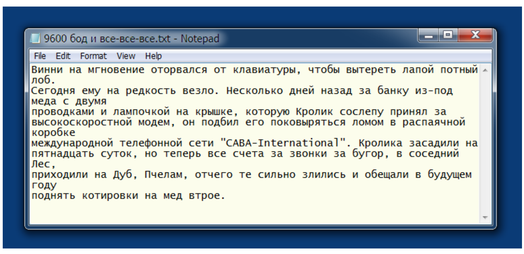
3 августа в 20:12

Многострадальный notepad: ошибка, которую не исправляют уже 13 лет

Реверс-инжиниринг*, Assembler*

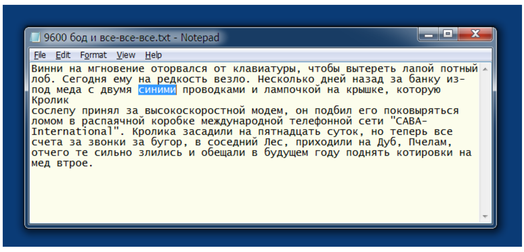


В стандартном блокноте для всех версий Windows, начиная примерно с 2001 года, имеется ошибка, про которую практически все знают, но никто не собирается её исправлять. И это понятно, ведь это не критическая уязвимость, ничьей безопасности она не угрожает. Да и пользуется ли кто блокнотом вообще?

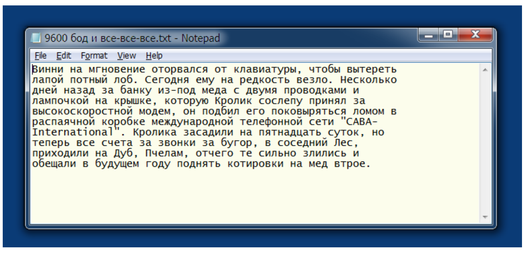
Тем не менее, сам факт довольно странный, поэтому мы попробуем найти эту ошибку в коде 64-битного и 32-битного notepad.exe от windows 7, исправим её, и выясним наконец, почему же она возникла. Заключается ошибка в следующем:

Если в блокноте включена опция «перенос по словам» (word wrap), то **после сохранения файла** начинаются всевозможные глюки: строки начинают разъезжаться, курсор улетает, текст вводится не туда, куда вы ожидаете, и так далее.

Для начала попытаемся поточнее выяснить, что же происходит. Откроем или введём какой-нибудь текст с длинными строками, чтобы они переносились. Сохраним файл. Если теперь попытаться его редактировать, например, добавив слово «синими», строки будут переноситься неправильно, ломая форматирование:



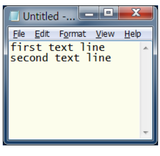
Если уменьшать окно блокнота, строки разрезаются (это видно на заглавной картинке), а при растягивании остаются на месте, не заполняя увеличивающееся окно. Как будто в каждой строке появился жесткий «перевод строки» в том месте, где она заканчивалась в момент сохранения. Видимо текст каким-то образом портится в памяти:



Если же теперь снова сохранить файл, станет ещё хуже. Все строки переформатируются, но окно не обновится. Поэтому курсор может переместиться в другое место, а если начать вводить текст, окажется, что вы вводите его не в то место, где находится курсор, а совсем в другое. Программисты, которые писали notepad, рассуждали логично: при сохранении файла ничего в окне не должно поменяться, поэтому и нет смысла его обновлять. Но в нашем случае с учётом этой ошибки весь текст меняется. Воспроизвести ситуацию может каждый пользователь windows, потому что последняя версия, где этой ошибки не было — Windows'98, и вряд ли у кого она ещё осталась.

Итак, по всей видимости, при сохранении файла что-то идёт не так и текст портится. Как найти это место в коде? Откроем notepad.exe в каком-нибудь отладчике. Как известно, в 64-битной системе для совместимости имеется два блокнота: 32- и 64-битный, надо не перепутать их.

Введём текст, на котором легко будет увидеть, как он портится при переносе строк. Наберём в одну строку «first text line second text line», а затем уменьшим окно так, чтобы она разрезалась посередине.



Резонно будет предположить, что запись делается с помощью функции WriteFile. Оказывается, она вызывается в коде целых 6 раз. Недолго думая, поставим точки останова на все 6 вызовов. Запускаем блокнот и нажимаем «сохранить». Выполнение останавливается здесь:

```
00000000FFA38A7C  and         qword ptr [rsp+20h],0
00000000FFA38A82  lea         r9,[rsp+40h]
00000000FFA38A87  mov         r8d,esi
00000000FFA38A8A  mov         rdx,rbx
00000000FFA38A8D  mov         rcx,r12
00000000FFA38A90  call        qword ptr [0FFA3C140h]  ; WriteFile
00000000FFA38A96  mov         edi,eax
00000000FFA38A98  mov         rcx,rbx
00000000FFA38A9B  call        qword ptr [0FFA3C258h]  ; LocalFree
00000000FFA38AA1  mov         eax,edi
```

Посмотрим все регистры, где содержатся параметры вызова. В rcx у нас 104, это непонятно что. А rdx = 002D45E0, это похоже на адрес в памяти. Посмотрим, что там.

```
002D45E0  66 69 72 73 74 20 74 65 78 74 20 6c 69 6e 65 20  first text line
002D45F0  73 65 63 6f 6e 64 20 74 65 78 74 20 6c 69 6e 65  second text line
002D4600  0d 0a 00 00 00 00 00 00 ea 27 5c 58 9f 00 00 88  ........к'\Xu..€
```

Отлично. Отсюда у нас идёт запись. Попробуем выполнить код дальше, чтобы посмотреть, где он портится. Однако почти сразу данные затираются, а это значит, что это всего лишь временный буфер, а сам текст хранится где-то ещё. Посмотрим выше по программе.

```
00000000FFA38A51  mov         qword ptr [rsp+38h],r15
00000000FFA38A56  and         qword ptr [rsp+30h],0
00000000FFA38A5C  mov         r9d,edi
00000000FFA38A5F  mov         r8,r13
00000000FFA38A62  mov         edx,r14d
00000000FFA38A65  mov         ecx,ebp
00000000FFA38A67  mov         dword ptr [rsp+28h],esi
00000000FFA38A6B  mov         qword ptr [rsp+20h],rax
00000000FFA38A70  call        qword ptr [0FFA3C150h]  ; MultiByteToWideChar
00000000FFA38A76  mov         edi,eax
00000000FFA38A78  test        eax,eax
00000000FFA38A7A  je          00000000FFA38A98
```

Ага, перед сохранением текст видимо преобразовывается из многобайтовой кодировки в однобайтовую. Точно так же, как в прошлый раз, посмотрим параметры. rax = 002D45E0, здесь у нас пока нули. Это как раз то место, куда попадёт результат. esi = 20, это длина текста. ecx = 4e3, без комментариев. edx = 400, то же самое. А вот r8 = 002D6780:

```
002D6780  66 00 69 00 72 00 73 00 74 00 20 00 74 00 65 00  f.i.r.s.t. .t.e.
002D6790  78 00 74 00 20 00 6c 00 69 00 6e 00 65 00 20 00  x.t. .l.i.n.e. .
002D67A0  73 00 65 00 63 00 6f 00 6e 00 64 00 20 00 74 00  s.e.c.o.n.d. .t.
002D67B0  65 00 78 00 74 00 20 00 6c 00 69 00 6e 00 65 00  e.x.t. .l.i.n.e.
```

Снова продолжим выполнение, наблюдая за содержимым этого участка памяти. Через несколько десятков команд мы выходим из подпрограммы, выполняются какие-то переходы, вызовы, но мы, не обращая на это внимания, продолжаем давить на «step over», выполняя код по шагам, и следя только за окном с текстом. И вот в какой-то момент он изменяется. Как видим, между 1 и 2 строкой появились коды 0d, 0d, 0a:

```
002D6780  66 00 69 00 72 00 73 00 74 00 20 00 74 00 65 00  f.i.r.s.t. .t.e.
002D6790  78 00 74 00 20 00 6c 00 69 00 6e 00 65 00 20 00  x.t. .l.i.n.e. .
002D67A0  0d 00 0d 00 0a 00 73 00 65 00 63 00 6f 00 6e 00  .....s.e.c.o.n.
002D67B0  64 00 20 00 74 00 65 00 78 00 74 00 20 00 6c 00  d. .t.e.x.t. .l.
002D67C0  69 00 6e 00 65 00 0d 00 0a 00 6c 00 69 00 6e 00  i.n.e.....l.i.n.
```

Как обычно бывает, мы проскочили нужную команду, постоянно давя на кнопку, поэтому придётся повторить всё

Популярное за сутки

Обнаружен ботнет, который исправляет уязвимости в зараженных им маршрутизаторах и сообщает об этом администратору 7

О безопасности UEFI, часть шестая

Обработка аннотаций в процессе компиляции 2

Разработка и публикация ассета в Unity Asset Store 1

Ускоряем отладку и прототипирование мобильных QML-приложений на живом устройстве 8

4 must-have-элемента тестирования ПО

Дайджест интересных материалов для мобильного разработчика #123 (28 сентября-4 октября)

Тутор по BabylonJS — камера и освещение

Дайджест интересных материалов из мира веб-разработки и IT за последнюю неделю №179 (28 сентября — 4 октября 2015)

все лучшие

Компания дня ?

PVS-Studio

Последняя публикация: Первый вздох PVS-Studio для C#

851 подписчик

ещё раз, запомнив, где примерно это произошло. Теперь по мере приближения к нужному месту в коде, замедляемся, и точно определяем, что текст испортился вот на этом вызове:

```
00000000FFA38ECE  call        qword ptr [0FFA3C178h]     ; LocalUnlock
00000000FFA38ED4  cmp         dword ptr [0FFA40048h],esi
00000000FFA38EDA  je          00000000FFA38EE1
00000000FFA38EDC  call        00000000FFA38AC8
00000000FFA38EE1  mov         rcx,qword ptr [0FFA400E0h]
00000000FFA38EE8  call        qword ptr [0FFA3C500h]     ; SetCursor
```

Можно попробовать, что будет, если не делать этот вызов. Снова доходим до этого места, и прямо тут, в отладке, изменяем RIP (регистр, где хранится адрес выполняемого в данный момент кода) на 00000000FFA38EE1, как будто мы пропустили этот call, который нам всё испортил. Удивительно, всё работает, текст не ломается!

Тут надо сказать, что в таких случаях обычно не разбираются, что это за подпрограмма, что она делает и зачем, а просто выкидывают её из EXE-файла. Это можно сделать разными способами, например, забить её всю NOP'ами, или изменить условный переход по равенству «je», который так кстати имеется сразу перед ней, на безусловный «jmp».

Но нам сейчас не столько нужно исправить эту ошибку, как интересно выяснить, откуда же она вообще взялась. Поэтому заходим внутрь и смотрим:

```
00000000FFA38AC8  sub         rsp,28h
00000000FFA38ACC  cmp         dword ptr [0FFA40048h],0
00000000FFA38AD3  je          00000000FFA38B23
00000000FFA38AD5  cmp         dword ptr [0FFA40054h],0
00000000FFA38ADC  je          00000000FFA38AEA
00000000FFA38ADE  mov         ecx,50200104h
00000000FFA38AE3  call        00000000FFA394F0
00000000FFA38AE8  jmp         00000000FFA38B23
00000000FFA38AEA  mov         rcx,qword ptr [0FFA400A0h]
00000000FFA38AF1  xor         r9d,r9d
00000000FFA38AF4  mov         edx,0C8h
00000000FFA38AF9  lea         r8d,[r9+1]
00000000FFA38AFD  call        qword ptr [0FFA3C578h]     ; SendMessage
00000000FFA38B03  mov         r9d,dword ptr [0FFA41A28h]
00000000FFA38B0A  mov         r8d,dword ptr [0FFA41A30h]
00000000FFA38B11  mov         rcx,qword ptr [0FFA400A0h]
00000000FFA38B18  mov         edx,0B1h
00000000FFA38B1D  call        qword ptr [0FFA3C578h]     ; SendMessage
00000000FFA38B23  add         rsp,28h
00000000FFA38B27  ret
```

Вот такая замечательная маленькая подпрограмма. Проходим её по шагам. Сначала сравниваются какие-то две переменные с нулём, в результате первый вызов неизвестно чего не делается, а делаются подряд два вызова SendMessage. То есть, всё, что происходит, это посылается два каких-то виндовых сообщения, причём текст портится сразу же после первого (выделен зеленым). Невооруженным глазом видно, что в EDX передаются их коды (выделен красным). Поищем код 0C8h.

Это оказывается сообщение EM_FMTLINES. Довольно похоже, посылаем сообщения для форматирования строк, вот и доформатировались. Пришло время почитать документацию. MSDN сообщает нам следующее:

> Это сообщение определяет включение «мягких» переводов строки в многострочный элемент редактирования. «Мягкий» перевод строки представляет из себя два символа [CR] и один [LF] и вставляется в строку там, где она разрезается при переносе по словам.
>
> Параметр wParam: true — вставить символы, false — удалить их.
>
> Сообщение влияет только на буфер, возвращаемый сообщениями EM_GETHANDLE и WM_GETTEXT, и не влияет на текст, отображаемый в элементе редактирования. Также оно не влияет на «жёсткие» переводы строки, которые состоят из одного [CR] и одного [LF].

Кроме того, мы узнаём, что данное сообщение было введено не позднее чем в Windows 95. Ну вот всё и стало понятно. В 95 году предполагалось, что оно не влияет, а сейчас видим, что влияет, да ещё как. Немного поизучав код, находим несколько аналогичных вызовов, и нашему мысленному взору предстаёт следующая картина:

Давным-давно, в первой половине 90-х годов, программисты Microsoft писали блокнот для Windows 95. Чтобы реализовать замечательную функцию переноса строк, они придумали посылать окну (или его элементу) сообщение, чтобы оно само переформатировало себя, навставляв специальных символов. Чтобы эти символы отличить от нормального перевода строки, они придумали последовательность 0d, 0d, 0a. Чтобы она не попадала в файл, перед сохранением все такие коды удалялись, а после сохранения добавлялись обратно.

Позже, когда делали windows XP, элемент стал сам всё переносить как надо, и ему уже не нужно было это сообщение. Однако, никто уже не помнил, зачем оно было нужно, и поэтому решили на всякий случай оставить как было. Тем более, вроде бы всё работало, а проблем после сохранения никто не заметил. С тех пор этот код так и остался, дойдя до самых последних версий Windows 7 и 8. Десятку я не ставил, но скорее всего, там он тоже есть.

Перейдем теперь к исправлению ошибки. После сообщения 0C8h посылается ещё 0B1h, а это EM_SETSEL — установка выделения. Похоже, выкидывать эту подпрограмму целиком всё же неправильно, да ещё там есть какой-то непонятный вызов в начале. Поэтому лучше удалить только первый вызов SendMessage, или поменять его параметр с 1 на 0, или изменить переход на другой адрес, чтобы после проверки переменной [0FFA40054h] сразу переходить ко второму вызову. Вариантов много, но результат будет одинаковый.

```
00000000FF118AEA 48 8B 0D AF 75 00 00 mov         rcx,qword ptr [0FF1200A0h]
00000000FF118AF1 45 33 C9             xor         r9d,r9d
00000000FF118AF4 BA C8 00 00 00       mov         edx,0C8h
00000000FF118AF9 45 8D 41 01          lea         r8d,[r9+1]
00000000FF118AFD FF 15 75 3A 00 00    call        SendMessage
```

Где же здесь параметр, равный 1? Всё очень просто — он в регистре r8. Для сокращения кода компилятор никогда не использует прямую пересылку нуля в регистры. Такая команда занимает 6 байтов: 2 байта код операции, 4 байта — 32-битный ноль. Вместо этого регистр XOR-ится сам с собой, в итоге получается ноль, и это занимает всего 3 байта. После этого r9, который равен нулю, пересылается в r8 с добавлением единицы (выделена зеленым). Эта операция тоже занимает всего 4 байта. Вот эту зеленую 1 нам и надо поменять на 0, и тогда текст не будет портиться.

А теперь найдём эту же процедуру в 32-битной версии блокнота. Если не хочется повторять все те же манипуляции с отладкой, её можно найти простым поиском числа 0C8h.

```
00067BA2  xor         eax,eax
00067BA4  cmp         dword ptr ds:[6C034h],eax
00067BAA  je          00067BF0
00067BAC  cmp         dword ptr ds:[6C040h],eax
00067BB2  je          00067BBF
00067BB4  push        50200104h
00067BB9  call        00068383
00067BBE  ret
00067BBF  push        esi
00067BC0  mov         esi,dword ptr ds:[61234h]
00067BC6  push        eax
00067BC7  push        1
00067BC9  push        0C8h
00067BCE  push        dword ptr ds:[6C028h]
00067BD4  call        esi  ; SendMessage
00067BD6  push        dword ptr ds:[6D790h]
00067BDC  push        dword ptr ds:[6D794h]
00067BE2  push        0B1h
00067BE7  push        dword ptr ds:[6C028h]
00067BED  call        esi  ; SendMessage
00067BEF  pop         esi
00067BF0  ret
```

Как видим, совершенно аналогичный код, только 32-битный. Теперь, чтобы исправить ошибку, осталось только найти это место в exe-шнике и поменять нужный байт. Перед этим не забудьте стать владельцем файла и дать себе права на его изменение.

64-битный notepad.exe (193536 байт) поменять байт по адресу [80FC] с 1 на 0
32-битный notepad.exe (179712 байт) поменять байт по адресу [6FC8] с 1 на 0

Не сомневаюсь, где-то в недрах майкрософтовского кода ещё много таких мест, где спят древние баги, которые, скорее всего, никто никогда не исправит. Нам остаётся только надеяться, что все они такие же безобидные как этот, и ничего страшного не случится, когда они будут перенесены в следующую операционную систему, которую с удовольствием установят себе пользователи по всему миру.

реверсинг, блокнот, notepad, древние баги

Похожие публикации

- YouTrack 6.5 — Баг-трекер для всей команды (32)
- 10 правил хорошего тона при описании багов (65)
- Букмарклеты в Internet Explorer 11: формат хранения, лимиты и негласные правила, коварный баг (6)
- Баг в софте автомобилей Land Rover приводит к самопроизвольному отпиранию дверей (20)
- Как студент баг в Яндекс.Музыке нашел (28)

Комментарии (106)

kentastik 3 августа 2015 в 20:21 #

+2

Или я что-то делаю не так или в десятке такого бага нет — yadi.sk/i/kW8wzzDDiFMhH

wapmorgan 3 августа 2015 в 20:35 # ↑

+14

Сначала сохраните, а потом уже меняйте размер.
Есть баг

▶ Скрытый текст

kentastik 3 августа 2015 в 20:40 # ↑

+5

точно, поторопился :)

Chijikson 3 августа 2015 в 20:38 # ↑

–40

Подтверждаю, в 10 такого бага не наблюдается.

Nikobraz 3 августа 2015 в 20:59 # +9

Win10, баг наблюдаю

amdf 3 августа 2015 в 21:02 # +9

Ещё у notepad.exe есть секретный ключ в командной строке, который вызывает странное поведение.

notepad /.SETUP

turbo_exe 3 августа 2015 в 21:31 # ↑ +2

а что происходит при таком ключе?

middle 3 августа 2015 в 21:50 # ↑ +54

происходит странное поведение.

xpert13 3 августа 2015 в 22:05 *(комментарий был изменён)* # ↑ +6

На окно невозможно навести фокус. Как только кликаешь на него — оно прячется. По Alt+Tab так же нельзя переключится, его банально нету в списке

EvilFox 3 августа 2015 в 22:46 # ↑ +9

Нашёл такую страницу, а там:

> /.SETUP SetupMode: E.g. notepad /.setup file.txt. I'm unclear what this is used for. It's a weird mode, it does not repaint the window if it was started restored. You'd have to press Alt-Space and Maximize it to view the file content. The window has 2 sets of scrollbars in that case (one set is apparently unused), and it closes if Escape or Ctrl+D are pressed. Perhaps some setup programs invoke notepad with these arguments to display the EULA? Who knows.

stas404 4 августа 2015 в 01:42 # ↑ +11

Подозрительная активность.

perfectdaemon 4 августа 2015 в 07:47 # ↑ +10

Вы наверняка знаете, какой антивирус вас спасет

ForeverYoung 3 августа 2015 в 21:07 # +1

Предлагаю баг на исправление — выделяю строчки в текстовом файле, начиная снизу, двигаясь вверх (с shift'ом). Переключаюсь в другую программу, потом обратно, опять зажимаю shift — курсор уже внизу блока. Windows XP, другие не проверял.

lolmaus 4 августа 2015 в 15:50 # ↑ +6

Предлагаю баг на исправление: bugs.launchpad.net/ubuntu/+bug/1.

bodqhrohro 3 августа 2015 в 21:41 # 0

Чем-то напоминает проблему с переносом текста в Opera Mini. Только там переносы не временные, а самые что ни на есть обычные, и при копировании текста их приходится вручную заменять пробелами или удалять. Что характерно, для новых версий обновляется и формат OBML, но воз и ныне там, хотя обратную совместимость, казалось бы, поддерживать не нужно.

Gorodnya 3 августа 2015 в 21:49 # +1

У меня в Excel на работе нельзя лист назвать словом с большой буквы «Ж») Другие буквы он принимает в начале слова, эту — нет, «Ж» в названии первой буквой может быть только маленькой)

xpert13 3 августа 2015 в 22:03 # ↑ 0

Судя по всему оно вообще не принимает большую «Ж», ни в начале, ни в середине

Gorodnya 3 августа 2015 в 22:05 # ↑ 0

Почему же, принимает строчную вначале и дальше нормально.

xpert13 3 августа 2015 в 22:09 # ↑ 0

Office 2013 — при попытке ввода в имени листа большой «Ж» (Shift+Ж) ничего не происходит, вне зависимости от того где находится курсор, проверил несколько раз. Но что самое интересное, так это то, что CAPS, а потом «ж» ставит большую «Ж» (опять таки вне зависимости от положения курсора, в том числе и в начале).

Gorodnya 3 августа 2015 в 22:14 # ↑ +1

Хм. Странно. Завтра проверю: кажется, у меня её можно было установить с зажатым шифтом (главное, чтобы не первой). Но согласитесь, всё равно странно — это же не твёрдый знак, не мягкий. Единственный вариант, который мне приходит в голову (но наверняка это просто совпадение) — буква находится на клавише с двоеточием в английской раскладке, поэтому и невозможно.

xpert13 3 августа 2015 в 22:19 # ↑ +1

Кстати да, вы правы: единственные клавиши в буквенном рядку клавиатуры, которые не добавляют символы с зажатым шифтом — это «Ж»(английская клавиша ":") и "," (английская "?"). В обеих случаях не ставится и в русской и в английской раскладке и в обеих случая то, что находится в русской раскладке можно добавить через копи-паст

khim 4 августа 2015 в 01:26 # ↑ +5

Почему не вставляется в английской раскладке — понятно (эти символы запрещены в именах файлов), а вот кто и каким местом написал проверку так, что она не зависит от раскладки — науке неведомо.

ID_Daemon 4 августа 2015 в 18:09 # ↑ +1

Они сначала отбрасывают 13 комбинаций по скан-коду, в том числе shift-Ж, а потом уже фильтруют запрещенные символы. То есть если убрать этот скан-код из списка, то 'Ж' можно будет ввести, а ':' по-прежнему нельзя, я проверил. Думаю не стоит по этому поводу писать статью, как это найти и где исправить.

ID_Daemon 6 августа 2015 в 20:16 # ↑ +4

Как оказалось, всё было сложнее — habrahabr.ru/post/264313

SVlad 4 августа 2015 в 16:17 # ↑ −1

Если она ставится с caps, но не ставится с шифтом, то, похоже, это какой-то хоткей «shift + ;» который перехватывает события клавиши.

dj_raphael 3 августа 2015 в 22:11 # ↑ 0

её нельзя туда напечатать но можно вставить

petuhov_k 4 августа 2015 в 05:32 # ↑ +1

У меня разрешает. Office 2013 en, Win7 en.

▸ Скрин

dMetrius 4 августа 2015 в 11:58 # ↑ 0

Можно вставить с помощью Alt+134 в русской раскладке.
habrastorage.org/files/e21/fdd/326/e21fdd326fba4c12ac97bb83d9b1f2cc.png

mayorovp 1 сентября 2015 в 17:03 # ↑ 0

Уже где-то на хабре была статья с разбором.

Кажется, такое происходит, если первое нажатие любой клавиши при активном Excel выполнялось при включенной английской раскладке. В таком случае комбинация SHIFT+; оказывается заблокированной (потому что двоеточие — невалидный символ для названия листа), русская же буква «Ж» блокируется вместе с ним заодно.

Mingun 1 сентября 2015 в 19:19 # ↑ 0

Да вот же она, написана как раз в «ответ» на родительский комментарий автором этой же статьи :)

mayorovp 1 сентября 2015 в 19:54 # ↑ 0

Хм, даже и не заметил как месяц прошел… :)

roboter 3 августа 2015 в 22:09 # +1

Ещё баг на исследование.
Заметил нелогичное поведение иконок на десктопе.
Иконки появляются на десктопе столбиком, но, если выделять с шифт то выделяются по строкам.
Заметил когда случайно распаковал зип на деском и хотел удалить.

Mulin 3 августа 2015 в 22:33 *(комментарий был изменён)* # +14

Есть еще один забавный баг, который никто и не думает исправлять. Во всех Windows что я ипользовал, в трее сохраняется значек закрытой программы до тех пор, пока не проведешь по нему мышкой. С чем это связано не знаю, но как-то где-то и от кого-то слышал такую информацию, что, дескать, в винде единственное что поддерживается на уровне железа видеокартой, это курсор мыши. Уж не знаю что это значит и правда ли это, но как-то навевает.

EvilFox 3 августа 2015 в 22:56 *(комментарий был изменён)* # ↑ +12

Насколько помню, это происходит потому что проводник подписывается на событие программы касаемо значков в области уведомления, если программа завершается неправильно то событие разрегистрации значка не приходит, наведение указателем инициирует какую-то функцию где видимо проверяется существование процесса. Что им мешало раз в некоторое время дёргать проверку существования процессов хз.

cjfynjy 4 августа 2015 в 01:29 # ↑ +6

> наведение указателем инициирует какую-то функцию где видимо проверяется существование процесса

Вряд ли прямо проверяется существование процесса, скорее там просто происходит молчаливая обработка ошибки. Вообще по ховеру иконки должно происходить обращение к процессу за тем, чтобы тот предоставил текст всплывашки. Запрос всплывашки возвращает ошибку или эксепшен — иконка убирается.

MrShoor 4 августа 2015 в 03:52 # ↑ +3

А эксепшн бросает / код ошибки возвращает кто? Разве не код, проверяющий существование процесса?

a553 4 августа 2015 в 01:33 # ↑ +7

Это происходит, если насильно убить процесс, не отправляя ему сообщений о терминации. Нет сообщений — нет и обработчиков событий, вот иконка и не пропадает.

Mulin 4 августа 2015 в 01:46 *(комментарий был изменён)* # ↑ −1

В том-то и баг, что это не всегда обязательное условие. Иногда я подозреваю Nvidia, с радеонами таких проблем вроде не замечал.

a553 4 августа 2015 в 01:54 # ↑ 0

Видимо у вас какой-то обработчик вылетает по таймауту. Антивирус, драйвер клавиатуры или мыши, ещё что-нибудь такое.

Mulin 4 августа 2015 в 02:22 # ↑ 0

На трех поколениях процессоров и 6 различных версиях Windows? Тут скорее биополе у меня того-этого)).

a553 4 августа 2015 в 02:26 # ↑ +1

Ну, может вы смотрите на него как-то не так (шутка конечно). У меня тоже парк машин и нигде спонтанного проявления бага не было :)

Bringoff 4 августа 2015 в 07:19 # ↑ 0

У меня такое постоянно и с разными программами, которые я «насильно» не завершаю. И кстати, у меня в ноуте интел и радеон видяхи, нвидиа отсутствует.

a553 4 августа 2015 в 08:11 *(комментарий был изменён)* # ↑ 0

Да у меня тоже куча сетапов со всевозможными конфигурациями. И везде баг проявляется только с насильным убийством через таск менеджер. Может, какое-то конкретное приложение это делает. Что-нибудь сильно внедряющееся в систему типа AltDrag или Punto Switcher (не они).

Зато у меня на ноутбуке периодически выбранный элемент на таск баре «подвисает» — система думает, что мышь на него всё время наведена. Баг пережил две чистых переустановки системы 8 → 8.1 → 10. Подозреваю какой-то софт от вендора.

TheRaven 4 августа 2015 в 10:11 # ↑ +1

Win 7 x64, Intel, Nvidia
Стабильно наблюдаю этот баг при использовании Mozilla Thunderbird. Когда приходит письмо генерится значок и оповещение, после закрытия программы значок остается. При чем их можно несколько накопить.

silvansky 4 августа 2015 в 11:03 # ↑ 0

Скорее, это баг в Thunderbird, может туда зарепортить?

Я остающиеся иконки наблюдал много раз ещё в XP, но обычно это было при неправильном завершении программы. Бывало и просто так, кажется, но это, видимо, из-за кривизны софта.

Когда я делал мессенджер под винду с иконкой в трее, тоже нарывался на этот баг при каждом креше мессенджера или остановкой в дебаге. И тоже накапливал, бывало, 5-10 значков, убиваемых одним привычным взмахом мышки по трею.

_silvansky 4 августа 2015 в 11:05 # ↑ 0

Разумеется, меня это бесило. И я не понимаю, почему MS не сделали эту работу за меня. В OS X я наблюдал подобное, но значок в трее висел после снятия программы от силы секунд 5-10, видимо, всё же там таймаут есть системный. Хотя и тут не всё гладко: иногда значок пропадает, а место от него остаётся, и убрать его можно лишь кликом мыши.

TheRaven 4 августа 2015 в 11:45 # ↑ 0

Возможно их баг, отписался на support@mozilla-russia.org
Посмотрим что ответят.

edogs 3 августа 2015 в 22:36 # +2

Нам всегда помогало при такой проблеме отключить режим переноса строк, а потом снова его включить.

Nyashkoshkko 25 сентября 2015 в 15:30 # ↑ 0

А мне помогает после сохранения свернуть блокнот в трей, а потом развернуть его заново.

evocatus 3 августа 2015 в 23:03 # +3

А если написать в Microsoft и попросить исправить?

varnav 4 августа 2015 в 09:26 # ↑ 0

Теперь ведь есть windows.uservoice.com

spmbt 4 августа 2015 в 02:03 # −10

Всегда помогало пользоваться другим текстовым редактором.

spmbt 4 августа 2015 в 04:17 # ↑ −8

Почему? С чем не согласны? В самом деле же помогало. Сумма багов и неудобств. У Notepad есть ещё более мощный баг — непонял LF как перевода строки, благодаря чему пользоваться им почти не приходилось — сразу заменялся на AkelPad и другие более ранние аналоги — BrEd3, UE32. Плюс заворот строк делает несколько нетрадиционно (то ли на минусах не заворачивает, то ли на подобном), поэтому вид строк с заворотами отличается от всех других редакторов. И зачем таким пользоваться или исправлять, если установить дружественный и нормальный — крайне просто? Никогда проблемы, как с Vim не возникает, когда нельзя что-то установить на удалённый сервер. Всякая установка Windows предполагает доустановку: редактор, Эксплорер, Калькулятор, Просмотр картинок, видео. Иногда — редактор реестра и донастройку. Это как 2*2.

Kanick 4 августа 2015 в 04:37 # ↑ +4

Очевидно, с тем, что пост не о том, чем кому-то пользоваться, а о баге в популярной программе. Да и калькулятор не всем нужно доустанавливать.

poxu 4 августа 2015 в 09:56 # ↑ +2

Таки не популярной, а широкоизвестной :)

Zveroloff 4 августа 2015 в 10:13 # ↑ +1

Действительно, а что не так с калькулятором?

Palomnik 4 августа 2015 в 12:25 # ↑ +7

Если у вас винда виста, семь, восемь или может даже десять (не проверял), любой редакции, переключитесь на инженерный вид, разделите единицу на число двести пятьдесят два и нажмите на функцию «F-E» (она в левом нижнем углу). Приложение упало. Забавно, что если делить, например, на двести пятьдесят один, то функция отработает корректно. Чисел на которых падает эта функция достаточно много, не одно.

Что, разумеется, означает, что калькулятор необходимо доустановить сторонний — потому что пользоваться этим невозможно! :)

Zveroloff 4 августа 2015 в 12:31 # ↑ 0

Не воспроизводится. Win 8.1 x64

TheRaven 4 августа 2015 в 12:35 # ↑ 0

Win XP Professional, не воспроизводится.
Win 7 x64, воспроизводится.

TheRaven 4 августа 2015 в 13:15 # ↑ 0

Попросил проверить на 8.1 — воспроизводится.

a553 4 августа 2015 в 15:31 # ↑ 0

Win 8.1 x64, i7 3770k — нет
Win 10 x64, i5 4200u — нет

EvilFox 5 августа 2015 в 01:39 *(комментарий был изменён)* # ↑ 0

Win 8.1 x64 Pro — не воспроизводится. Может в каком-то обновлении исправили?

Palomnik 4 августа 2015 в 12:44 # ↑ 0

Я на 32-х битной проверял. Очевидно, что это ошибка с плавающей запятой, так что, разрядность системы может влиять на эту ошибку, как мне кажется.

ID_Daemon 7 августа 2015 в 22:41 *(комментарий был изменён)* # ↑ 0

Нет, все вычисления там целочисленные, и ошибка происходит от бесконечной рекурсии. Этот баг уже был проанализирован www.exploit-db.com/docs/30416.pdf

MaximChistov 4 августа 2015 в 14:16 # ↑ 0

падает) Win 7 x64

Bringoff 4 августа 2015 в 15:51 # ↑ 0

Windows 10 не воспроизводится, ибо калькулятор там полностью переписали, судя по всему :) Старого я в системе не обнаружил.
▸ Скрытый текст

olekl 5 августа 2015 в 14:40 # ↑ 0

А еще постоянная необходимость виды калькулятора переключать. Т.к. в програмерском невозможно обычное нецелочисленное деление выполнить.

maybe_im_a_leo 4 августа 2015 в 02:39 *(комментарий был изменён)* # +13

Мне лично в notepad очень не нравится как Ctrl-Z отменяет само себя. Т.е. отмена возможна на один шаг только. Может там тоже такой же досадный баг, до которого нет никому дела…

Kozack 4 августа 2015 в 02:44 # −10

Всегда помогало пользоваться другой ОС.

Kanick 4 августа 2015 в 04:26 # +2

Да, этот баг — это что-то. Когда и в Windows 8 его увидел, захотелось плакать.

Тут же смешно что. У Микрософта дофига сотрудников — и что, никто из них не пользуется Блокнотом в режиме с переносом по словам или не мог сделать баг-репорт? Ох уж эта неповоротливость крупных компаний.

andreishe 4 августа 2015 в 06:20 # ↑ +2

Простите, а каков по вашему мнению сценарий использования блокнота сотрудником Microsoft?

SkidanovAlex 4 августа 2015 в 08:59 # ↑ +4

Как бывший сотрудник Microsoft, который пересекался с другими сотрудниками Microsoft, могу вам сказать, что как и все, сотрудники Microsoft иногда используют блокнот.

AllexIn 4 августа 2015 в 12:44 # ↑ +1

С переносом строк? Это зачем?

Mr_well 4 августа 2015 в 08:22 # ↑ 0

Перенос по словам удобен когда вы пишите литературный текст. Из всех примеров использования блокнота сотрудниками MS я заметил только редактирование скриптов и создание быстрых заметок. И там и там перенос по словам не используется.
Сам пользуюсь блокнотом практически каждый день для снятия форматирования когда надо текст скопипастить, например из браузера в аутлук ну или подправить какие конфиг-файлы.

raacer 4 августа 2015 в 11:47 # ↑ +1

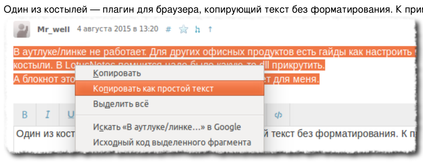
А что, Ctrl+Shift+V не снимает форматирование?

Mr_well 4 августа 2015 в 13:20 # ↑ 0

В аутлуке/линке не работает. Для других офисных продуктов есть гайды как настроить горячие клавиши для этого действия, но это уже костыли. В LotusNotes помнится надо было какую-то dll прикрутить.
А блокнот это железный вариант вот уже больше 10 лет для меня.

ploop 4 августа 2015 в 14:40 *(комментарий был изменён)* # ↑ +1

Один из костылей — плагин для браузера, копирующий текст без форматирования. К примеру:



А так да, простым блокнотом или другим текстовым редактором.

safari2012 4 августа 2015 в 15:09 # ↑ 0

в офисе (включая outlook) хорошо работает Ctrl+Alt+V

Mr_well 4 августа 2015 в 15:25 # ↑ 0

Работает, но это спец. вставка. В линке она действительно убирает форматирование, но в других продуктах вылетит меню.

safari2012 4 августа 2015 в 15:32 # ↑ +1

Не понял проблемы. В аутлуке, word-е и т.п. действительно вылезает меню спец.вставки, выбираешь там «неформатированный текст» и задача решена.

Mr_well 4 августа 2015 в 15:50 # ↑ +2

Да нет никакой проблемы, есть привычка. Ctrl+V это всегда вставка текста из буфера. А вставка неформатированного текста может поддерживаться горячими клавишами, а может и нет. Где-то это Ctrl+Shift+V, где-то Ctrl+Alt+V. А блокнот, он всегда блокнот. В любом случае толик не про это.

darkdaskin 4 августа 2015 в 19:32 # ↑ +1

С Punto Switcher во всех программах работает Ctrl+Win+V (можно сменить).

qwerty1023 4 августа 2015 в 10:10 # ↑ +2

> У Микрософта дофига сотрудников — и что, никто из них не пользуется Блокнотом в режиме с переносом по словам или не мог сделать баг-репорт?

Пользуюсь блокнотом еще со времен Windows 95, но как-то не разу не приходилось в нем включать режим переноса по словам. Честно говоря до этого даже не подозревал, что там такой режим есть :). И пользуюсь блокнотом при этом очень часто.

Zveroloff 4 августа 2015 в 10:16 # ↑ +10

Пользуюсь Windows уже лет 20, в т.ч. Блокнотом. Про этот баг узнал из этой статьи.

Ununtrium 4 августа 2015 в 12:24 *(комментарий был изменён)* # ↑ +1

Извините конечно, но когда есть нормальные редакторы типа Notepad++, стандартный блокнот ни один опытный юзер (тем более программист) в своем уме использовать не станет. При условии, что можно ставить сторонний софт, конечно.
Для любых заметок WordPad раз в 10 удобнее.

Smerig 4 августа 2015 в 13:41 # ↑ +1

Да ладно? Вообще не перевариваю notepad++. Когда хватает обычного блокнота — использую его, когда не хватает — VS.

rock 4 августа 2015 в 13:46 # ↑ +3

Месье знает толк в извращениях.

Ole 4 августа 2015 в 17:13 *(комментарий был изменён)* # ↑ 0

Это да. Но когда ты приходишь к бухгалтеру, чтобы помочь ему отправить бухгалтерскую отчетность, то не станешь ставить ему Notepad++ только для того, чтобы исправить в XML-файле отчетности пару символов.

Приходится открывать notepad и править XML-файл там. А строки там бывают очень большой длины.

gnmb 12 августа 2015 в 01:34 *(комментарий был изменён)* # ↑ 0

Notepad ставит время+дату на F5. Только ради этого и использую. А для сохранения есть комба Ctrl+(S, A).

Vokkz 4 августа 2015 в 08:15 # 0

Нашёл баг в Windows 10, зарепортил в инсайдер хабе, но результата пока никакого.

В трее среди системных значков есть индикатор ввода, который показывает РУС или ENG. Из-за того, что я пользуюсь Punto Switcher, индикатор этот привык отключать, ибо у Punto есть собственный.

Так вот, после перезагрузки Windows, системный индикатор возвращается на место. Всегда. Проверял как Home, так и на Pro-версиях, как на русской редакции, так и на американской — баг есть везде.

silvansky 4 августа 2015 в 11:00 # ↑ 0

Этот баг я ещё в XP ловил.

mayorovp 1 сентября 2015 в 17:08 # ↑ 0

На XP этот индикатор сам собой включается даже если уже был включен…

vbif 4 августа 2015 в 10:39 # 0

А ещё, он до сих пор не понимает ctrl+backspace

nehaev 4 августа 2015 в 11:14 # +5

Всегда недоумевал, почему так происходит в блокноте. Теперь наконец все стало на свои места. Спасибо за такое подробное расследование!

Londoner 4 августа 2015 в 11:15 # +3

Просто оставлю это здесь

vbif 4 августа 2015 в 13:58 # ↑ 0

Всего каких-то 8 лет против более чем 14

khim 4 августа 2015 в 15:37 # ↑ 0

Вы б ещё на сам баг №20786 взглянули! Он из разряда вот этих: начиная с версии MySQL 5.1.11 информация, которая раньше терялась начала запоминаться в дампе. Что, разумеется, немедленно испортило кому-то жизнь. Это совсем другая история, неужели то, что в здешней статье обсуждается!

Psychosynthesis 4 августа 2015 в 13:05 # 0

У меня баг не воспроизводится. Файл сохранял, пересохранял, как только не изгалялся — глюка нет, строки разрезаются при уменьшении окна и возвращаются на место при увеличении.

ЧЯДНТ?

vbif 4 августа 2015 в 13:05 # ↑ 0

А попробуй сначала уменьшить окно, потом сохранить, а потом увеличить.

Psychosynthesis 4 августа 2015 в 13:45 # ↑ 0

Попробовал. Уменьшает, сохраняется уменьшенный вариант форматирования, после этого растягивание окна эффекта не даёт, а уменьшение и последующее растягивание работают корректно. Бага нет.

vbif 4 августа 2015 в 13:47 *(комментарий был изменён)* # ↑ 0

Так в том-то и баг, что после сохранения растягивание окна должно давать эффект.

Psychosynthesis 4 августа 2015 в 13:53 # ↑ +1

Окей, я понял.

Мне просто казалось такое поведение логичным.

safari2012 4 августа 2015 в 18:01 # +1

К стати, Microsoft планирует включить Notepad в Windows Store.
Можно будет все баги зарепортить туда в отзывы и наставить всем сообществом по 1*.
Пусть попробуют не отреагировать :)

delure 4 августа 2015 в 18:05 # 0

Баг с периодическим залипанием ctrl, т.е. клавиша физически не нажималась, но система считает, что клавиша зажата. Закономерности возникновения не выявил. Лечится многократным нажатием сей клавиши.

EvilFox 5 августа 2015 в 01:50 *(комментарий был изменён)* # ↑ 0

Скорее всего клава отвалилась (провод где-то перебит) в момент нажатия клавиши. Ещё такое можно словить с тимвивевером.
В остальных случаях не наблюдал.

miriarder 4 августа 2015 в 20:37 # 0

«В rcx у нас 104, это непонятно что»
На самом деле 104 это hFile типа HANDLE. Что бы понять параметры функции достаточно посетить страницу по API и Calling convention.

Только зарегистрированные пользователи могут оставлять комментарии. Войдите, пожалуйста.

Что обсуждают

Сейчас Вчера Неделя Месяц

- Тутор по BabylonJS — камера и освещение 1
- DoubleDomain и свобода 531
- Обработка аннотаций в процессе компиляции 2
- Ускоряем отладку и прототипирование мобильных QML-приложений на живом устройстве 8
- Ansible и Rails — гибкая замена Capistrano с сохранением знакомого комфорта 12

Самое читаемое

Сейчас Вчера Неделя Месяц

- Обнаружен ботнет, который исправляет уязвимости в зараженных им маршрутизаторах и сообщает об этом администратору 11k
- Удачи в цифровую эпоху! Или включите параноика и проверьте защиту своих данных 34k
- 40 ключевых концепций информационных технологий доступно и понятно 26k
- О безопасности UEFI, часть шестая 4k
- Разработка и публикация ассета в Unity Asset Store 2k

Лучшее на Geektimes

- Pokemon Company закрыла фанатскую вечеринку и требует с организатора $4000 (31)
- Люди на орбите Марса (1)
- В сеть попали данные пользователей и исходные коды краудфандинга Patreon (1)
- Почему поиск обитаемых планет так сложен (3)
- Как исправить осанку с помощью Kickstarter'а: подборка интересных «неврологических» проектов (2)

Все публикации Популярные хабы Компании

Лучшее на Мегамозге

- Как замена кнопки помогла компании увеличить выручку на $300 млн в год
- Национальная система платежных карт опубликовала тарифы и правила (3)
- «Вымпелком» обещает вскоре определиться с продажей сотовых вышек (1)
- 68% «Ростелекома» могут стать государственными

Все публикации Популярные хабы Стартапы

Вакансии на «Моём круге»

- Ruby-разработчик (Санкт-Петербург)
- Тестировщик (QA) (Санкт-Петербург)
- UX / UI Designer (Санкт-Петербург)
- .NET Developer (Санкт-Петербург)
- Java / Javascript разработчик (front/backend) (Санкт-Петербург)
- **Experienced PHP Backend Developer (New York Company)** (Санкт-Петербург)
- Backend программист (PHP/MySQL) (Санкт-Петербург)
- Разработчик для мобильных устройств (Санкт-Петербург)
- Senior PHP Developer (Санкт-Петербург)
- Senior Javascript Developer (Санкт-Петербург)

разместить вакансию все вакансии

Заказы на «Фрилансим»

- Скопировать сайт + внести правки
- Тестирование Web и iOS приложения
- Написать полноценное тз
- Нарисовать статичную инфографику медицинской тематики
- Motion design
- Изменить лицензию плагина для 3D Maya на языке программирования MEL
- Проведение PR действий на YouTube
- Простой монтаж видео
- Устранение багов, доработка сайта
- Запустить github.com/alongubkin/phonertc на XCode 6.1

разместить заказ все заказы