлагерь, а людям предлагается быть шестерёнками или батарейками. Такая заражённая паразитизмом цивилизация ничего не достигнет и никуда не пойдёт, и не намерена никуда идти. В Космосе она тоже никому не нужна — скорее такая планета будет искусственно изолирована, чтобы вирус не распространялся.

> Во времена оно официально утверждалось, что и капитализм загнивает. Что загнило в действительности — все прекрасно видят.

Я вижу сплошную ложь; вижу, как многие люди учатся обманывать, считая что это и есть «успех»; вижу перепроизводство под песенку про то, что «спрос рождает предложение»; вижу ненадёжность, убогое качество производимых вещей, ибо это выгодно для того чтобы побольше стричь потребителя; вижу науку, которая не может и не хочет объяснять простейшие вопросы об окружающем мире; вижу медицину, которая не умеет лечить; вижу загаженный океан и «прорывные космические технологии» (ракеты, средневековая технология), которые уничтожают озоновый слой атмосферы; вижу, как целая страна с богатейшим человеческим капиталом сходит с ума в погоне за деньгами, тупо копируя всё чужое и далеко не самое лучшее. По комментам в англоязычной Сети я вижу, как большинство населения западных и заокеанских государств уже насытились по горло политкорректностью и толерастией.

Вонь стоит по всему миру. По всему миру капитализм. Так вопрос — что же загнило?

> завидно

Да, всему вышеперечисленному. Притом, что мы всё вышеперечисленное имеем на руках. Ничего больше не смогли придумать?

> могла эта держава нагибать сателлитов, как хотела. И никто не считал это глобализмом, т. е. «игрой по чужим правилам»

Ну какбе считали, если вы имеете в виду социалистический блок и всё, что Союз мог ему навязать. Даю гарантию, что никому не нравится, когда твоя страна пляшет под чью-то дудку или работает на кого-то.

Если вы, внезапно, не о том, то я вообще не понял про глобализм, определитесь к чему это было написано, учитывая что глобализм есть единый одинаковый порядок для всех по всей планете.

> А теперь мы никого нагибать не можем, хотя имперские замашки остались — мы же все такие из себя огромные, широкие и вставшие с колен.

Ну если вы от лица владельцев США пишете, то вы полностью правы, действительно уже не можете, но замашки никуда не делись) Встали же с колен после разграбления павшего СССР, правда награбленного надолго не хватило)

> А насчет «простых честных трудяг» — вы уж меня простите… Как раз простые честные трудяги работают за денежки. Или вы, может быть, отказались в своей организации от зарплаты? «И никаких предпосылок к тому, чтобы эти интересы поменялись, нет и не будет» — вот тут я с вами полностью согласен. Что же до наступления «желанных перемен», то… для кого они желанны? Мы уже строили государство простых честных трудяг, и что из этого получилось — хорошо известно. А как выглядели бы государства, в которых всё устроено рационально, как положено, хорошо описано у Замятина («Мы») и Оруэлла («1984»), И человечество потому и не сгинуло до сих пор в пучине тотальной диктатуры, что устроено нерационально!
>
> _ и далее про «рациональность»

Усматриваю приём профессиональной демагогии. Вы придумали и нарисовали некую мою позицию, и делаете вид, что моя позиция такая, какой вы её придумали, и пытаетесь возражать чему-то, что сами придумали.

Начнём с того, что если вы решили что я сторонник коммунизма или восстановления СССР в прежнем виде, то это не так.

«Как раз простые честные трудяги работают за денежки» — здесь вы уводите в сторону, я писал о труде «на совесть», а не о деньгах. Если вы намекаете на мою позицию отказа от денег в целом, и что я якобы предлагаю людям пахать ни за что — это тоже не так, и я такого не предлагал и вообще об этом не писал. Справедливую систему обмена ценностями, благами и т.д. можно разработать и создать, было бы желание. И да, без денег и без всех приносимых ими последствий.

«наступления «желанных перемен», то… для кого они желанны?» — автор статьи пойдёт? Я, поддерживающий его — подойду? Другие люди, которые его поддержали в комментариях? Вообще любые люди, кому тоже надоело всё перечисленное в статье?

popov654 18 октября 2018 в 10:00

0

> преуспел в отдельных технологических областях

В информатике, насколько я помню, всегда отставал. Значительная часть технологий просто втупую копировалась с иностранных образцов (отдельные вычислительные машины и даже языки программирования).

> и потому резаная бумага перестанет быть первичной целью

Это вопрос философии каждого конкретного человека и его взглядов на жизнь, политика и языки тут ни при чём.

> Чем занимается эта самая «мировая индустрия» — чем-то хорошим?

Ну вы знаете, есть хорошие продукты, всех не стоит под одну гребёнку грести, имхо :)

rebus_x 18 октября 2018 в 13:20

0

> В информатике, насколько я помню, всегда отставал. Значительная часть технологий просто втупую копировалась с иностранных образцов (отдельные вычислительные машины и даже языки программирования).

Да, это так, однако степень сложности задач, которые успешно выполнялись — например, помимо тупого копирования, полный обратный инжиниринг и собственное производство в уменьшенных размерах (однокристальная микросхема 1801ВМ1 против неоднокристального «pdp11») — немаленькая.

Что касается копирования, следования за кем-то — это было политическим решением на том или ином уровне, проще говоря сабота:жем. Можно обратить внимание, что электроникой занимались два министерства сразу, и неясно, как именно между ними были распределены задачи (что безусловно увеличивало заборокрапчатость всего, что делалось). Инженер Горохов разработал устройство, которое вполне подходит под звание прототипа ПК, заявка на патент №383005 подана в 1970-м — но устройство не пустили в производство, а через несколько лет Джобс и Возняк выпустили Эппл-1. Для меня это выглядит так, что заявку советского инженера преднамеренно придержали, пока её не прочитают друзья-товарищи-партнёры по вопросам изобретений и патентов.

Так что и до причин, почему было именно так, можно докопаться.

> Это вопрос философии каждого конкретного человека и его взглядов на жизнь, политика и языки тут ни при чём.

(про резаную бумагу)

Политика (в широком смысле) ого как причём, поскольку именно она определила то, что деньги в мире существуют. А после этого ею, по сей день, определено навязывать людям деньги как (ложную) ценность.

> Ну вы знаете, есть хорошие продукты, всех не стоит под одну гребёнку грести, имхо :)

Согласен, однако в статье речь идёт об общих веяниях, и видимо, плохого в мире пока больше, чем хорошего.

popov654 23 октября 2018 в 03:32

0

> Политика (в широком смысле) ого как причём, поскольку именно она определила то, что деньги в мире существуют.

А вы представляете мир без денег? Как будем вести обмен услуг и товаров?)

> а через несколько лет Джобс и Возняк выпустили Эппл-1

Мне не кажется, что они скопировали эту идею, а не *дошли* до неё сами, если честно.

> Для меня это выглядит так, что заявку советского инженера преднамеренно придержали, пока её не прочитают друзья-товарищи-партнёры по вопросам изобретений и патентов.

Зачем такое было делать? Это же целенаправленное вредительство. Тайные агенты, завербованные ЦРУ что ли?

rebus_x 23 октября 2018 в 12:53

–1

> А вы представляете мир без денег? Как будем вести обмен услуг и товаров?

Сперва важная оговорка — что это будет возможно только вместе с переустройством общества в целом, ибо для того чтобы нижеследующее работало, там должны выполняться два условия:
1) каждый человек должен знать, чем хочет заниматься, к чему имеет предрасположенность, чётко представлять себя свои качества, достоинства и недостатки; что у него получается хорошо, а что не очень, что ему нравится а что нет. Это задача системы образования, условие предварительное и должно соблюдаться в дальнейшем.
2) все люди по определению, по факту своего существования, обеспечиваются питанием и крышей над головой. В целом условие не является предварительным, можно запускать его одновременно с нижеследующим.

Я бы предложил подход «ты мне — я тебе», но дополненный, а также подкреплённый современными возможностями связи людей друг с другом.
Суть: каждый человек а) волен заниматься чем хочет, делать что хочет (разумеется, не совершая преступлений); б) может обратиться к другому человеку/нескольким людям за той или иной помощью (можно сказать «услугой»), чтобы получить то что ему нужно (помочь с каким-то делом, получить какую-то произведённую вещь («товар» — без всякого ценника, просто попросить и получить) и так далее — что угодно что потребуется).

Для этого они сообщают о своём запросе. Все люди, которые в данное время готовы помогать другим (читай — заниматься любимым делом, трудом к которому предрасположены), тоже сообщают о себе и о том, что они умеют хорошо делать / производить. Обе стороны взаимно видят списки (условно говоря) друг друга, и договариваются.
При этом желательно, чтобы участники сделки договорились о том, что другая сторона оплатит ответным соразмерным трудом (ты мне — я тебе). Но в крайнем случае принимающая сторона может «заплатить» тем, что человек, который выполнил работу, получит отметку «заслуга» с указанием, какая работа была выполнена / что было произведено (чтобы можно было предложить за неё что-то соразмерное).
Одна «заслуга» позволяет получить чью-либо помощь / услугу, не отплачивая в ответ.

Сразу понятно, что чтобы обеспечить всех питанием и жильём, чтобы поддерживать общественный порядок и обеспечивать прочие вещи из разряда «необходимого» и «необходимого производства», нужен труд, который не обязательно будет нравиться всем его совершающим — однообразный, волокитный, тяжёлый. Все, кто таким трудом занимается, имеют «вечную заслугу» (ты нам — мы тебе) всё время, пока занимаются им, и на какой-то дополнительный срок после смены деятельности.

На вопрос «а почему кто-то будет помогать за „заслуги", не получая ничего взамен» ответ — в силу личного признания заслуг этого человека перед обществом. Например, к пожарным, как правило, относятся с большим уважением, и не сомневаются в том, что они достойны благодарности окружающих.

Общий смысл всего этого: дать всем возможность заниматься творчеством, саморазвитием, созиданием — любое желаемое количество времени. Сделать так, чтобы труд каждого человека был вкладом на пользу всем остальным людям.

Что всё это даёт:
— у всех всё есть. Первое время возможна «мания богатства», но в силу своей бессмысленности она быстро прекратится.
— все заняты (полезным) делом. Любой волен бездельничать, но ни один человек на это не способен — полезет на стену от тоски и скуки.
Чем можно заняться, если все личные дела сделаны, не хватает воображения, творческих сил и так далее:
очисткой природы по всей планете от загрязнений;
сборкой и переработкой информации, накопленной человечеством;
участием в обсуждении и решении общественных вопросов;
участием в совершенствовании технологий, переводе их на безвредные и природоподобные рельсы;
воспитанием будущих поколений.
— возвращается свободное и непринуждённое общение между людьми, отношение к ближнему как к соседу.

Исходя из того, что людям становится гораздо проще жить, естественным образом происходят следующие вещи:
— люди становятся лучше, преступления становятся исключением, а не правилом
— люди умеряют свои запросы
— люди осознаЮт со временем необходимость бережного отношения к занимаемой среде, природа вздыхает с облегчением.

Это как костяк, подробности обсуждаемы.

> Это же целенаправленное вредительство. Тайные агенты, завербованные ЦРУ что ли?

Я не готов сейчас точно говорить, что и как было организовано, однако хотя бы история с Горби намекает, что во власть СССР можно было проникнуть, если захотеть. Шпионские истории с внедрением куда угодно, опять же. Материала про всё это хватает.

DMGarikk 23 октября 2018 в 14:05

+1

> получит отметку «заслуга» с указанием, какая работа была выполнена

ага… а потом будет у каждого человека счет заслуг… типа 500заслуг, за которые можно в ресторан сходить и поесть… но погодите, если мы поменяем слово «заслуга» на «рубль»… получатся обычные деньги! ;)
==
Вы не учитываете что есть много профессий которые не понятны многим для оценки. Удобней конечно ИТ в пример привести, но я приведу в пример более приземленный например маляр который ЖД вагоны красит снизу… кто будет значимость его, несомненно очень вредной и в тоже время для большинства людей «не нужной», работы оценивать? сколько «заслуг» он получит чтобы автомобиль себе «получить»? (вы давно видели вагон метро снизу? а многие никогда его не видели и не увидят, так зачем его красить?)

rebus_x 23 октября 2018 в 14:32

–1

> сколько «заслуг» он получит чтобы автомобиль себе «получить»?

Вредная/тяжёлая работа — «вечная заслуга», см. выше. Но вы не поняли. Любой человек в любое время может сказать, что ему нужен автомобиль, и он его получит (ну, совершеннолетний, с водительскими правами и всё что полагается). И так с любой вещью. (Контроль за опасными вещами — само собой, государство в том или ином виде я никуда не выбрасываю.)
Если спросите «и что, они (автомобили) будут везде?» — да, какое-то время, пока после этого люди не поймут, что это безумие, даром никому не нужное, после чего договорятся между собой прекратить, и количество автомобилей резко уменьшится. Скорее всего ничего подобного не случилось бы, но я рассказываю это чтобы описать сам подход.

DMGarikk 23 октября 2018 в 14:47

+1

вы описываете сферическую утопию в вакууме

> безумие, даром никому не нужное

это применимо только для людей живущих в крупных городах, вы не забывайте что за МКАДом есть жизнь

> личного признания заслуг этого человека перед обществом

И как общество будет признавать заслуги маляра? он же «ничего полезного не делает» для общества
А заслуги инженера работающего на военном заводе? или будут люди с «вечными заслугами» которые получены в результате (сов.секретно)… и как сейчас мажоры ездят по встречке… также и такие заслуженные люди будут делать… откуда получили заслугу? неважно, это секретно.
==
«люди поймут и осознают» — не поймут и не осознают

rebus_x 23 октября 2018 в 15:02

0

> И как общество будет признавать заслуги маляра?

Я же расписал — обществу ничего признавать не надо. Если кто-то маляр, то кому-то другому надо покрасить стены. Маляр говорит «я могу (и люблю, и хочу) красить стены» — другая сторона говорит «мне надо покрасить стены». Если кто-то чего-то не хочет — им это делать не обязательно. Две стороны друг друга видят, вторая сторона либо может оплатить маляру ответным делом (починить машину, например — потому что вторая сторона это механик), либо дать ему «заслугу», чтобы ему отплатили тем делом, которое ему нужно (изготовить маляру новые спецовки, изготовить ему краску — такой трудяга маляр, что дай только поработать).

DMGarikk 23 октября 2018 в 15:30

+1

Чтото я сомневаюсь что есть люди которые любят/хотят работать малярами (подвагонного оборудования) или ассенизаторами например
потом, возьмем ЖД депо (пассажирское), оно ремонтирует по 10-20 вагонов в месяц, и мне затруднительно представить какую услугу оно может оказать маляру чтобы занять его работой на каждый день и при этом эта услуга была бы маляру нужна

rebus_x 23 октября 2018 в 16:02

–1

Тяжёлая необходимая работа = заслуживаешь безвозмездных услуг. Безвозмездная услуга — это та, на которую не надо отвечать ответной услугой.

> ЖД депо (пассажирское), оно ремонтирует по 10-20 вагонов в месяц, и мне затруднительно представить какую услугу оно может оказать маляру чтобы занять его работой на каждый день

Ничего не понял. Если маляр покрасил (один раз) в этом депо вагоны, то либо ему ответная услуга, либо если эта услуга ему не нужна — то получит от других людей соразмерную услугу/вещь, которая нужна.

staticlab 23 октября 2018 в 16:29

+1

Как определить соразмерность услуги или вещи?

rebus_x 23 октября 2018 в 16:41

0

Приблизительно. Вероятно, по количеству затраченного труда. Если речь идёт о вещи — то труда, потраченного на её изготовление / производство (потому что вещь тоже надо будет произвести).
Остальное — предмет договорённости.

staticlab 23 октября 2018 в 16:53

+1

И чем это отличается от денег? В СССР деньги тоже считались мерой труда, и расчётом цен занимался Госплан. Коммунизм, однако, так и не построили.

rebus_x 23 октября 2018 в 17:10

−1

Здесь коммунизм и не строится)
От денег в их нынешнем виде отличается тем, что нынешние деньги имеют кредитную природу, выпускаются под проценты, являются ложной ценностью.
От денег в их «правильном» виде (когда их можно было бы заменить билетами / талонами / другими фантиками или золотыми пластинками, когда они чем-то обеспечены и являются просто средством обмена) отличие в том, что деньги нужны в условно средневековом обществе, когда ещё нет производственных мощностей, возможности обеспечить всех едой и питьём; когда надо обменять труд на выживание (купить себе еду, заплатив при этом, чтобы тот человек которому ты заплатил — в свою очередь мог заплатить, чтобы выжить, и так далее).
Когда же есть возможность обеспечить выживание каждому (при определённом уровне развития общества и производственной базы), деньги уже лишние, ни к чему от них зависеть. Здесь предлагается обмен своего времени на чужое время, своего труда на чужой.

staticlab 23 октября 2018 в 17:25

+1

"Как эквивалент всех товаров деньги при социализме являются всеобщей формой учёта затрат общественного труда, осуществления планирования и организации производства и распределения общественного продукта в соответствии с экономическими законами социализма." — БСЭ

Вы именно, что коммунизм нам тут предлагаете.

rebus_x 23 октября 2018 в 17:36

−1

Честно говоря, мне без разницы кто как будет это называть, потому что эти названия работают как штампы. Главное, что я сам точно знаю, что не предлагаю что-либо у кого-либо отбирать и объявлять собственностью рабочих.

popov654 23 октября 2018 в 17:41

−1

И это не есть плохо. Если вы знаете идеи коммунизма — я не думаю, что у вас что-то вызовет там негатив. «Каждому по потребностям», классно же

DMGarikk 23 октября 2018 в 17:00

0

> то получит от других людей соразмерную услугу/вещь

а как другие люди поймут что надо этому человеку оказать соразмерную услугу? вот вы можете оценить «сколько стоит» работа маляра? допустим вы арбузы на рынке «отдаете» в Краснодаре и к вам маляр подошел, который вагоны в Москве красит

rebus_x 23 октября 2018 в 17:15

0

Ну не в прошлом веке же, связь и информационные системы есть. Со мной связывается человек (маляр), потому что я предлагаю арбузы (питанием и любой едой все и так обеспечены, но ладно). Он говорит, что ему не отплатили, и у него есть «заслуга». Я проверяю, вижу что это действительно так, решаю что согласен закрыть его «заслугу», мы договариваемся сколько арбузов я ему отдам. За «заслугу» отплачено.

popov654 23 октября 2018 в 17:43

0

Интересно… Только договориться будет сложно. Я бы одновременно боялся и обидеть вторую сторону предложением слишком низкой оплаты, и боялся бы потерять в итоге сам. Сейчас-то проще — приходишь в магазин, платишь, сколько на ценнике написано :)

rebus_x 23 октября 2018 в 18:55

−1

Просто пример неудачный, с арбузами. На практике никаких «базарных» обменов не предполагается, будет затребована конкретная услуга / вещь.

DMGarikk 23 октября 2018 в 18:03

+1

> Я проверяю, вижу что это действительно так, решаю что согласен закрыть его «заслугу», мы договариваемся сколько арбузов я ему отдам. За «заслугу» отплачено.

а не проще назвать заслугу — деньгами? а инфосистема будет банковским счетом, и больше не мучится? вы всеравно не сможете оценить «правильность» и «ценность» заслуги в острасли которой не разбираетесь?

rebus_x 23 октября 2018 в 19:08

−1

Не проще. Деньги вам дают, чтобы вы покупали главным образом еду, чтобы вы могли выжить. Здесь вам уже обеспечено выживание, а трудитесь вы по собственному решению, добровольно, на благо общества.
Не стоит к слову «заслуга» придираться, это условное обозначение, смысл которого — показать, что человек помог кому-то, но ему ещё не отплатили. Единственное, я считаю оправданным указывать, что именно делалось, чтобы за протёртую пыль на площади 1 м2 человеку не платили 4-этажным особняком.

DMGarikk 24 октября 2018 в 08:34

0

ну хорошо, придрались к еде
===
Лопату купить, поездку на море, транспорт до работы, новый телефон… любое подставьте

а теперь как вы представляете продавца путевок который сидит и проверяет доходы КАЖДОГО покупателя чтобы лично оценить «таак, он пыль протёр — не заслуживает, таак, а этот картошку окучивал — заслуживает»… а потом начнется, я не с пустого места маляра привел… шахтер посчитает что маляр вообще ничего не делает по сравнению с его работой, а программист так вообще халявщик
==
и не забывайте что есть очень много видов товаров которые не делаются одним человеком, тотже компьютер например, как вы представляете себе тиражную сборку компьютера через такую цепочку оказания услуг друг-другу?

rebus_x 24 октября 2018 в 09:21

0

Я вижу, что вы всё равно не поняли. Если уж вас так это заинтересовало — перечитайте.
Ничего покупать не надо. Если что-то кому-то понадобилось, он просто это получает (подробности ниже). При этом всё, что является необходимостью — питание, жильё, всякие государственно-правовые вопросы (напр. помощь государственных служб, экстренных служб), поездки/транспорт — обеспечено каждому. Как платят тем людям, которые взялись выполнять эту необходимую работу, я писал выше.
Получается, что каждый человек может решить, как ему проводить своё время. И если у человека настроение поработать — он может предложить обществу свои услуги.

Если то, что человек решил делать, НЕ относится к «необходимым» (см. выше) работам — то всё, что касается сделки, решается между людьми в личном порядке. Поэтому предполагается, что принимающая сторона либо отплатит ответным трудом, либо даст право тому, кто работал, в свою очередь получить от кого-либо услугу, но безвозмездно.
И просто в целях справедливости — нельзя за какое-либо небольшое дело получить огромную отплату, и наоборот — за большой труд человек вправе ожидать большого труда в ответ.

Ваш пример — надо новый компьютер. Человек находит тех людей, которые могут производить компьютеры, обращается к ним, они изготавливают компьютер (и этих людей несколько). Каждый из этих людей работал, следовательно отплатить надо каждому из них. Если есть что предложить в ответ — идёт ответная работа, если нет — каждый из них может получить от других безвозмездную услугу (любую соразмерную или — по желанию — меньшую, поэтому они могут выбрать то, что им надо, найдя соответствующих людей).

Понятия «дохода» нет, т.к. понятия «деньги» нет. Нет понятия «выгодно / невыгодно производить», поэтому (первое время) те же компьютеры можно делать поштучно, используя оборудование для массового производства (а кто запретит?). В дальнейшем технологическая база более гибко подстраивается под нужды общества.

DMGarikk 24 октября 2018 в 09:34

0

> обращается к ним, они изготавливают компьютер (и этих людей несколько)

Их не несколько, чтобы изготовить компьютер нужны десятки тысяч человек
ваша теория рассчитана на самообслуживание какихто небольших групп людей и не учитывает «тяжелые» технологии которые охватывают много людей
например перелет самолета из Москвы в Мехико (специально выбрал чтобы поменьше пассажиропоток)… за это отвечают сотни людей, причем работа многих лишь косвенно относится к этому перелету
Нельзя найти человека (пару людей) которые обеспечат ваш перелет и грубо говоря оказать им услугу по пошиву рубашки например… просто по чисто физическим соображениям
Сейчас перелет туда стоит относительно недорого (из-за размывания косвенных затрат на все рейсы авиакомпании, это буквально плата за керосин), даже несмотря на то что направление непопулярное, в вашей же модели оно будет совершенно неподъемным поскольку «обществу это не нужно», а индивидуум не сможет договорится со всеми этими тысячами людей о бартере

rebus_x 24 октября 2018 в 09:47

0

> чтобы изготовить компьютер нужны десятки тысяч человек

Притом, что есть всё готовое для производства, потому что всё предусмотрено? Потому что специальные люди занимаются разработкой, схемами плат и чипов, готовят программы для производственного оборудования (что является, см. выше, «необходимым трудом», потому что это технологическая база общества)? Заготовлены, стандартизированы необходимые радиокомпоненты?
Допустим, один человек = один специалист по изготовлению той или иной части (проц, видеокарта, БП, запоминающее устройство, материнка, ОП, кабели, корпуса, монитор, клавомышь). Им помогают роботы, станки — всё то, что есть на заводах. Откуда десятки тысяч человек?

Про перелёты — ну сколько можно. Цитирую себя:

> поездки/транспорт — обеспечено каждому

Я за сохранение государственных границ ради порядка, но при этом предполагаю, что такое общество установится по всей планете (никакого противоречия), и вопросы поездок по всей планете — это из разряда того, что обеспечено и не обсуждается.

DMGarikk 24 октября 2018 в 10:48

0

кем предусмотрено? все люди же работают по вашей схеме, услуга-за-услугу
один человек спец по изготовлению процессора? вы в курсе как устроен и производится микропроцессор? это огромная и крайне высокотехнологическая фабрика которых в мире совсем немного а десятки тысяч человек — это доставка сырья, топлива, энергии, проектирования, испытания… сверху это уборщики, работники служб безопасности, контролирующих органов и т.п. тут цепочку вплоть до добычи кремния и нефти протянуть можно

или вы выделяете некоторых людей которые будут работать на некоторых предприятиях которые не входят в вашу модель?

кто будет «заготавливать эти компоненты»? кто будет заказчик этой «услуги»? какие «услуги» будут платится в замен людям занятым на этих работах?

> поездки/транспорт — обеспечено каждому

Самолет очень дорогая штука чтобы обеспечить его каждому, даже нынешних ресурсов на это не хватает
сколько надо самолетов чтобы отправить ВСЕХ желающих из РФ в Новую Зеландию? Построить кучу новых самолетов? если всем дать возможность ехать куда они захотят, то по первой ВСЕ захотят поехать туда где теплее-лучше… а там тупо места не хватит

rebus_x 24 октября 2018 в 11:03

0

> все люди же работают по вашей схеме, услуга-за-услугу

Цитата из моей схемы

> Сразу понятно, что чтобы обеспечить всех питанием и жильём, чтобы поддерживать общественный порядок и обеспечивать прочие вещи из разряда «необходимого» и «необходимого производства», нужен труд, который не обязательно будет нравиться всем его совершающим — однообразный, волокитный, тяжёлый. Все, кто таким трудом занимается, имеют «вечную заслугу» (ты нам — мы тебе) всё время, пока занимаются им, и на какой-то дополнительный срок после смены деятельности

Это значит, что они могут безвозмездно получать услуги других.

> специальные люди занимаются разработкой, схемами плат и чипов, готовят программы для производственного оборудования (что является, см. выше, «необходимым трудом», потому что это технологическая база общества)

> вы в курсе как устроен и производится микропроцессор? это огромная и крайне высокотехнологическая фабрика

И производятся там сегодня сотни тысяч экземпляров, поэтому нужны десятки тысяч людей. Сколько надо для производства одной штуки, притом что сырьё готово, привезено — просто бери и используй?

DMGarikk 24 октября 2018 в 11:14

+1

> Это значит, что они могут безвозмездно получать услуги других.

прикольно, тоесть будут люди которые за то что они программируют чтото-где-то-там будут получать безвозмездно что захотят?
Вы игнорируете то что я пытаюсь до вас донести еще раз.__нельзя_оценивать_людей_по_труду__ потому что нет адекватной оценки ценности работы. то вы предлагаете делать услугу за услугу… то внезапно некоторые виды работ дают индульгенцию на получение безвозмездных услуг… и любая штука где фигурируют «безвозмездное право» сразу повлечет разделение достатков

ИТшники с точки зрения слесарей на заводе — ничего не делают и не нужны, по этому слесарь никогда не оценит свою услугу для программиста также как для другого слесаря
А если к слесарю придет 19 летний паренёк и начнет просить оказать ему безвозмездных услуг на «гуляй рванина, за всех плачу»… то получится перекос вашей системы

> Сколько надо для производства одной штуки, притом что сырьё готово, привезено — просто бери и используй?

ну вот мне захотелось Iphone 3… а его уже кучу лет как не производят… как «взять и использовать»?

rebus_x 24 октября 2018 в 11:48

–1

> нет адекватной оценки ценности работы

Назовите, что обеспечивается работой данного человека — вот вам и оценка.

Что такого невероятного может понадобиться человеку от (одного) слесаря?
И слово «безвозмездная» не означает «неограниченная», а значит только то, что не нужно отвечать на неё своей услугой. Всегда в действии личная договорённость людей друг с другом, которой многое определяется — каждая сторона может захотеть условий получше для себя, но им придётся договариваться, чтобы прийти к справедливому решению. Это и есть регулятор, притом очень привычный и естественный.

Учитывая, что в предлагаемом обществе все люди трудятся добровольно, нет никакой нужды грызть друг друга за выживание — справедливость договорённостей будет напрашиваться сама собой.

> ну вот мне захотелось Iphone 3… а его уже кучу лет как не производят

Начать можно с хорошего вопроса — где вы его возьмёте сегодня, кроме как с рук? Но это было отступлением.

Если предположить, что в ходе переработки и принятия «в наследство» технологий и разработок предыдущих поколений всё было сохранено (а иначе — увы, Айфон 3 не получится), то решение может быть таким:
находите людей, которые могут произвести предыдущие модели

электроники, либо организуете группу таких людей сами;
— если нашли таких людей
всё как описано выше.
— если организовали сами
вытаскиваете из наработанных архивов всю нужную документацию, разбираетесь что вам требуется для производства;
берёте всё необходимое сырьё у отрасли «необходимого труда»;
идёте на завод, и просите специалистов изготовить, либо всё делаете сами под их присмотром (так или иначе им надо будет отплатить).

Таким подходом, если у вас есть необходимые данные, можно хоть автомобили разных лет делать. Может, вы соберёте целое сообщество, которое будет этим заниматься. Никто не запретит.

DMGarikk 24 октября 2018 в 12:13

0

> Назовите, что обеспечивается работой данного человека — вот вам и оценка.

ну допустим для программиста обеспечивается работоспособность включения лампочек в ЖД вагонах например… сейчас там все электронное и умное… раньше это на релюшках и тумблерах было сделано (и вообще проще в некотором роде), а для маляра… обеспечивается мммм… несущественная защита от коррозии подвагонного оборудования?
как вам оценка? заслуживает программист столько же благ сколько получает сейчас (сильно выше среднего по ЗП среди не-меджерского персонала)? а маляр который вроде чето полезное делает, не особо тяжелое но очень вредное для своего здоровья? (сейчас он получает копейки и молоко)… хотя вагоны красили всегда, и без процессоров в каждом выключателе вагоны ездили

> Учитывая, что в предлагаемом обществе все люди трудятся добровольно, нет никакой нужды грызть друг друга за выживание

Ну сейчас как минимум в мегаполисах стран золотого миллиарда нет нужны грызть друг-друга за выживание, а работа на которой надо работать чтобы только поесть доступна всем вокруг (многим только лень и на ней работать)… но люди всеравно хотят повышения достатка и уровня жизни

rebus_x 24 октября 2018 в 13:04

–1

В данном случае с программистом и маляром я бы сказал, что в идеале в скором времени их задачи стали бы ненужными, потому что технологии совершенствуются.
У программиста — потому что глупо управлять светом через чипы, вряд ли очень уж надо программировать освещение. Лично я бы вернул простой выключатель, ну и регулятор светимости.
У маляра — потому что вредная работа, и потому что, ящетаю, задача окрашивать вагон снизу роботизируема; к тому же можно и конструкцию вагона переделать.
Но до тех пор пока задачи есть — оба (ведь они сами взялись за них?) за их выполнение получают право брать что захотят. Если хочется именно выяснять, чья работа важнее, и тому у кого важнее давать больше — то вопрос «как это делать» надо выносить на обсуждение. У меня сейчас нет предложений, к тому же это уже подробности, а я про общий подход.

DMGarikk 24 октября 2018 в 13:16

+1

> потому что глупо управлять светом через чипы, вряд ли очень уж надо программировать освещение.

ну вот видите, вы уже делаете выводы о нужности работы по поверхностным ощущениям

освещение программируется в зависимости от внешнего освещения, времени суток и от уровня заряда аккумуляторов. а реализация всего этого на чипах — это упрощение в эксплуатации — оно гораздо реже ломается

> к тому же можно и конструкцию вагона переделать.

срок эксплуатации вагона 50 лет, и переделывать его под покраску никто не будет ради только исключения маляров
с роботизированием — ок, возможно, хотя это ещё более нереально :)

> а я про общий подход.

а я вот указываю на откровенные «косяки» такого подхода, которые ооочень сложно преодолимы и при этом совершенно несущественны при наличии денег

rebus_x 24 октября 2018 в 13:34

–1

> освещение программируется в зависимости от внешнего освещения, времени суток и от уровня заряда аккумуляторов.

При изменении первых 2х переменных человеку не влом будет включить свет самому, если он ему правда нужен.
А третье вообще не должно влиять, я хочу книгу почитать а тут аккумуляторы садятся, понимаешь.

> а реализация всего этого на чипах — это упрощение в эксплуатации — оно гораздо реже ломается

М. То-то вы про программиста вопросы задаёте, который исходя из вопроса должен сопровождать это дело. И попробуй почини, если прямо в дороге таки сломается.

> срок эксплуатации вагона 50 лет, и переделывать его под покраску никто не будет

В обществе будущего другие правила, вкратце — «как можно лучше», вместо «как можно выгоднее».

> а я вот указываю на откровенные «косяки» такого подхода, которые ооочень сложно преодолимы

Преодолимы, просто вы предлагаете мне решать за многих, а также предлагаете решать ещё не существующую проблему. Я не считаю, что в приведённом вами примере кто-то скажет, что его труд оценён несправедливо, потому как есть достаточно условий для свободного решения, что делать и как жить, что за свой труд получить. А вот как только скажет, тогда и надо думать — и не только мне одному)

DMGarikk 24 октября 2018 в 14:06

0

> При изменении первых 2х переменных человеку не влом будет включить свет самому,

В вагоне помимо вашего купе есть еще общий коридор, служебные помещения и внешние сигнальные фонари

> В обществе будущего другие правила, вкратце — «как можно лучше», вместо «как можно выгоднее».

вы настолько уверены что не будет проблем с достаточностью ресурсов?

> Я не считаю, что в приведённом вами примере кто-то скажет, что его труд оценён

ну почему? вы думаете поменяется кардинально мышление людей? попытки это сделать уже были в прошлом веке

rebus_x 24 октября 2018 в 14:27

0

> В вагоне помимо вашего купе есть еще общий коридор, служебные помещения и внешние сигнальные фонари

Да, но меня это не смущает. Есть проводники, более того — они есть на каждый вагон. Внешние огни вообще можно от машиниста включать.
Кстати, датчики освещённости, емнип, изобрели ещё задолго до электроники, но тут я могу ошибаться.

> вы настолько уверены что не будет проблем с достаточностью ресурсов?

Да.

> вы думаете поменяется кардинально мышление людей?

Это предварительное условие. Это вариант общества в мире, который полностью готов становиться космической цивилизацией. Для этого перемены в сознании людей необходимы.

DMGarikk 24 октября 2018 в 15:25

+1

> Это предварительное условие. Это вариант общества в мире, который полностью готов становиться космической цивилизацией.

Ну тогда это рассуждения в стиле «ах еслибы», а мир не готов, и чтото сомневаюсь что в ближайшие лет 200 будет готов

popov654 25 октября 2018 в 02:06

–1

Вы всё-таки передёргиваете немного, имхо. Пример с лампочками крайне неудачен. Нельзя сравнивать программиста и маляра по тому, насколько критичны эти системы для поезда.

1. Программа для выключателя может быть крайне простой
2. Выключатель может быть без особых проблем заменён на механический тумблер, как вы сказали выше — а значит, он как раз менее ценен, чем антикоррозийное покрытие
3. Самое главное — программа пишется один раз и легко тиражируется на любое число устройств. А труд маляра — не тиражируется никак
4. У маляра вредность для здоровья больше

Так что имхо, оплата должна быть примерно равной.

Сейчас ведь как — оценивают трудозатраты и квалификацию (деньги и время, вложенные в обучение специалиста). Причём квалификацию оценивают сильно выше трудозатрат. Имхо, даже чересчур высоко. Тяжёлая физическая работа тоже должна оплачиваться хорошо.

DMGarikk 25 октября 2018 в 08:45

0

> Так что имхо, оплата должна быть примерно равной.

Имхо будет такой что если маляру будут платить как программисту, то вагоны реально будут красить роботы, а цена билета в плацкарт будет как в бизнес-класс самолета
А вот если программисту будут платить как маляру, программисты все разбегутся

popov654 25 октября 2018 в 01:58

–1

> не сможет договорится со всеми этими тысячами людей о бартере

Вы преувеличиваете. Всё, что нужно для такого перелёта — самолёт, пилот, топливо, два диспетчера в конечных точках маршрута и две взлётно-посадочные полосы (даже аэропорт по сути опционален).

Тут еле-еле десяток человек наберётся. А вы про тысячи пишете)

UPD: да, нужны ещё диспетчеры по пути, чтобы в небе самолёт с другим не столкнулся и все долетели в целости. Ну ещё пара десятков человек, которые в любом случае обслуживают это дело массово и круглосуточно.

DMGarikk 25 октября 2018 в 08:43

0

А службы которые делают ТО самолета? Доставляют топливо? Запчасти? Антиобледенение? Сеть приводных маяков? и аэропорт это не 2 человека, вахтер и диспетчер, это огромный комплекс, только сейчас с ними авиакомпании договариваются, а вы как пассажир сбор платите… а тут придется самостоятельно… ну или «всем все доступно, а как люди сами решат как-нибудь, это сложно» (с)

popov654 25 октября 2018 в 15:33

0

> и аэропорт это не 2 человека, вахтер и диспетчер, это огромный комплекс

Чтобы просто тупо долететь из точки в точку — весь этот комплекс не нужен. Это скорее всё дань удобству и безопасности (досмотр, таможня и так далее).

Если вы вспомните самое начало авиастроения, когда ещё самолёты не были такими большими, людей летало мало и редко — то поймёте, что можно и без этого. Разве что, современные самолёты стали сильно больше и сложнее своих предшественников, особенно если брать дальнемагистральные.

> А службы которые делают ТО самолета

Им можно платить на постоянной основе (как особо важной службе), а не на сдельной, там выше @rebus_x писал об этом.

DMGarikk 25 октября 2018 в 16:04

0

> особенно если брать дальнемагистральные.

а я в примере специально очень большое расстояние выбрал



Alexeyslav 29 октября 2018 в 12:17

0

Пилоту надоело, он ушёл пообедать(ой, получать причитающиеся услуги) и будет через 3 дня…

popov654 31 октября 2018 в 09:59

0

Надо держать несколько человек в горячем резерве, как в парикмахерских :)



Wesha 24 октября 2018 в 00:06

0

Сдаётся мне, всё уже украдено придумано до Вас.

> — Лады. Я, конечно, мог бы тебе сказать, что такое об, но сделаю лучше. Я тебе покажу, что это такое. — Джефф сполз с табурета и переместился к задней двери. — Не знаю даже, с чего это я должен стараться и учить уму-разуму человека в униформе. Так разве что, от скуки. Иди за мной.
> Гаррисон покорно последовал за ним во двор. Джефф Байнс показал ему груду ящиков:
> — Консервы. — Потом показал на склад рядом. — Вскрой ящики и сложи консервы в складе. Тару сложи во дворе. Хочешь — делай, хочешь — нет. На то и свобода, не так ли?
> Он потопал обратно в лавку.
> Предоставленный самому себе, Гаррисон почесал уши и задумался. Дело пахло каким-то розыгрышем. Кандидата Гаррисона искушали сдать экзамен на диплом сопляка. Но, с другой стороны, стоило посмотреть, в чем дело, и перенять трюк. И вообще, риск — благородное дело. Поэтому он и сделал все, как было сказано.
> Через двадцать минут энергичного труда он вернулся в лавку.
> — Так вот, — объяснил Бэйнс, — ты сделал что-то для меня. Это значит, что ты на меня имеешь об. За то, что ты для меня сделал, я тебя благодарить не буду. Это ни к чему. Все, что я должен сделать, это погасить об.
> — Об?
> — Обязательство. Я тебе обязан. Но зачем тратиться на длинное слово, когда и короткого хватает? Обязательство — это об. Я с ним обхожусь таким путем: здесь через дом живет Сет Вортертон. Он мне должен с полдюжины обов. Так что я погашу об тебе и дам возможность один об погасить Сету следующим образом: пошлю тебя к нему, чтобы он тебя покормил. — Джефф нацарапал что-то на клочке бумаги. — Отдашь это Вортертону.
> Гаррисон уставился на записку. Очень небрежным почерком там было написано: «Накорми этого обормота».
> Несколько ошарашенный, он выбрался за порог [...] внимание Гаррисона привлекла витрина через дом от лавки Джеффа. Она ломилась от яств, а над ней висела вывеска из двух слов: «Харчевня Сета». [...] худенький человечек в белой куртке поставил перед ним тарелку с жаренным цыпленком и гарниром из трех сортов неизвестных Гаррисону овощей.
> Гаррисон отдал ему записку. Взглянув на нее, официант крикнул кому-то, почти невидимому в струях пара за стойкой:
> — Ты погасил Джеффру еще один.
> И разорвал бумажку в мелкие клочки.

rebus_x 24 октября 2018 в 07:36

0

Спасибо, я учту.

Akon32 23 октября 2018 в 16:36

0

Напоминает бартерный обмен.



Alexeyslav 29 октября 2018 в 11:44

0

Если бы люди умели договариваться, в мире уже небыло бы ни одной серьёзной войны.

rebus_x 29 октября 2018 в 11:54

0

Не угадали с предметом обсуждения. С соседями по лестничной клетке вы войну не ведёте, со встречными на улице тоже. А здесь идёт речь именно о таком уровне взаимодействия, человек-человек, вопрос решения международных противоречий и устранения заказчиков войн здесь не рассматривается.



Alexeyslav 29 октября 2018 в 12:28

+1

На самом деле, все мы ведём свои войны, они не такие кровавые но всё же мы постоянно сражаемся с людьми нас окружающими. Будь то соседи или еще кто. Просто временное перемирие, временные альянсы, всё может изменится в любой момент и добрый сосед может стать вашим личным врагом просто неаккуратно вынеся мусор, или бабка кормящая кошек в подъезде. Всё очень шатко… с этих мелочей, шаткости и принципиальной неустойчивости человеческой психики в куле со стадным инстинктом и начинаются войны.

popov654 31 октября 2018 в 10:02

0

Тут надо всё же разделить понятие ваших чувств к человеку (ненависть к соседу) и реальных действий. Я таки могу кого-то ненавидеть, но при этом не гадить ему при каждом удобном случае, оставляя эту ненависть внутри себя.

Так что всё же @rebus_x прав, война с соседями или кем-то там ещё — скорее редкое исключение и форс-мажор.

Не говоря уже о том, что большинство людей и сами ведут себя осторожно, чтобы не стать таким объектом ненависти, и в случае чего даже спрашивают, всё ли ок, или что-то не нравится.

Alexeyslav 31 октября 2018 в 10:33

0

Это называется сдерживающий фактор. Вы сдерживаете свою ненависть только потому что понимаете что будет ответка. Но это всеравно война… как только пропадёт сдерживающий фактор будет очень трудно удержаться и не нагадить. Тут вступает в действие второй фактор который называют цивилизованностью, но он возник не на пустом месте.

popov654 31 октября 2018 в 11:13

0

> как только пропадёт сдерживающий фактор

С чего он вдруг пропадёт? Имхо, ответка всегда возможна. Но это не только из-за того, что люди не хотят ответку, на самом деле.
Есть ещё другие факторы:
1) Ухудшение настроения из-за мыслей об объекте ненависти — проще игнорировать и стараться не вспоминать
2) Страх получить эту самую ответку, даже если шансы на неё малы, отсюда стресс
3) Обдумывание методов, как нагадить — отнимает время и в перспективе деструктивно действует на личность

И можно ещё много придумать причин, почему люди стараются не раздувать конфликты, а гасить их.

Alexeyslav 1 ноября 2018 в 10:39

0

Страх получить ответку возникает только тогда когда ответка идёт от более сильного. Когда, к примеру, у тебя есть оружие ты не боишься получить ответку от прохожих у которых нет оружия, но боишься получить ответку от полиции или группы вооруженных лиц, что сдерживает владельцев оружия от массовых отстрелов мирных жителей. Посмотрите что происходит в зонах военных конфликтов, как только меняется расклад сил — уходит полиция из региона так сразу вылазят мародёры. Причем это всё те же жители которые жили там 20-30 лет и мародёрством не занимались т.к. был сдерживающий фактор в виде полиции.
Есть люди, для которых «обдумывание как бы нагадить» доставляет истинное удовольствие, их личность уже «разрушена» по мнению «нормальных» людей.
Причины придумывать можно, но в реальности есть и причины почему люди стараются РАЗДУВАТЬ конфликты и получают от этого удовольствие.

popov654 1 ноября 2018 в 15:45

0

> Есть люди, для которых «обдумывание как бы нагадить» доставляет истинное удовольствие, их личность уже «разрушена» по мнению «нормальных» людей.

Согласен, наверное, есть и такие.

AmOralist 29 октября 2018 в 16:58

0

> С соседями по лестничной клетке вы войну не ведёте, со встречными на улице тоже.

Эх, везёт вам с соседями по лестничным клеткам и с районами проживания и посещения…

popov654 23 октября 2018 в 17:42

0

Мне очень нравится ваша модель. Хотя эти самые «заслуги» (которые зачисляются на будущее") очень похожи на нынешние деньги, тут нет большой разницы…

popov654 18 октября 2018 в 10:02

+1

Какой смысл? Сейчас английский все знают в совершенстве (кое-кто и получше русского, хех). В школах с 1-2 класса изучается, и потом ещё в вузе. А те, кто изучает IT — вообще в этот язык погружены постоянно — чтение технической литературы, написание самого кода, обсуждения на иностранных форумах. Смысл заниматься какими-то переводами, если после этих переводов всё станет МЕНЕЕ понятно? Английский язык как раз тем и хорош — краткостью и однозначностью. Его во многом именно поэтому выбрали стандартом де-факто для IT -отрасли, а не потому, что «злые штаты так захотели».

VolCh 18 октября 2018 в 10:38

0

Думаю, что основным фактором выбора английского стандартом де-факто было просто лидирующее положении США в «информатике», а не какие-то объективные преимущества самого английского. Грубо говоря, если бы в ВМВ победила Германия, то писали бы «als dan anders», а не «if then else» (ругл транслейт, если что)

popov654 18 октября 2018 в 11:30

0

И всё же языки романской группы в целом короче и однозначнее, чем тот же русский и прочие славянские :) Просто английский и немецкий оба туда относятся, вроде. Ну и плюс английский на тот момент (когда информатика зародилась как наука) уже был в статусе международного и лидирующего языка.

Но вообще да, наверное фактор лидерства США тоже роль сыграл.

staticlab 18 октября 2018 в 12:11

0

Нет, английский и немецкий относятся к германской группе. И в немецком слова не самые короткие. Что же касается однозначности, то в английском у слов может быть весьма много значений. Например, как у слова set.

popov654 18 октября 2018 в 13:10

0

Ну в математике и IT есть одно общепринятое значение этого слова в форме существительного — «множество». А в целом — посмотрел, и правда очень много значений, я о большей части даже не подозревал)

staticlab 18 октября 2018 в 13:16

+1

А вот не только, в IT это может быть ещё как минимум глагол "установить":setValue(5)

popov654 23 октября 2018 в 03:34

0

Так я же сказал — «в форме существительного» :)

Wesha 19 октября 2018 в 03:09

0

> И всё же языки романской группы в целом короче и однозначнее, чем тот же русский

— Да? — удивился шеф. Ну так бы сразу и сказал, что ты поймал Hottentottenbegriffsstutzigstottternmutterbeutelrattenbedecktlattengittermörder.

staticlab 19 октября 2018 в 08:17

0

Там есть и по поводу однозначности:

> Так, например, одно и то же словечко означает "вы", означает "она", означает "ее", означает "оно", означает "они" и означает "их". Представьте же себе, как нищенски беден

> должен быть язык, где одно слово исполняет обязанности шести, будучи при том хилым, хрупким созданием из трех букв. А главное, представьте себе положение слушателей, которые не знают, о чем с ними говорят. Вот почему, когда я слышу обращенное ко мне словечко sie, я готов растерзать говорящего, особенно если мы незнакомы.
>
> Например, слово Schlag и слово Zug. В словаре "Шлягу" отведено три четверти колонки, а "Цугу" – все полторы. "Шляг" означает-удар, ушиб, розга, рана, сотрясение, грохот, размер, ритм, такт, темп, род, сорт, вид, штрек, апоплексия, огород, поле, валка леса, пенье, щебетанье, чириканье, щекот соловья. Но таковы только собственные, прямые значения этого слова, его постоянные, строго прикрепленные значения; дело в том, что есть тысячи способов открепить его, и тогда оно взовьется к небу и пойдет парить на крыльях, так что и не удержишь. Достаточно прикрепить к его хвосту то или иное слово-довесок, и оно примет любое угодное вам значение. Начните хотя бы с Schlagader – "артерия", а потом пройдитесь по всему алфавиту – вплоть до Schlagwasser – "трюмная вода" и включая Schlagmutter – "теща"".
>
> То же самое и "Цуг". Его прямое значение – тяга, сквозняк, движение, шествие, колонна, вереница, стадо, стая, упряжка, обоз, поезд, караван, черта, линия, штрих, шахматный ход, закидывание сети, дыхание, предсмертная судорога, агония, орудийная нарезка, шнурок. Когда к слову "Цуг" присоединены его законные довески, оно может значить решительно все на свете, – значения, которое ему несвойственно, ученым еще не удалось открыть.

popov654 23 октября 2018 в 03:39

0

Ну это же нормально… Взять вот русский. У нас «ключ» — и родник, и от замка, и музыкальный, и в значении «стиль», и возможно, ещё куча всего… И если первое, третье и четвёртое ещё более-менее можно свести к одному подсознательно, то вот со вторым уже проблема :)

popov654 23 октября 2018 в 03:35

0

Ну да, этот принцип в немецком я знаю. Из того анекдота как раз)
Но в среднем слова в немецком всё же короче, чем в русском (или почти такой же длины).
Если судить по самым распространённым разговорным фразам.

Хотя может мне кажется.

rebus_x 18 октября 2018 в 13:49

+1

Я подробно описал, какой смысл — читайте внимательней, может несколько раз прочтите (ничего постыдного в этом нет).

> Сейчас английский все знают в совершенстве (кое-кто и получше русского, хех)

То-то я и смотрю, что половина (Русского) ютуба, например, не может без переводов.

> А те, кто изучает IT — вообще в этот язык погружены постоянно — чтение технической литературы, написание самого кода, обсуждения на иностранных форумах. Смысл заниматься какими-то переводами, если после этих переводов всё станет МЕНЕЕ понятно?

Объясняю смысл на примере. Берёте фильм «Матрица. Революция» с исходной Английской звуковой дорожкой, мотаете на сцену с Архитектором и пытаетесь объяснить любому другому человеку, что он говорит. Только не общими словами, а поточнее. Чтобы было интересно, можно попробовать объяснить вообще происходящее по сюжету.

Потом берёте Кернигана и Ричи, про язык Си. Открываете на случайной странице, и делаете то же самое.

Потом, для интереса, берёте что-нибудь по матану, и повторяете процедуру.

Так вот смысл: избавиться от птичьего языка. Если вы что-то хотите сказать — научитесь выражать это без наукообразного нагромождения словечек, значение и суть которых очень часто сами говорящие разъяснить не могут.

> после этих переводов всё станет МЕНЕЕ понятно

Только на первое время, а вот станет ли понятнее и проще в дальнейшем — зависит исключительно от качества проделанной работы по переводу и переработке базы знаний. И увы, в том что с переводчиками у нас невероятная беда, я вынужден согласиться целиком и полностью.

Но я ведь не говорил, что я был бы готов поручать эту работу им… Я говорил, что эта работа, в моём видении, является одной из целей/задач.

> Английский язык как раз тем и хорош — краткостью и однозначностью

Для любого человека лучше всего тот язык, который он впитал первым в своей жизни, проще говоря родной. Ибо этот язык определил режим работы его мозга на всю жизнь.
Да, языки могут быть более совершенными и менее совершенными. Но и совершенствовать то, что есть, тоже можно — и на мой взгляд, обходясь при этом без заимствований, по крайней мере в случае Русского языка уж точно.

> Его во многом именно поэтому выбрали стандартом де-факто для IT-отрасли, а не потому, что «злые штаты так захотели»

Английский язык стал языком ИТ-отрасли просто потому, что основная столица, которая определяет эту отрасль — англоязычна. И всё. «Выбирать» язык отрасли никто и не собирался, кстати — это просто глупо, если ты этой отраслью сейчас покоришь весь мир, и весь мир будет от тебя зависеть. Что касается «злые штаты захотели» — зачем самим штатам чего-то хотеть? У государства США есть владельцы и хозяева, и было удобно воспользоваться положением и под шумок (BOB-BMB) обогатиться на войне, загрести под США побольше власти и влияния, свезти туда побольше мозгов.

rebus_x 18 октября 2018 в 17:47

0

Про «Матрицу» ошибся, имел в виду вторую часть, «Перезагрузка».

popov654 23 октября 2018 в 03:47

0

> То-то я и смотрю, что половина (Русского) ютуба, например, не может без переводов.

Ну их проблемы, значит, в школе не учились, а фигнёй страдали. У меня с английским всё в порядке. Как и у стримеров, которых я смотрю :)

> Так вот смысл: избавиться от птичьего языка. Если вы что-то хотите сказать — научитесь выражать это без наукообразного нагромождения словечек, значение и суть которых очень часто сами говорящие разъяснить не могут.

Эм, что?.. Вообще вас не понял здесь. Я думал, вы подведёте к тому, что «вы знаете лишь свою узкую предметную область (IT), а в целом английский, скорее всего, знаете не очень, а точнее — в меру частоты и интенсивности использования лексики по конкретной тематике». С этим я бы согласился. Но «птичий язык»-то при чём тут?

Люди общаются на том языке, на котором мыслят, и на котором им удобно понимать друг друга. В музыке, в стихах, в программировании и IT, да даже в тех же диалогах в играх и кино — для меня это английский. Я чаще смотрю игры в английской озвучке с русскими субтитрами, и для меня это интереснее и в чём-то приятнее, чем смотреть в русском дубляже, даже если дубляж прекрасный.

> значение и суть которых очень часто сами говорящие разъяснить не могут

Так я-то могу, по крайней мере в IT точно. Только не вижу смысла (за редкими исключениями). Понятнее не станет от такого разъяснения.

> зависит исключительно от качества проделанной работы по переводу и переработке базы знаний

У вас и правда получается неплохо и понятно переводить. Но таких, как вы — мало. А спрос на сам такой перевод невелик…

> Для любого человека лучше всего тот язык, который он впитал первым в своей жизни, проще говоря родной. Ибо этот язык определил режим работы его мозга на всю жизнь.

Ну да, в повседневности я мыслю на русском. И всё же это не мешает мне в информатике мыслить на английском (по крайней мере 30-60 процентов времени).

> У государства США есть владельцы и хозяева

И кто же эти владельцы?) Какой-то теорией заговора попахивает…

rebus_x 23 октября 2018 в 14:21

0

> Но «птичий язык»-то при чём тут?

Имелось в виду, что все 3 упомянутых источника изобилуют непонятным языком для стороннего человека. Но при этом, по крайней мере могу сказать за первые 2 — по сути там излагаются не такие сложные вещи.

Если стоит задача что-либо подобное объяснить (допустим, обучающемуся), то это было бы намного проще сделать, если бы используемая терминология была на родном языке обучающегося, и желательно при этом была бы удачной.

Да, очень часто термины в ИТ относительно просты, например по сравнению с математикой. Просты и в отношении того, что стоит за ними — большинство технологий можно объяснить на пальцах. Но их много.

Сколько времени надо готовить программиста? Если нужен такой, чтобы сам думал и понимал что делает — то вот тебе сначала электричество, потом электроника и транзисторы, потом как работает процессор, потом ассемблер, потом как от ассемблера перейти к высокоуровневым языкам, потом языки, потом ООП, потом графика, потом интерфейсы (API) операционок, потом что уже наработано в области в которой он будет работать. Кто-то скажет, что надо бы ещё преподать информатику.
И на каждом этапе — специальный язык. И он является тормозом, с моей точки зрения.
Объясни проще — слушатель быстрее (и лучше) поймёт, меньше времени будет затрачено.

popov654 23 октября 2018 в 17:47

0

> Если нужен такой, чтобы сам думал и понимал что делает — то вот тебе сначала электричество, потом электроника и транзисторы, потом как работает процессор, потом ассемблер, потом как от ассемблера перейти к высокоуровневым языкам, потом языки, потом ООП, потом графика, потом интерфейсы (API) операционок, потом что уже наработано в области в которой он будет работать. Кто-то скажет, что надо бы ещё преподать информатику.

Да вы идеалист… Я уже 10-11 лет постоянно в обучении и самообучении, но даже я всех этих областей не знаю (минус электричество и транзисторы, минус ассемблер, графика — очень базовый уровень, API ОС — почти никак).

А что взять с людей, которые только 2-3 год обучения проходят, например студенты младших курсов?

Это утопия… :)

Да и потом, даже если пытаться всему этому научить — не факт, что всё будет хорошо понято.

rebus_x 23 октября 2018 в 19:01

0

Да, я такой. Убеждён, что если человек занимается компьютерами, но не понимает до конца как именно они работают — он в глубине души неспокоен :)

DjSapsan 24 сентября 2018 в 22:41

+1

Те же мысли посещают каждый день. Поэтому восхищаюсь редкими примерами настоящего искусства программирования (вот прям сейчас пишу этот комментарий с 1 ФПС, а процессор занят на 15%)

Am0ralist 25 сентября 2018 в 11:02

0

Какой-нибудь хром?

DjSapsan 25 сентября 2018 в 13:16

0

Да, и сейчас эта страница грузилась в два раза дольше чем ОС + 3 программы. Такое чувство, словно каждый комментарий весит по 100 Мб

DMGarikk 25 сентября 2018 в 13:21

0

ну всетаки скорость SSD и шин памяти компьютера и их задержки в сотни-тысячи раз быстрее чем канал интернета

VolCh 26 сентября 2018 в 10:34

0

Ну гигабитный канал и SSD точно не в тысячи раз отличаются.

Ogra 26 сентября 2018 в 13:04

0

По скорости — нет, а вот по задержкам запросто, особенно если жить вдалеке от ДЦ.

roma86 25 сентября 2018 в 10:25

0

AirDrop раньше глючил не по детски, но сейчас это лучший способ передачи файлов в экосистеме устройств Apple, на мой взгляд. Как и copy on iPhone paste on Mac.

Gizmich 25 сентября 2018 в 10:57

0

Статья прямо в душу. Мне кажется эти проблемы растут из желания программиста ускорить/облегчить свою работу, а не работу конечного пользователя продукта.

lxsmkv 25 сентября 2018 в 11:34

0

> облегчить **свою** работу

да, это часть причины.

«Три главных добродетели программиста — лень, нетерпение и самомнение.» © Larry Wall

Gordon01 25 сентября 2018 в 12:38

+1

У меня ЭТА страница открывается и работает на уровне 0.5 ФПС. 8 ГБ памяти

Am0ralist 25 сентября 2018 в 12:50

0

Браузер напишите? И в техподдержку (ниже — помощь, обратная связь)

Londoner 25 сентября 2018 в 13:15

0

У меня тоже на хроме под линуксом с 8 Гб памяти эта страничка тормозит не по детски. Этот текст набираю в текстовом редакторе, потом скопирую в форму. В форме же быстрее пяти символов в минуту набирать не получается.

Londoner 25 сентября 2018 в 12:50

0

Вопрос в тему — кто знает плагины для хрома или лисы, сами умеющие вырезать лишний джаваскрипт, но только ненужный, чтоб gmail, facebook, ebay и ещё 100500 разных тормозных порталов работали так же, но побыстрее? Я так понимаю, там нужно будет для каждого домена вести белый список нужных скриптов. Noscript и Yesscript — хорошая попытка, но всё ещё очень далеко от комфортного для работы идеала.

Alexeyslav 25 сентября 2018 в 17:10

0

На старом движке лисы это было возможно, теперь доступа плагинов к внутренностям нет и теперь это невозможно в принципе, только через внешний прокси-сервер который будет модифицировать страницы на лету, но тогда возникнут проблемы с HTTPS… есть только uMatrix, который способен блокировать скрипты и иные элементы по-доменно.

sumanai 26 сентября 2018 в 20:30

0

А разве uBlock не режет нормально?

Alexeyslav 26 сентября 2018 в 23:05

0

Насколько я понял, новый API не позволяет получать прямой доступ к DOM открытой страницы, а следовательно плагин не может разобрать на элементы и блокировать их отдельно. А вот перехватывать запросы на ресурсы и анализировать их тип/назначение может. Поскольку сайты сейчас все с подлисками, безболезненно вырезать какие-то скрипты отдельно невозможно, остаётся только блокировать целиком.
Есть консоль, через которую это всё можно было бы проделать но плагины не имеют к ней доступа.

popov654 18 октября 2018 в 10:09

0

HTTPS совершенно не помеха. Погуглите Resource Override (есть под все браузеры). Позволяет на лету удалять или модифицировать скрипты и стили, и даже подменять заголовки запросов в обе стороны. В том числе и решать проблемы с CORS.

Londoner 25 сентября 2018 в 13:06

0

И ещё. Недавно мне посчастливилось ~~взлпать~~ остановиться в отеле с ну очень медленным вайфаем. Так вот, вкладочка gmail работала, но вместо привычных 500 мегов кушала в районе тридцати. Подозреваю что gmail просто не успевал накачать кэш чтоб занять им полгига памяти.
А что если перехватывать системный вызов и говорить браузеру, что у нас не 16 гигов RAM, а всего два, веди себя прилично. Пробовал кто-нибудь устроить такую диету браузеру?

vadimzz 25 сентября 2018 в 17:14

0

> Google Play Services
Может он ими и не пользуется, но ими пользуется половина приложений в его телефоне :) Музыку, он видите ли, не покупает.

misha_shar53 26 сентября 2018 в 17:20

+1

Со статьей согласен на 200%. Программирование превратилось в слоеный даже не пирог, а башню. Весь этот стек программных слоев не отлажен и никогда отлажен не будет. Фундамент этого стека ОС состоит из тысяч программ на Си. Уже там неимоверное количество ошибок. Остальные слои не лучше. Попытка совершенствовать этого монстра ведет в никуда. И бросить нельзя и тащить невыносимо. Нужен другой подход. Начиная от нормального языка к архитектуре процессора, новому протоколу интернета, разработке нового функционального браузера и создании на его основе ОС.Программирование должно быть в развитой среде, где вопросы манипулирования данными, коммуникаций и GUI должны быть решены ОС. Просто выбросить все наработки конечно же нельзя. Поэтому придется какое то время поддерживать оба направления. Но от нынешнего положения дел необходимо уходить. И искать новые пути не в мире языков Си, архитектуры x86 и ОС UNIX.

PsyHaSTe 26 сентября 2018 в 18:38

0

Звучит неплохо, но никогда такого не будет сделано. По крайней мере не ближайшие 10 лет точно.

khim 26 сентября 2018 в 22:55

0

Такое не просто «будет сделано», такое «уже делается». Просто мы не знаем какой из десятков исследовательстких проектов приведёт к успеху, в конечном итоге.



Alexeyslav 26 сентября 2018 в 23:27

+1

Ни один.



misha_shar53 27 сентября 2018 в 15:37

0

Не обязательно. Все зависит от того, найдут ли они верную дорогу. Если будут переписывать LINUX, то точно не найдут.



garrydvaraza 27 сентября 2018 в 10:24

0

Михаил, как раз в языке си нет ничего плохого:) и там как раз проблем нет) Потому что язык си очень нишевый. Большинства аппаратных драйверов и сетевых и низкоуровневых библиотек написано на си. И они не весят по 10 Гб. И работают быстро.
И люди оторые на нем пишут обычно обычно не понаслышке знакомы с асмом и привыкли думать:)



misha_shar53 27 сентября 2018 в 15:45

0

Плохо то что на нем и его различных вариантах написаны огромные программные комплексы, а он для этого не подходит. С ростом объема кода на нем он становится совершенно неотлаживаемым. Кроме того он оказал негативное влияние на создателей других языков. Хотя не только он. Отличился в этом и Паскаль.



0xd34df00d 29 сентября 2018 в 18:02

0

В языке С плоха небезопасность. Возьмите лучше ATS для всех этих драйверов.

catharsis 26 сентября 2018 в 17:32

0

Ctrl-F "Android Go": 0



botyaslonim 28 сентября 2018 в 13:09

0

Я практически не могу читать тут комментарии, потому что Хабр никак не может сделать их пагинацию, и интерфейс адски тормозит даже на мощном компьютере. Вот такая живая иллюстрация к посту автора

staticlab 28 сентября 2018 в 13:22

0

Уже выяснили, что ФФ и Сафари не тормозят. Разработчики Хабра, наверное, должны переверстать комментарии, но проблема-то в Хроме, на который никто не жалуется. Наоборот, самый популярный браузер, как когда-то ИЕ (в котором эта страница тоже тормозит).



botyaslonim 28 сентября 2018 в 13:27

0

И правда, в Осле грузится быстро.
Но всё равно это проблема просто, а не проблема Хрома. Во-первых, потому что Хром — самый популярный браузер. Во-вторых, потому что огромная простыня комментариев дико неудобно. Создатели блог-платформ и форумов пришли к этой нехитрой мысли ещё лет 20 назад



youngmysteriouslight 1 октября 2018 в 14:27

+1

Для справки: в 12-й опере тоже всё тормозит.

Тема с пагинацией тут не такая очевидная. Пагинация хороша там, где объекты данных (темы, посты, сообщения) линейно упорядочены, обычно по времени. Форумы относятся к этой группе. Комментарии на хабре имеют древовидную структуру и мне не понятно, по какому признаку предлагается разбивать сообщения на страницы? По дате, как на форумах? По веткам обсуждения?
Вообще, известны примеры пагинации для подобных данных?



vintage 3 октября 2018 в 08:35

0

Любое дерево можно линейно упорядочить со всеми вытекающими.

Druu 3 октября 2018 в 11:23

+2

> Любое дерево можно линейно упорядочить со всеми вытекающими.

Линейно упорядочить можно абсолютно любую конечную структуру. Только если речь о выводе информации, то упорядочить надо не абы как, а так, чтобы человеку было удобно ее воспринимать.В в этом случае такого упорядочивания уже может не быть.



vintage 4 октября 2018 в 08:08

−2

Давайте не теоретизировать. Вы прямо сейчас читаете древовидные комментарии вертикально линейно упорядоченные. Иерархия показана лишь смещением. И то лишь до определённого уровня.

Druu 8 октября 2018 в 03:48

0

> Вы прямо сейчас читаете древовидные комментарии вертикально линейно упорядоченные.

Так любое представление любых данных по определению линейно упорядоченно вдоль любой выделенной координаты. Вы какую-то тавтологию высказываете.



vintage 8 октября 2018 в 08:12

0

Не придирайтесь к словам. Из контекста вполне ясно, что речь идёт о наглядной линейной упорядоченности.

Druu 9 октября 2018 в 06:25

+1

Так никакой наглядной линейной упорядоченности в постах на хабре и нет — посты выводятся иерархически, а не линейно, т.к. есть вторая семантически значимая координата. И там, где посты действительно становятся линейно упорядоченными (как вы сами заметили, при сильном смещении), наглядность сразу и исчезает.



vintage 9 октября 2018 в 06:31

0

Наличие второй координаты никак не лишает нас первой и возможности паджинации по ней.

VolCh 9 октября 2018 в 09:12

0

Тут скорее хронология вторая/вспомогательная координата, а не первая, для начала. А, главное, как вы представляете вообще такую паджинацию? 100 комментов на странице всего с переносом комментариев, например, 7-го уровня через страницы? 10 комментов первого уровня на страницу независимо от того сколько других уровней?

Druu 9 октября 2018 в 10:06

0

Хотелось бы увидеть, какой вы представляете эту паджинацию.



popov654 18 октября 2018 в 10:15

0

Лично я бы резал где придётся, чтобы страницы были равными (предсказуемой для пользователя длины). А остаток ветки выводил во всплывающем балуне с прокруткой, по требованию (сделать иконку для этого специальную у коммента, который перенесён на новую страницу, и

при этом не верхнего уровня).

P.S. Но у меня и так не тормозит (почти, немного подвисает при копировании текста), поэтому и так устраивает.

s-kozlov 3 октября 2018 в 18:39

0

> Комментарии на хабре имеют древовидную структуру и мне не понятно, по какому признаку предлагается разбивать сообщения на страницы?

Пагинация не единственный способ. Посмотрите на Ж.Ж.

sumanai 3 октября 2018 в 18:55

0

Ж.Ж ужасно неудобен, ни разу там больше 10 комментариев не прочёл из-за этой системы разворачивания.

s-kozlov 3 октября 2018 в 19:17

0

Не понимаю, о чем вы. ИМХО, интерфейс комментариев Ж.Ж далек от совершенства, но это значительно лучше, чем на Хабре. Хуже Хабра в этом плане только VK.

sumanai 3 октября 2018 в 21:06

0

Видимо, дело вкуса. Для меня ровно наоборот.

AmOralist 28 сентября 2018 в 13:28

0

> но проблема-то в Хроме, на который никто не жалуется. Наоборот, самый популярный браузер, как когда-то ИЕ (в котором эта страница тоже тормозит).

И, что характерно, теперь никто из программистов никакой беды не видит в том, что теперь место МС с ИЕ занял Гугл…

AmOralist 28 сентября 2018 в 13:22

0

И это проблема хромов и иже с ним.
На лисе — только долго грузится.

kovserg 28 сентября 2018 в 17:59

+1

Для таких простыней комментариев надо не «пагинацию» делать, а фильтры по дате, по возрасту сообщения, по дереву, по никам, по оценкам… А нарезание 2тыс комментариев на страницы просто еще сильнее осложнит поиск нужного места в этой простыне.

sumanai 28 сентября 2018 в 18:26

0

Сворачивание деревьев комментариев с запоминанием.

kovserg 29 сентября 2018 в 00:13

0

Типа такого

sumanai 29 сентября 2018 в 14:48

0

Не запоминает. И в идеале можно было бы не передавать клиенту свёрнутые деревья.

Wiemo 28 сентября 2018 в 20:12

0

Комментарии к этому посту грузились секунд 15

smplcd 1 октября 2018 в 11:06

+1

> Linux намеренно убивает случайные процессы. И всё же это самая популярная серверная ОС.

Звучит устрашающе. Да вот только это делается только в **очень** экстренных случаях, а именно — в ситуации критической нехватки свободной памяти. Говорить что GNU/Linux намерено так делает не очень не корректно, следовало бы отметить что это делает делает механизм ядра OOM Killer:

> OOM Killer — это способ ядра решить проблему, когда памяти недостаточно. Иногда процессы системы съедают ее всю, и системе надо кого-то убить, чтобы продолжить работу. Текущая реализация OOM Killer в Linux стремится выбрать наименее важный процесс. Он выбирает среди всех процессов, кроме init и kernel threads, самый негодный (badness).

catap.ru/blog/2009/05/03/about-memory-oom-killer

rurhwaqusk 1 октября 2018 в 11:06

0

И посмотрите, как недопрограммисты реагируют на указание им на их раздутое ПО.
https://forums.getpaint.net/topic/112055-paintnet-is-excellent-software-and-not-bloatware/
Они не только не собираются исправлять проблему, но даже не желают признавать проблему, и даже гордятся этой ситуацией и считают её правильной.

kovserg 9 октября 2018 в 11:44

0

Самое интересно что это очень распространенное поведение. Такие люди почему-то считают что если что-то не поддерживается, то этим не только нельзя пользоваться остальным, но и обязательно необходимо глумится над теми кто продолжает этим успешно пользоваться.
Наши любимые корпорации специально культивируют такое поведения, для внедрения "запланированного устаревания", дабы увеличить прибыль.

servdo 1 октября 2018 в 11:06

0

Та что говорить. У меня на 16гб и i7 эта страница из-за количества комментариев грузилась 5 минут, а данное текстовое после еле еле работает.

springimport 4 октября 2018 в 17:57

0

Приукрашивать не надо все же. Реальная загрузка страницы занимает 10-15 секунд. А текстовое поле работает медленно, да.
Ну и важно какой i7: первых поколений или 8700k на 5.0гц. Разные вещи все же.

servdo 4 октября 2018 в 18:07

0

Только ваш комментарий вынудил меня снова пройти этот ад. Грузилось 3 минуты до этого комментария и продолжает грузится прямо сейчас спуся еще 2 минуты. Текстовое поле все так же тормозит, вводимый текст появляется с задержкам и целыми кусками.

springimport 4 октября 2018 в 18:17

0

В лисе не тормозит, кстати. Разработчикам хабра надо оптимизировать комментарии, но кто же этим будет заниматься, как выше написано в статье.

popov654 18 октября 2018 в 09:15

0

У меня Chrome 45, текстовое поле тоже не тормозит. Может, виновата версия браузера или версия вашей ОС? У меня часть знакомых, например, жалуется на странные тормоза в Win10.

popov654 18 октября 2018 в 10:17

0

Имхо, не настолько важно. У меня Core i5 второго поколения с 4 ядрами по 3.3 GHz, но в целом ничего не тормозит (ну загрузку, вероятно, уже админы пофиксили, а вот скрипты за такой короткий срок — не уверен).

VladimirPashutin 1 октября 2018 в 11:06

0

К сожалению бизнес-цели работают в противоположном направлении к построению надёжных, долговечных решений. Улучшать положению в индустрии можно только в том случае, если будет подходящая задача и серьёзные инвестиции. Например если дальнейший рост аппаратных возможностей упрётся в технологические пределы.

Lordz 1 октября 2018 в 11:06

0

> «Мы накрываем дерьмо одеялами, чтобы не убирать его. „

The best! Не в бровь, а в глаз…

unsignedint 1 октября 2018 в 11:07

0

я конечно все понимаю, но андроид приложения памяти жрут немерянно. Тот же инстаграм или ютуб складывает видео в кэш, а потом его не чистит. Вот зачем мне видео которое я просмотрел месяц назад? Почему бы его не удалить? Кэш нужен ИМХО для стабилизации воспроизведения при плохой сети, но не 300 же мегабайт?
Помню первый андроид у меня был с 512 мб флеша, забилось за два года, потом купил с 2 гигами, еще быстрей кончился. Год назад решил больше не совершать такой ошибки, купил 8 с запасом, оказалось 4 гига занимает система, еще два то что нельзя удалить, и остается всего 2 гига под все остальное. При этом маркет отказывается устанавливать приложения если объем свободной памяти менее 512 мб(почему бы сразу 2 гига ограничение не сделать?) В итоге память кончилась за пару месяцев. Судя из статьи уже и 16 мало, так что видимо текущий смартфон будет у меня последним. Ибо зачем мне телефон которым нельзя пользоваться? Уже сейчас как и в старину приходится покупать билеты на электричку в автомате, потому что смартфон может уйти в ребут и начать оптимизировать приложения, а контролер ждать не будет.

khim 1 октября 2018 в 22:32

0

> Год назад решил больше не совершать такой ошибки, купил 8 с запасом

8 гигабайт. С запасом. В 2017 году. Это шутка такая или как? Извините, но телефоны с 16GiB были актуальны лет 8 назад. А в прошлом году речь могла идти либо о 32GiB модели (если уж вы хотели съэкономить), либо, что разумнее, о 64GiB модели. Мой телефон 3-летней давности с 32GiB держится вполне себе достойно пока (хотя и обновлений безопасности больше нет) — и это бюджетная модель! Если класс был чуть повыше — то там 16GiB было в принципе не предусмотрено было три года назад…

> я конечно все понимаю, но андроид приложения памяти жрут немерянно.

С одной стороны — об этом и статья. С другой — если пользователи выбирают огромные приложения и не готовы платить деньги за оптимизации — то откуда взяться оптимизированным приложениям???

> Ибо зачем мне телефон которым нельзя пользоваться? Уже сейчас как и в старину приходится покупать билеты на электричку в автомате, потому что смартфон может уйти в ребут и начать оптимизировать приложения, а контролер ждать не будет.

Уж не знаю что с ним вы делаете, что он у вас так непредсказуемо себя ведёт. У меня никаких «внезапных оптимизаций приложений» не наблюдается…

unsignedint 1 октября 2018 в 23:06

+1

16 гиг, эконом класс,, вы шутите? Там лежали и с 4 гигами, я взял еще не самый дешевый, 400 зл за него отбашлял. Вот сейчас пишу, а кори И7 отображает текст как ворд под виндой 3.11 в 90-х на проце с66 мгц и 100 мб ОЗУ. А там тоже помню документы были, по 200 страниц писали. На мобильнике пытался ответить, так одно слово в минуту удавалось набрать. Даже на старом 10 летнем ноуте с гигом оперативы и сильно более медленным процом и винда семерка работала и матлаб, а на гораздо более шустром мобильнике даже страничку весом 50 мб не открыть? Это же как надо недооптимизировать чтобы оно вот так работало? Как надо неоптимизировать код чтобы простейшее приложение а ля райнэйр, где надо сделать 4 хттп реквеста, получить 1 кб джисон и распарсить, безбожно тормозило на телефоне с двумя процами и 2 гигами оперативы?

Alexeyslav 2 октября 2018 в 08:49

0

Цена очень редко отражает качество покупаемой вещи. Высокая стоимость не говорит о высоком качестве и выдающихся характеристиках, это называется на современном языке «впарить».
ИМХО Дешёвые смартфоны с память 4/8Гб предназначены для 1-2 нетребовательных приложений и пару сотен фоток хранить.
Хотя сам отходил с 16Гб почти 5 лет и НИКОГДА не удалял фото с телефона, кроме изначально неудачных.

unsignedint 2 октября 2018 в 11:04

+2

ну я покупая новый телефон с памятью в четыре раза большей, ожидаю что хотя бы все приложения со старого туда влезут. А не так что что на старом стояло 50 приложений, а на новый только 20 влезает.

popov654 18 октября 2018 в 09:21

0

Так это не есть нормально. Ведь на телефонах с 1 ГГц, одним ядром и 512 Mb RAM можно было решать довольно широкий класс задач в своё время (сам такой мобилой до сих пор пользуюсь, в первые полтора-два года проблем не было, всё летало, и приложения и сайты). Да, потом стало подтормаживать, когда стало накапливаться много фоток, потому что неэффективный код Медиа Сканера и одно ядро. Но если почистить фотки, или ещё лучше сделать софт-ресет — всё снова начинает летать. Пока не открываешь какое-нибудь современное приложение…

popov654 18 октября 2018 в 09:18

0

Не надо утрировать. 7 лет назад (а не 8) почти не было телефонов с более чем 1 Гб встроенной памяти. У меня модель мидл сегмента 2011-ого года, там вообще 512 Мб.

Да и в 2012-2013 больше 2-4 гигабайт почти не предлагали. Это в последние годы какая-то лютая дичь происходит с раздутием объёмов памяти и размера приложений и самой ОС.

BigBeaver 18 октября 2018 в 10:24

+1

Вообще есть ощущение, что с 2012 года тенденци в мире софта стали хуже. При чем, во всем и вне зависимости от разработчиков (при Джобсе было лучше, гугл тормозит, линукс уже не тот, десятка провал). Ну то есть, в 2012 я бы написал оду о том, что «все говно, перелезайте на линукс», но в 2018 я не смог настроить новый Дебиан так, чтобы было удобно работать. Ни в какой из пользовательских сред. Юзабельность и функциональность стала ниже, а не выше. В условно новом Андроиде я не могу сбросить BT кэш, не сбратывая wifi и тд.

Я могу бесконечно ныть по этому поводу — абсолютно все системы и браузеры 2018 года ужасны. Просто к некоторым я немного привык, а некоторые наоборот ненавижу чуть больше.

popov654 18 октября 2018 в 11:35

0

Хотя есть приятные исключения из этих правил.

Например, недавно брал у друга телефон с новым андроидом, посёрфил немного инет, захотел удалить данные просмотра. Сначала посетовал было вслух, что нельзя удалить за выбранный период историю, как в десктопном Хроме (например, за последний час). Но потом с радостью заметил, что по крайней мере группу «Сегодня» можно отметить в один клик галочкой, и потом уже снять в ней лишнее, если надо.

Не знаю, как давно это появилось (у друга Андроид версии 8, у меня 2.3), но в моём штатном браузере так нельзя, а у него в Samsung Internet — можно. Прогресс налицо. :)
Вот если бы ещё приложения так не раздувались по памяти и потребляемому месту — вообще было бы неплохо.

P.S. Десятку не люблю, так и не смог к ней привыкнуть, сижу на XP и 7.

BigBeaver 18 октября 2018 в 14:26

0

Ну я не отрицаю, что какие-то фишки таки появляются.

sumanai 18 октября 2018 в 18:50

0

> 7 лет назад (а не 8) почти не было телефонов с более чем 1 Гб встроенной памяти.

Я с SGS2 (2011 год) на руках с 16ГБ памяти смотрю на это сообщение с недоумением.

popov654 23 октября 2018 в 03:48

0

Ого, не знал, что в нём так много. У вас там и процессор двухъядерный наверное?

sumanai 23 октября 2018 в 19:56

0

Других тогда не завезли.

DMGarikk 23 октября 2018 в 08:57

0

вы путаете оперативку и память «для фоточек»

popov654 23 октября 2018 в 17:48

0

Не, он не путает. Мы именно память «для фоточек» и обсуждали. У меня правда её 1 Гб, телефон 2011 года от SonyEricsson за 12 тысяч рублей по тем деньгам.

P.S. И то есть подозрение, что он про SD карту. Карту и я могу себе поставить, но не всё и не всегда на неё переносится, плюс страдает скорость доступа.

sumanai 23 октября 2018 в 19:57

0

> P.S. И то есть подозрение, что он про SD карту.

Карта памяти у меня на 64ГБ.

khim 24 октября 2018 в 00:30

0

Всё банальнее и проще. Вышедший в 2010м году Nexus S задал **стандартные** характеристики: 512MiB RAM, 16GB флеш. На эти объёмы и стали равняться. Хорошо это или плохо, но в сегмент «до 12 тысяч рублей» сегодня попадают телефоны с 64GB памяти, так что обсуждение 8GB как чего-то взятого «на вырост» просто удивляет, извините.

Хорошо ли это? Нет — об этом и статья. Но поскольку объёмы продолжают увеличиваться, то у вас есть выбор — брать нормальные телефоны с памятью уж как минимум 32GB (а лучше 64GB)… или мучится снося всё, что можно, пихая SD карту и так далее.

masterziv 1 октября 2018 в 11:07

0

Что характерно: данная страница загружается секунд 5-7.

dima_s98 1 октября 2018 в 11:07

0

Как иронично, что пост о том, как все плохо в современном it, повесил платформу, на которой был опубликован)

Red_Odysseus 1 октября 2018 в 11:07

0

Ещё чуть-чуть и эти инженеры задумают коммунизм.

apxl 1 октября 2018 в 11:07

0

Я думал, что я один такой остался. Полностью поддерживаю автора!

SergeyPerm 1 октября 2018 в 11:07

0

Примерно тот же вопрос я задаю, когда ПК/мобила подвисает и не может открыть выпадающее меню. А ведь это кусок пикселей на экране размером 200 на 300 и все. Неужели устройство не может закрасить кусок экрана пикселями без анимации. Это же элементарная операция когда пользователь делает запрос к устройству (нажал кнопку) и ожидает что бы на экране закрасились нужные пиксели и больше ничего. В такие моменты осознаешь, что что-то в этом мире не оптимально :)

sumanai 1 октября 2018 в 14:30

+1

> без анимации

Нет! Нельзя! Это же некрасиво!!!

khim 1 октября 2018 в 22:33

0

Не в бровь, а в глаз…

komis55 1 октября 2018 в 11:07

0

:) Форум то вроде для взрослых, а не дописав комментарий, полез свериться и мимоходом мелком подчеркнул, l.facebook.com/l.php?u=https%3A%2F%2Fwww.youtube.com%2Fwatch%3Fv%3DAqlY3UFqEdc&h=AT21I1FZkQAHRuztPdKHcm99yNHhjoNhr6jO3NGNNcuwS3YGuxnVwsc9R_9ILGa-d9ZyZU3Pzr_TPlgW6eRter4WDDVDuGa_Oa_dPCS9VbQhrxKpA.
Физика — это объём Математики. Не попутайте, кузнец математик, а бард физик.
3D языковая модель, скиньте у кого есть.

DGolubets 1 октября 2018 в 11:07

0

Автор просто ноет.
Конечно, **доля правды** есть, особенно в вебе. Но все логично объясняется.

Во-первых никто не отменял законы экономики, спрос и предложение. Если что-то перестает удовлетворять потребителей — обязательно появится конкурирующий продукт, соответствующий требованиям. Ну и если покупают неэффективные решения — значит устраивает за их цену.

Во-вторых рассуждения о «слоях дерьма» — это какая-то самоуверенность. Остановитесь на секунду и просто представьте себе что реально происходит при выполнении вашей простой программы. Готовы ли вы писать все сами? Шифрование, HTTP, или тот же терминал самостоятельно сделать? Операционную систему сами с нуля свалеете?

Докер и ко не были нужны 20 лет назад, потому что и требований как сейчас никто не предъявлял. А сейчас CI, масштабирование, отказоустойчивость — это уже норма. Что прикажете делать? Нанимать штат в 3 раза больше чтобы все это сделать без него? Ну и чем собственно он вам навредил? У докера оверхеда почти нет.

Еще сравнение «простых» текстовых редакторов с игрой…
Такие простые редакторы для которых каждый может написать расширение для синтаксического анализа языков программирования, подсветки, сворачивания кусков кода и даже рефакторинга.
И игра — конечный продукт, оптимизированный до умопомрачения под одну конкретную цель. Игра, которая работает быстро, потому что Nvidia специально под это разработала железку, которая рисует вам эти миллионы полигонов и при этом стоит как все остальное вместе взятое в компе.

stul5tul 1 октября 2018 в 11:07

0

> Вы бы купили машину с расходом 100 литров на 100 километров?

Экономия на топливе — это веяние последних десятилетий.
А так то:
spez-tech.com/tehnika/gruzovie-avto/ural/375-harakteristiki-istoriya-razrabotki-tsena.html



Сто не сто, но:

> Хотя заявленный расход был 48 литров на 100 километров, на самом деле он доходил до 70 литров.

И автомобили в США в те годы тоже не были особо экономичными.

Typeckuu 1 октября 2018 в 11:07

0

Я даже зарегистрировался из за этой статьи, я это давно заметил. Когда пишешь подобные вещи, большая часть людей смотрит на тебя как на идиота, понимают лишь единицы. Но проблема не в этом! Проблема в нашей стране в целом. Почему люди стали разочаровываться в софте, в оборудовании, потому что они стали разбираться в этом. Эта проблема была всегда где то там… А мы ее не считали проблемой, потому что не было знаний об этом. И вот единицы начали разбираться в этом, и что далее? Написать об этом статью, рассказать народу? Пробовал. Большая часть тебя просто не поймет, они считают это нормальным, часть начнет гнуть свою версию — так сказать понты, начнут давать советы, хотя у них руки из одного места и знаний 1%.
Я сам до сих пор использую Windows XP sp2, все работает как часы, все старое проверенное годами, и нет никаких проблем.
Большая часть людей не разбирается в тонкостях софта, они упорно сидят ждут, когда все протупит, чтобы нажать кнопку и опять ждать до желаемого результата, у них по 500 папок Яндекса, Mail.ru — и ведь они не знают, что Яндекс — это огромная троян-корпорация, которая наплодила за всю историю огромнейшую кучу троянов.
Так же у них браузеры — сэндвичи, некое подобие слоенного софта друг на друге, открывающего друг друга.
И когда что то читаешь на русских сайтах, понимаешь, что человек хочет поделиться знаниями, но их нет! Начало взято из научной статьи, а далее отсебятина ведущая в тупик, да понты, что я этим занимаюсь мол, а вы не умеете. Везде так, даже на хабре, где человек пишет, что это очень просто, но сам не может понять что он делает.
И если ты решил написать какую то софтину, то бесполезно изучать что то у нас на сайтах или у наших программистов, потому что:

1. Начало, красивое вступление, кусок вырезанный из научной статьи, переведенной на русский, та единственная часть, в которой разобрался автор. И он понял только этот кусок статьи.
2. Кусок кода, который работает, который протестировал сам автор. Далее куски кодов не относящихся к статье, текст отсебятины.
3. Всевозможные мечтания, что бы можно было создать и то что не под силу автору.
И получается не учебник и не статья, а своеобразный селфи статус.

Что остается пользователю? Учить английский и изучать основы из первоисточника, потому как у нас нет нигде нужной информации. Или изучать то, что есть и становиться таким же селфи — публикатором. У нас мало кто начнет что то делать сам, все переделывается из чужого кода. Итог — лучше чем было не будет! Так делает любой русский программист, потому как образование большего не дало. Заметил еще не маловажную вещь, очень часто висят базы данных государственных структур и почему то ссылаются на хакеров, почему на руки программистов не ссылаются?

И есть те единицы, которые что то сделали сами, обучались правильно, такие люди долго не задерживаются на весу, они востребованы и они не будут публиковать учебники или статьи, потому как делиться такими знаниями — финансовый ущерб. Так вот примерно все устроено.

Есть железо и софт работающие правильно, но у единиц, которые разбираются в этом сами! Остальное все кидают в необразованные массы и зарабатывают на этом. Нужно же как то зарабатывать =).

OruScUkG 2 октября 2018 в 15:57

0

Слушайте, а вы не замечали, что раньше и трава была зеленее, и небо голубее? Это называется «старые добрые времена». Не знаю, как вам, а мне ещё удалось застать людей, которые «до сих пор используют Windows 2000 SP4 и не собираюсь переходить на эти ваши XP, потому что у меня всё отлажено-настроено, а там одни свистоперделки, аляповатый интерфейс и вообще она аж вон столько на диске занимает…

И вообще — все эти разговоры про то, что „ведь по сути там ничего нового"… По сути, знаете ли, в информатике уж эдак годков 70 ничего нового, с тех пор как фон Нейман свою архитектуру придумал. Чем вы измеряете „степень новизны"? В каких единицах? Ну да, *в принципе* в каком-нибудь Microsoft Word'е или LibreOffice Writer'е мало чего *принципиально* нового по сравнению с… не знаю… ну, допустим, Vi. А то, что есть
новое (тот же WYSIWYG), всегда можно записать в свистоперделки.

Но вы подумайте при этом, как изменилось *количество* компьютеров за это время, и как изменилось влияние компьютеров на нашу жизнь за это время! Наверное, программы на первых компьютерах были гораздо более оптимизированы, нежели сейчас. Но эти компьютеры были достоянием узкого круга профессионалов и существовали в единичных экземплярах. И программисты выступали в роли этаких гуру, изрекающих мутные речения про „смотлную яму" и „серебряную пулю". Сейчас компьютеры, — увы, — превратились в бытовую технику, а программисты стали чем-то вроде автослесарей. Но кому от этого плохо? Разве что самим программистам?

Вспомните времена, когда для покупки билета на самолет или поезд вам было нужно ехать в кассу! А сейчас это можно сделать через интернет за 5 минут, и это уже воспринимается как норма. Конечно, интерфейс многих порталов чудовищен, и наверное, они могли бы работать в 2 или 5 раз быстрее. Но… „чудо не в том, что [медведь] танцует хорошо, а в том, что вообще танцует".

DrPass 2 октября 2018 в 16:04

+1

> Вспомните времена, когда для покупки билета на самолет или поезд вам было нужно ехать в кассу! А сейчас это можно сделать через интернет за 5 минут, и это уже воспринимается как норма. Конечно, интерфейс многих порталов чудовищен

Ок, моя бабушка в молодости не могла купить билет на самолет или поезд через Интернет. Но я это могу сделать уже *лет* десять, а первый интернет-магазин с доставкой еды на дом в моем городе (обычный областной центр) появился ещё 20 *лет* назад. Этого времени достаточно, чтобы сделать удобный и отзывчивый интерфейс?

OruScUkG 2 октября 2018 в 16:32

0

Ещё раз: давайте определимся, что означает «удобный» и «неудобный», как это измерять, и вообще — удобный или неудобный для кого? Всё *дело* в том, что как сама статья, так и множество комментариев выше наполнены рассуждениями о неком абстрактном удобстве или о соответствии неким стандартам, которым, по мнению авторов, ПО *почему-то* должно соответствовать.

Например, мои родители так в полной мере и не освоили интерфейс смартфона (я специально даже не заостряю здесь внимание на его ОС — Android или iOS), и для них он является неудобным. А какой-нибудь школьник начальных классов ориентируется в нем, как рыба в воде и даже не представляет себе, что может быть как-то по-другому. Удобен ли он для него? Безусловно.

Суть-то моих возражений к другому сводится. Понимаете, рассуждать о том, что «а вообще смартфон мог бы загружаться за 5 секунд» или «а вообще ОС могла бы помещаться на дискете» просто *бессмысленно*. Ну, может быть, и мог бы. Наверное. Ну и что? Это как рассуждать о том, что наверное, булки могли бы расти на деревьях. Теоретически, наверное, да — но ведь не растут же!

Всё дело в том, что этот мир изначально несовершенен (по крайней мере то, что сделано людьми). Поэтому и сделанное нами не оптимально. И эта неоптимальность суть двигатель прогресса. Обратите внимание — я говорю не «в большинстве случаев не оптимально», а вообще «не оптимально», то есть в принципе. Вдумайтесь: по генеральной совокупности признаков не оптимальна ни одна вещь! ОС, умещающаяся на дискету, может быть оптимальна по критерию объёма, но не оптимальна по критерию удобства или, скажем, универсальности. И т.д.

Как говорят в таких случаях, необходимо задать критерий оптимизации. И вот тут приходит некто и заявляет, что таким критерием, например, для смартфонов должно быть время загрузки. Зачем? Почему? В чем потребительская фишка телефона, грузящегося за 5 секунд? Я понимаю, тема статьи звучит "*Моё* разочарование в софте" — ну так пусть он (автор) хотя бы объяснит, для чего ему лично это нужно?

А так получается, что автор выбрал обвинительную линию, — презумпцию виновности, если угодно, — и излагает лишь те факты, что её подтверждают.

khim 2 октября 2018 в 17:25

0

> Теоретически, наверное, да — но ведь не растут же!

Не растут — потому что мы не умеем создать такое дерево. Вот в принципе не умеем.

> Понимаете, рассуждать о том, что «а вообще смартфон мог бы загружаться за 5 секунд» или «а вообще ОС могла бы помещаться на дискете» просто *бессмысленно*.

Почему бессмысленно? Если мы не будем об этом говорить — то этого никогда и не случится.

> И эта неоптимальность суть двигатель прогресса.

Только не в софте. Задумайтесь вот над чем: вы **не можете** откатить версию iOS или Android (на многих телефонах, на некоторых можно, хотя это всегда непросто). **Почему** это сделано? Почему у вас нет подобных ограничений, скажем, по отношению к автомагнитолам или каким-нибудь системам турбонаддува? Да потому что у внушительного количества пользователей выход новой версии возникает острое желания откатить всё назад! Вспомните историю продвижения Windows 10 хотя бы.

> Как говорят в таких случаях, необходимо задать критерий оптимизации.

А он очень прост: люди должны **хотеть** пользоваться новинками. Четверть века назад — это *происходило*. Когда MS DOS 4 оказалась хуже, чем MS DOS 3.30 (на которой люди сидели долгое время сидели после выхода MS DOS 4.x), то Microsoft приложил **массу** усилий для того, чтобы на MS DOS 5 люди **реально** сами захотели переходить. Люди добывали бета-версии Windows 3.1 или Turbo Pascal 7 и переходили на них за полгода *до* релиза. Люди покупали железо, когда вышла Windows 95, чтобы её можно было запустить.

Сегодня же выход новой версии MS Office, Skype или какого-нибудь GMail'а вызывает **массовые** вопрос с вопросом: «как теперь жить и нельзя ли вернуть старые версии, как-нибудь».

DMGarikk 2 октября 2018 в 17:39

0

> Почему у вас нет подобных ограничений, скажем, по отношению к автомагнитолам или каким-нибудь системам турбонаддува? Да потому что у внушительного количества пользователей выход новой версии возникает острое желания откатить всё назад!

Ооо, а вы выбрали пример который гораздо хуже чем откаты версий ОС
= «коробки» автомобиль практически невозможно проапгрейдить дальше его базовой комплектации, а в некоторых случаях выбора вообще нет… например купив какойнить мерседес за килограмм денег вы получается уродскую встроенную навигационную систему которая хуже яндекса и гуглкарт… и вы не можете её проапгрейдить или поменять… вообще… только купить китайскую магнитолу и слететь с гарантии (и есть ли китайские магнитолы на мерседес за кучу лямов?)
=с турбонаддувами ещё веселее, зачастую «штатных» способов апгрейда нет вообще, а нештатные официально имеют кучу бюрократии в регулирующих органах уже со стороны государства
==
ну а ответ на

> Почему это сделано?

прост

так дешевле, поддерживать годами несколько параллельных версий ОС, которые замечу бесплатны после их покупки… и например эпплу или микрософту придется содержать штат бесплатного саппорта для того чтобы тянуть старую версию «винды» или «макоса» которые вы купили лет 15 назад? им это элементарно невыгодно.
А зарабатывают они (пока) на продаже новых версий… но тут вы уже правильно описали мотивацию

popov654 18 октября 2018 в 09:29

0

> и например эпплу или микрософту придется содержать штат бесплатного саппорта

А мне не нужен саппорт от них — ни платный, ни бесплатный. К чему вообще это тут, мало кто вообще хоть раз в саппорт Гугла/Эппла писал: это долго, и с вероятностью 99 процентов ответа вообще не будет. Я даже в стиме 2 недели ждал ответа поддержки на официальном тикет-форуме, когда у меня аккаунт взломали…

OruScUkG 2 октября 2018 в 19:13

0

Вот, вот! Вы нас, можно сказать, значительно к пониманию сути дела продвинули. Только мы с вами к вопросу с разных сторон подходим:
1) в том, что если не ставить перед собой цели, то её и невозможно *достигнуть*, я с вами совершенно согласен. Просто не всякая цель достойна того, чтобы её перед собой ставить. Ведь ясно же, что для достижения цели нужно идти на компромиссы (делаем быстро, дешево, качественно — выбери любые 2 пункта). С чего автор материала решил, будто компромиссы, на которые нужно было бы пойти для достижения цели «загрузка смартфона за 5 сек», были бы для пользователя менее болезненными, чем те, на которые он идет сейчас? Ведь ни опросов, ни анализа, ни статистики какой-нибудь не приводилось. Хочу загрузку за 5 сек, и всё тут.
2) По поводу того, почему нельзя откатить версию ОС на смартфоне. Думаю, не стоит здесь искать какой-то заговор разработчиков. Просто так дешевле в смысле техподдержки. Смартфон воспринимается как *аксессуар*. Ведь основная масса потребителей смартфонов — это не гики, согласитесь. И они (потребители, в смысле, а не гики) давно уже забыли, что это по сути настоящий компьютер, только маленький (да и не помнили этого, наверное, никогда). Опять же — обратите внимание, что смартфоны стали массово популярны *именно после того*, как для рядового потребителя они перестали быть «компьютером» и стали «аксессуаром». Тут страшного ничего нет. Так было со всеми технически сложными товарами. Вот так же фанаты космонавтики ждут, когда SpaceX сделает полеты в космос доступными «для всех». Вы же не требуете от автолюбителя навыков автомеханика, а от владельца телевизора — навыков телемастера. ОС в смартфоне — такая же деталь, как кинескоп в ТВ, только не аппаратная, а программная. Поэтому её заменить (обновить) можно, а кинескоп — нет. Думаю, производители бытовой техники с удовольствием обновляли бы детали аппаратуры удаленно — да невозможно.
3) А вот теперь об обновлениях. Самый интересный пункт. Вот работаете вы в сфотописательской конторе, и кормитесь вы с продажи своего ПО. И разрабатываете вы его давно — не 1-2 года, а 10-20 лет. У вас офис, вы обросли штатом… Даже если вы разрабатываете какое-то реально специфическое ПО (типа CAD/CAM/CAE), рано или поздно вы подходите к пределу функциональности. И что тогда? Вы приходите к программистам, у которых семьи и кредиты, и говорите: «Друзья мои, наш продукт себя исчерпал. Контора закрывается, ищите удачи в *другом* месте?» Конечно же, вы так *не* сделаете (кто будет возражать — говорите, во-первых, за себя и, во-вторых, приводите проверяемые данные). Поэтому что вы делаете? Вы начинаете *придумывать* функционал. Вы начинаете *создавать потребности*. Иначе вы должны лишить людей их кука хлеба во имя идеи о соответствии разрабатываемого ПО неким идеалам. Уж даже не знаю, что менее гуманно — медленный смартфон и 100 сытых человек, или быстрый смартфон и 1 сытый и 99 голодных. Цифры, сами понимаете, условные, так что просьба не придираться. И ещё, кстати, не забывайте, что разогнать команду — это не значит потерять людей; это значит потерять компетенции. Если завтра они понадобятся, вы вынуждены будете начинать с нуля.

Alexeyslav 3 октября 2018 в 09:22

0

А потом проходит время люди привыкают и «как вообще можно было пользоваться старыми версиями?». У людей вызывает отторжение именно новизна, непривычность нового окружения и т.п.

khim 3 октября 2018 в 14:33

0

С одной стороны новизна отталкивает. С другой стороны полезные новые фичи принципаются «на ура». Да, человек такое существо, противоречивое. Но до определённого момента люди были готовы мириться с тем, что за новые плюшки им нужно мириться с новыми интерфейсом. А сейчас — подавляющее большинство нужно пинками заставлять переходить. Вплоть до того, что люди выискают .APK'шки старых версий на сомнительных сайтах, чтобы «новинки» не ставить.

Wesha 3 октября 2018 в 20:43

+1

> за новые плюшки им нужно мириться с новыми интерфейсом.

А почему "новые плюшки" неразрывно связаным с новым UI? Добавить ещё одну кнопку "НоваяПлюшка190" религия не позволяет — обязательно при этом надо все кнопки переместить на дуракий ribbon?

khim 3 октября 2018 в 21:08

0

Потому что очень часто для того, чтобы получить новые плюшки нужно провести глубокий рефакторинг. И он накладывается на идею по новому, «улучшенному» интерфейсу.

Да собственно всегда так было. Как и увеличение в размерах. Сравните как-нибудь Norton Commander 3, Norton Commander 5 и, когда-то всем известный, Volkov Commander. Я думаю вы порадуетесь… но при этом Semantec успешно продавал и Norton Commander 5 и Norton Commander 5.5 — хотя весь СНГ сидел на VC…

Разница только в том, что железо всё это безумие перестало маскировать.

Wesha 3 октября 2018 в 21:15

0

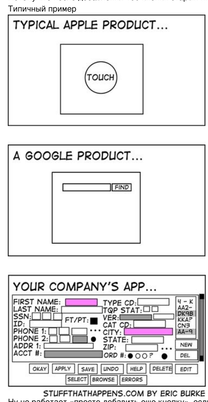
> рефакторинг. И он накладывается на идею по новому, «улучшенному» интерфейсу.

Вот поймать бы этих идеологов, и руки-то им поотрывать...

PsyHaSTe 4 октября 2018 в 11:59

+1

Потому что после добавления 190 кнопки старый интерфейс может развалиться. Типичный пример



Ну не работает «просто добавить еще кнопку», если делать удобное приложение.

SirEdvin 5 октября 2018 в 10:03

0

> Почему бессмысленно? Если мы не будем об этом говорить — то этого никогда и не случится.

Сколько вы готовы за это заплатить? А остальные потребители? А может на самом деле есть куда более критичные проблемы, например, проблема обновлений, которые перестают в какой-то момент выходить, может начнем с нее, а на загрузку можно забить. Или время загрузки для вас важнее?

> Только не в софте. Задумайтесь вот над чем: вы не можете откатить версию iOS или Android (на многих телефонах, на некоторых можно, хотя это всегда непросто). Почему это сделано?

Это катастрофически неправильный вопрос. Потому что, во-первых, вы можете, надо просто бекап делать перед обновлей. А во-вторых, правильный вопрос "**Почему это не было сделано?**", так как откат на обратную версию еще написать надо.

> А он очень прост: люди должны хотеть пользоваться новинками. Четверть века назад — это происходило. Когда MS DOS 4 оказалась хуже, чем MS DOS 3.30 (на которой люди сидели долгое время сидели после выхода MS DOS 4.x), то Microsoft приложил массу усилий для того, чтобы на MS DOS 5 люди реально сами захотели переходить. Люди добывали бета-версии Windows 3.1 или Turbo Pascal 7 и переходили на них за полгода до релиза. Люди покупали железо, когда вышла Windows 95, чтобы её можно было запустить.

Вам случайно не больше 30 лет? Мне кажется, у вас случился переход до консерватора. Люди все так же ставят себе беты, новые версии софта и покупают железо ради новинок. Просто сейчас не нужно обновлять железо, что бы вместо win 7 поставить win 10.

DrPass 5 октября 2018 в 21:43

+1

> Вам случайно не больше 30 лет? Мне кажется, у вас случился переход до консерватора. Люди все так же ставят себе беты, новые версии софта и покупают железо ради новинок.

Не так же. Новинки нынче скисли. Ранее пользователь получал новую версию раз в год, в два, в три, но это было событием. Новая версия — это означало множество новых фич, улучшение и упрощение работы на основе пользовательского опыта и т.д. Сейчас смотреть на обновления особо нечего. Что-то действительно интересное появляется редко, чаще это простой фейслифтинг UI, а самое неприятное — фейслифтинг происходит не на основе опыта пользователей, а просто по картинкам дизайнеров, чтобы «освежить». В результате нередко интерфейс становится хуже. Взять ту же упомянутую ниже звонилку Андроида. Раньше для приема звонка нужен был свайп влево, и это легко делалось большим пальцем, когда телефон находится в правой руке. В Андроид 8 теперь нужен свайп вверх. И о-па, сделать этот жест одной рукой уже намного сложее.

popov654 18 октября 2018 в 09:06

0

Это имхо зависит от кастомного UI от вендора телефона. У меня в моём SOny XPeria 2011-ого года требуется свайп вправо, чтобы ответить, и свайп влево, чтобы перевести звонок в беззвучный режим (то же самое можно сделать, погасив экран).

apxi 13 октября 2018 в 13:12

0

Сравните старый mail агент и последний после обновления. Сравните скайп старый и новый. Сравните офис 2003 и 2007. При этих обновлениях было очень много жалоб от обычных пользователей.

popov654 18 октября 2018 в 09:07

0

Новый Скайп (до версии 8) имхо всё же визуально красивее. Хотя функционально многое выпилили по сравнению с совсем старыми версиями :)

VolCh 13 октября 2018 в 14:00

+2

Тут не в консерватизме дело, по-моему, а в политике компаний. Теперь многие из них они могут позволить себе принудительно прекращать у пользователей работу старых версий софта. В старые времена, если ты купил себе софт, то пользоваться им мог неограниченно долго, обновлялся если только тебе нужны новые фичи новых версий. Сейчас когда очень много софта как бы локального является лишь клиентом к серверной части очень часто компании просто не оставляют тебе варианта «не обновляться», установленная версия софта просто перестаёт работать, если своевременно не обновился и приходится обновляться в надежде, что важных для тебя фич в новой версии хотя бы половину оставили, а новые хотя бы не сильно мешать будут.

DrPass 3 октября 2018 в 15:59

0

> Ещё раз: давайте определимся, что означает «удобный» и «неудобный», как это измерять, и вообще — удобный или неудобный для кого?

О, это комплексный показатель. Но к счастью, вполне объективный. Включает в себя и количество действий, которое нужно провести, чтобы получить результат, и отзывчивость интерфейса, и быстродействие, и интуитивную понятность для целевого пользователя.
Мир несовершенен, да. Но мы-то разработчики, наша профессия как раз и предназначена для

того, чтобы сделать его совершеннее, а не плодить всякую ерунду.
Вот, взять, к примеру, стандартное приложение звонилки в упомянутом вами Андроиде. С версии 4.x до версии 8 его функционал, в общем-то, практически не поменялся. Только внешний вид подтюнинговали. Но! Там добавилась какая-то функция, которая позволяет отправить смску, если вы не дозвонились. Фишка в том, что звонилка предлагает отправить эту смску в любом случае после звонка, даже если дозвонились. Как бы и не мешает пользоваться, но раздражает, примерно как заводской брак вроде облезлой краски на корпусе нового пылесоса. Эта мелочь хорошо отражает состояние дел в индустрии. Огромная компания-разработчик, армия девелоперов, армия тестеров, пять лет разработки, минимум новых фич, но даже при этом в релиз пошла корявая версия с видными невооруженным взглядом багами. Потому что всем плевать.

staticlab 3 октября 2018 в 16:13

0

Справедливости ради, это вполне может быть баг конкретного вендора телефона. Или у вас Google Pixel?

DrPass 4 октября 2018 в 22:57

0

Не Pixel, но тоже Андроид стоковый. У меня Moto Z.



san-smith 10 октября 2018 в 20:00

0

По поводу перехода на новую версию: в четвертом андроиде я мог выбрать, сохранять новые контакты на сим-карте, телефоне или в гугл-аккаунте, в седьмом (xiaomi) у меня забрали возможность сохранить в телефоне или симке. Вот такой прогресс наоборот.

Как Вы сказали, не то чтобы мешает, но раздражает.



Am0ralist 8 октября 2018 в 09:38

+1

Короче, после пересечения 2к комментариев стринги у меня вообще перестали грузить полностью комменты.
На мозилке (во всяком случае старой) — полет нормальный.
Я считаю, что наглядное подтверждения тезисов статьи даже искать не надо!

titbit 23 октября 2018 в 18:38

+1

Причин описанному в статье несколько:

1. Производительность железа растет, а стоимость пока удается держать более-менее одинаковую. За последние 20 лет, каждый год «современный» компьютер можно было купить примерно за 1000$, но при этом у него понемногу улучшались всякие объемы памяти, дисков, число и/или частоты процессоров и т.п. Это позволяет не заботится об оптимизации, не тратить на нее время, потому что просто незачем. Этот фактор пройдет, как только гонка железа с софтом замедлится, а подтверждения, что железо уже не может угнаться за софтом уже появляются.

2. В IT за последние годы «поверили» очень много людей, все бросились автоматизировать всё подряд и переходить на электронные носители и оборот и т.д. Это породило всплеск огромного кол-ва разнообразного софта, написанного быстро и некачественно (включая «кровавый энтерпрайз»), потому что надо было занимать и окучивать новые рынки и как можно быстрее становится там монополистами, чтобы получать максимум выгоды. Этот процесс до сих пор еще идет, хотя число незанятых ниш сильно сократилось. Да и потихоньку люди понимают, что IT — это не серебряная пуля.

3. Индустрия разработки (и доставки) контента прошла за последние 20 лет огромный рывок вперед. Многократно выросли размеры изображений, видео, качество музыки, появились всякие 9.1 звуки, 4Д видео и т.д. Да еще и помимо простого увеличения в размерах кол-во самого контента выросло тысячекратно, т.е. музыки, видео и фото стало во много раз больше. Для быстрой (а объемы растут очень быстро!) обработки всего этого барахла очевидно лепится очередной чудо-софт, не блещущий архитектурой. Этот фактор пока еще будет действовать в ближайшие десятилетия, т.к. тут и ниш больше, да и тупо размеры можно гнать вперед довольно долго, уповая на ненасытность людей. 8К видео — вчерашний день, у нас уже 64К видео есть, да еще в 960 фпс, да еще по 16 бит на компоненту цвета…

4. В современном мире софта можно реализовать практически что угодно, а вот продать удается далеко не всё. А это определяет стратегии. Действовать надо быстро, как только ниша появилась (освободилась), значит плевать на всё, пилим лишь бы хоть как-то работало. Потом допиливать даже не будем, потому что уже новые «вызовы» будут. Этот фактор будет действовать и дальше, т.к. он экономический, а не из области IT.

5. Стоимость разработки качественного (во всех отношениях) софта за последние 20 лет непрерывно растет. Причем растет быстрее, чем растет эффект от его внедрения. Архитектуры железа и софта усложнились неимоверно, наверное, уже в мире нет человека, который бы понимал бы от А до Я как работает его техника. Стоимость качественных программистов тоже растет, и бизнесу это не нравится. Поэтому компании вынуждены либо говнокодить, снижая стоимость, либо уходить в узкие ниши, т.е. становится махровыми монополистами и выкручивать руки всем подряд. Иными словами разрыв между сложностью и оптимальностью растет, и баланс перекашивается. Этот фактор тоже продолжит действовать, т.к. сложность имеет свойство накапливаться, так же накапливаются слои и уровни абстракции в современном софте, но это не делает его принципиально проще.

6. Ну и наконец, психологический фактор. В мире ежечасной погони за новыми версиями, сомнительными и высосанными из пальца функциями софта, переписывания иконок и внешнего вида, и прочей «перестановки кроватей», по-другому просто не бывает. Тут замешана и экономика, и человеческий фактор (хайп вокруг IT), и непонимание куда вообще двигаться в IT, причем ни программистами, ни компаниями, отсутствие вменяемых долгосрочных стратегий (ну а зачем, пипл хавает же) и т.д. Все это приводит к тому, что на проблемы «ожирения» просто не обращают внимание. Этот фактор наверное будет действовать до тех пор, пока не сменится понимание функций IT в этом мире, причем как пользователями, так и разработчиками.

Может сумбурно, но мне видится как-то так. Бороться в данный момент с «ожирением» и общим отупением софта глупо, надо этим пользоваться :)

Emulyator 23 октября 2018 в 21:35

+1

Господа, помяните уже Гитлера, подумайте о детях и прекращайте добавлять комментарии, страница и так прогружается с 10 раза.)

sumanai 23 октября 2018 в 23:01

0

А я думал до 3к дотянем.

Druu 24 октября 2018 в 09:36

0

Мне кажется, что-то оптимизировали в сторону замедления работы, потому что с последними ~200 постами стало прям совсем невыносимо скачком :)



Am0ralist 24 октября 2018 в 14:17

0

Вот не надо ля-ля. Мозила тупит, но грузит с первого раза! *Просто нечего пользоваться неподходящими для Хабра браузерами!*

khim 24 октября 2018 в 14:41

0

Так Хром тоже тупит, но грузит с первого раза. Не знаю уж чем пользуется @Emulyator. Может он просто несколько нетерпелив?



Am0ralist 24 октября 2018 в 14:46

0

Стринги у меня на 1,5к отваливаются недогрузив комменты (оперативки всего 16, SSD диск, я даже не знаю, что им не нравится).
А так, даже IE быстрее грузит)

tvr 24 октября 2018 в 15:50

0

> оперативки всего 16, SSD диск

Да вы мажор, однако. Win XP Pro SP3, какой-то из пней, ОЗУ 4Gb, мозилла чуть задумывается при загрузке этого тредика, но при наборе коммент никаких тормозов не вижу.



Am0ralist 24 октября 2018 в 16:13

0

> Да вы мажор, однако.

Это был сарказм в отношении хромых браузеров)))
Мозила же старенькая у меня и то же жует без проблем, хотя около 2к у мне тоже показалось, что дальше и она не справится, однако — оказалось, что только показалось.

BigBeaver 24 октября 2018 в 15:54

0

У меня ничего не отваливается нигде, но все ужасно тупит везде. Оперативки всего 12, ссд нет.

Emulyator 24 октября 2018 в 20:33

0

У меня Яндекс — он лучше остальных подошел после 12-ой Оперы, менять пока не собираюсь.



postgres 24 октября 2018 в 21:58

0

Эта страница отличный тест для браузера. Я удивлен, что мой Iphone 6s в Safari скроллит ее комфортнее чем стационарный компьютер с хромом, который съел пол гигабайта оперативки на эту вкладку

Только полноправные пользователи могут оставлять комментарии. Войдите, пожалуйста.

САМОЕ ЧИТАЕМОЕ

Сутки
Неделя
Месяц

Суд обязал Роскомнадзор заблокировать «пиратские» научные порталы на территории России

+50 23,6k 50 154

Почему «Юному технику» не удастся построить лазер

+108 34,7k 85 112

Настолько ли стар твой Windows?

+24 22,9k 157 113

Как я чуть не подхватил вирус, пытаясь продать сапоги

+129 24,7k 94 58

Китайское искусственное солнце…

+36 17,5k 32 43

ХАБР РЕКОМЕНДУЕТ

Разместить



Настало время шести историй digital-компаний о Tilda

Конкурс IT-статей для авторов Хабра

© 2006 – 2018 «TM»