@ID_Daemon
Пользователь карма рейтинг

Профиль | 33 Публикации | 98 Комментарии | 7 Избранное | 151 Подписчики

6 августа 2015 в 19:40

О сколько нам открытий чудных готовит Office Microsoft

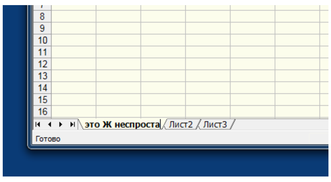
Реверс-инжиниринг*, Assembler*

```
^Alt^Ctrl+I^Sh^Alt^Ctrl+g^Sh^Alt^Ctrl+G^Sh^Alt^Ctrl+w^Sh^Alt^Ctrl+W^Sh
+^Alt+^Ctrl+k^Sh+^Alt+^Ctrl+K^Sh+^Alt+^Ctrl+h^Sh+^Alt+^Ctrl+H^Sh+^Alt+^Ctrl+p^Sh
+^Alt+^Ctrl+P^Alt+Escape^Sh+^Ctrl+^Alt+TabSpaceSh+F1^Sh+^Alt+^Ctrl+c^Sh+^Alt+^Ct
rl+CCtrl+/Ctrl+?F1a..z0..9A..ZF6Sh+F6Ctrl+F1Default^Scrl+Up^Scrl+Down^Scrl+End^S
crl+Home^Scrl+PageUp^Scrl+PageDown^Scrl+Ctrl+Up^Scrl+Ctrl+Down^Scrl+Ctrl+End^Scr
l+Ctrl+HomeCtrl+PageUpCtrl+PageDown^Scrl+Left^Scrl+Right^Scrl+Ctrl+Left^Scrl+Ctr
l+Right^Sh+Cancel^Alt+^Sh+EscapeUpDownLeftRightPageUpPageDownCtrl+PageUpCtrl+Pag
eDown^Sh+Cancel^Alt+^Sh+Escape^Sh+^Alt+^Ctrl+Return^Sh+^Alt+^Ctrl+Execute^Sh+^Al
t+^Ctrl+Escape^Sh+^Alt+^Ctrl+Cancel^Sh+^Alt+^Ctrl+Tab:+?Ctrl+:Ctrl+:Ctrl+^Ctrl+^
F2DefaultCtrl+ACtrl+Sh+BSpace^Sh+Return^Sh+ExecuteCtrl+ReturnCtrl+ExecuteCtrl+Sh
+ReturnCtrl+Sh+ExecuteBackspaceDeleteCtrl+:Ctrl+:Ctrl+^Ctrl+^Ctrl+2Ctrl+xCtrl+vC
trl+cCtrl+uCtrl+bCtrl+iCtrl+Delete.Alt+ReturnAlt+ExecuteAlt+=F2^Sh+^Alt+TabCtrl+
Sh+^Alt+Tab^Sh+^Ctrl+Cancel^Alt+EscapeF4Ctrl+Alt+TabCtrl+Alt+Sh+TabAlt+'Ctrl+Sh+
Alt+ReturnCtrl+Sh+Alt+ExecuteCtrl+Alt+ReturnCtrl+Alt+ExecuteSh+Alt+ReturnSh+Alt+
ExecuteAlt+Sh+LeftAlt+Sh+RightCtrl+9Ctrl+Sh+9Ctrl+0Ctrl+Sh+0Ctrl+7Ctrl+8^Sh+^Ctr
l+SpaceCtrl+5Ctrl+4Ctrl+3Ctrl+2Ctrl+1DummyCtrl+Alt+F7DummyShF10DefaultF3F4F5Bac
kspaceDeleteSh+F1Sh+SpaceCtrl+GCtrl+uCtrl+hCtrl+jCtrl+;Ctrl+'Ctrl+zCtrl+xCtrl+vC
trl+cF2Sh+ReturnAlt+=F1Alt+^Sh+F5^ShF7F12Sh+F12Ctrl+9Ctrl+Sh+9Ctrl+0Ctrl+Sh+0C
trl+8Ctrl+FDummyCtrl+Alt+F7DefaultBackspaceDeleteSh+F1TabSh+Tab^Sh+Ctrl+ReturnSh
+SpaceUpDownLeftRightHomeEndPageUpPageDownCtrl+RightCtrl+LeftCtrl+UpCtrl+DownCtr
```

По сообщениям в комментариях к статье про блокнот, во всех версиях Microsoft Excel, начиная по крайней мере с '97 и до самых новых, в имени листа не всегда можно ввести большую букву Ж. Данная проблема обсуждается в сети уже давно, например на этом форуме забавно наблюдать, как некоторые утверждают, что у них проблемы нет, а у других есть, но не всегда, и никто не понимает, почему так. На первый взгляд можно подумать, что это просто недоработка программистов: они хотели не дать пользователю ввести символ ':', и просто не подумали о том, что Ж находится на той же кнопке.

На деле оказалось всё гораздо хуже. Описать нормальными словами то, что происходит в excel, когда вы просто нажимаете кнопку 'Ж', практически невозможно. Поэтому я попытаюсь обрисовать в целом процесс исследования, сократив его где возможно, и не слишком перегружая статью ассемблерным кодом. В итоге мы узнаем, почему получается так, что не любые символы можно ввести, и как это можно исправить.

С чего начать? Поэкспериментируем немного. Оказывается, что иногда ввести Ж в название листа всё-таки можно, причём если уж один раз это сработало, то её можно будет вводить сколько угодно и где угодно, пока не закроешь Excel. А если не получилось, то как ни старайся, ввести эту букву уже не получится никак. Выяснить, почему так происходит, пока не удаётся. Известно одно: скопипастить её можно всегда.



Ну хорошо. Раз ошибка имеется где-то в коде проверки символов, попробуем найти её через действительно запрещённый символ ':'. Копипастим его в название листа, нажимаем Enter, и получаем ошибку, так такие символы не допускаются. Прервём на этом месте выполнение программы. Мы попадаем куда-то в дебри системных вызовов, во главе с user32.dll, общая вложенность 22 в глубину стека. Начинаем искать, как мы сюда попали. Поднявшись примерно на 15 уровней вверх, обнаруживаем следующий код:

```
0000000140395D97  cmp   bx,3Ah ;        ':'
0000000140395D9B  je    0000000140395DDC
0000000140395D9D  cmp   bx,5Ch ;        '\'
0000000140395DA1  je    0000000140395DDC
0000000140395DA3  cmp   bx,2Ah ;        '*'
0000000140395DA7  je    0000000140395DDC
0000000140395DA9  cmp   bx,3Fh ;        '?'
0000000140395DAD  je    0000000140395DDC
0000000140395DAF  cmp   bx,2Fh ;        '/'
0000000140395DB3  je    0000000140395DDC
0000000140395DB5  cmp   bx,5Bh ;        '['
0000000140395DB9  je    0000000140395DDC
0000000140395DBB  cmp   bx,5Dh ;        ']'
0000000140395DBF  je    0000000140395DDC
```

Очевидно, здесь в уже набранной строке проверяется наличие всех запрещённых в имени листа символов. Конечно, как мы уже знаем, до этого места буква 'Ж' не доходит, она срубается ещё при нажатии самой клавиши, но можно предположить, что там анализ происходит аналогично. Поищем подобный код в программе. Похожих мест оказывается несколько, и одно из них как раз срабатывает при нажатии клавиши, когда мы находимся в редактировании имени листа:

```
0000000140395D6  cmp   bx,3Ah ;        ':'
0000000140395DA  je    000000014039547D
00000001403953E0  cmp   bx,5Ch ;        '\'
00000001403953E4  je    000000014039547D
00000001403953EA  cmp   bx,2Ah ;        '*'
00000001403953EE  je    000000014039547D
00000001403953F4  cmp   bx,bp
00000001403953F7  je    000000014039547D
00000001403953FD  cmp   bx,si
0000000140395400  je    000000014039547D
0000000140395402  cmp   bx,5Bh ;        '['
0000000140395406  je    000000014039547D
0000000140395408  cmp   bx,5Dh ;        ']'
000000014039540C  je    000000014039547D
```

Пока всё идёт неплохо. При нажатии проверяются те же самые символы. Код немного отличается, ну уж так видно сработал компилятор. Пора выяснить, почему же не вводится буква 'Ж'. Нажимаем её, и тут нас ждёт сюрприз: точка останова не срабатывает! Как же так? Опять приходится возвращаться выше по стеку вызовов. В эту подпрограмму мы попадаем отсюда:

```
000000013FD08C12  bt    ecx,0Eh
000000013FD08C16  jae   000000013FD08C23
000000013FD08C18  and   ecx,3FFFh
000000013FD08C1E  call  qword ptr [rdi+8]
000000013FD08C21  jmp   000000013FD08C26
000000013FD08C23  call  qword ptr [rdi+8]
000000013FD08C26  xor   ecx,ecx
000000013FD08C28  mov   edi,eax
```

То есть вызывается подпрограмма по адресу, который содержится в регистре rdi плюс 8. Вот только оказывается, что при нажатии большинства клавиш там будет адрес той самой «правильной» подпрограммы, которую мы видели выше. А если мы нажимаем Ж с шифтом (чтобы она была большая), в этом месте оказывается совсем другая подпрограмма. Вот она:

```
000000013FB0A454 mov     eax,1
000000013FB0A459 ret
```

Это подпрограмма типа «давай, до свидания!» То есть она сразу завершается с кодом 1, ничего не анализируется, и нажатая клавиша никуда не сохраняется.

Мы обнаружили классический вызов по таблице. rdi — индексный регистр, его содержимое указывает на таблицу адресов, и в зависимости от его содержимого, вызывается та или иная подпрограмма. Вот эта таблица:

```
003981A80  0e 00 00 00 0e 00 00 00 0d 0d 00 00 00 00 00 00 54 55 39 40 01 00 00 00 00
003981A98  00 00 b3 03 00 00 00 00 2b 2b 00 00 00 00 01 00 54 55 39 40 01 00 00 00 00
003981AB0  00 00 64 03 00 00 00 00 1b 1b 00 00 00 00 25 00 b4 58 39 40 01 00 00 00 00
003981AC8  00 00 b3 03 00 00 00 00 03 03 00 00 00 00 a0 f0 b4 58 39 40 01 00 00 00 00
003981AE0  00 00 78 03 00 00 00 00 09 09 00 00 00 00 00 00 54 55 39 40 01 00 00 00 00
003981AF8  00 00 b3 03 00 00 00 00 ba ba 04 00 2c 00 00 00 54 a4 b0 3f 01 00 00 00 00
003981B10  00 00 78 03 00 00 00 00 38 38 04 00 2c 00 00 00 54 a4 b0 3f 01 00 00 00 00
003981B28  00 00 b3 03 00 00 00 00 bf bf 04 00 2c 00 00 00 54 a4 b0 3f 01 00 00 00 00
003981B40  00 00 97 03 00 00 00 00 ba ba 08 00 2c 00 00 00 54 a4 b0 3f 01 00 00 00 00
003981B58  00 00 b3 03 00 00 00 00 ba ba 0c 00 2c 00 00 00 54 a4 b0 3f 01 00 00 00 00
003981B70  00 00 97 03 00 00 00 00 de de 08 00 2c 00 00 00 54 a4 b0 3f 01 00 00 00 00
003981B88  00 00 b3 03 00 00 00 00 de de 0c 00 2c 00 00 00 54 a4 b0 3f 01 00 00 00 00
003981BA0  00 00 98 03 00 00 00 00 71 71 00 00 2c 00 00 00 54 a4 b0 3f 01 00 00 00 00
003981BB8  00 00 b3 03 00 00 00 00 00 ff 00 00 00 00 00 00 0C 53 39 40 01 00 00 00 00
```

Анализируя код, который с ней работает, удалось выяснить следующее: в начале таблицы — число строк (0Eh = 14), правда почему-то 2 раза. Каждая строка — описание комбинации клавиш. Сначала диапазон скан-кодов (выделен зеленым), затем допустимые состояния shift, alt и Ctrl (биты 4,8,20 — синим), маска для них (красным), и в конце 64-битный адрес подпрограммы (желтым), которая выполняется при совпадении условий. Большая Ж в этой таблице находится в 6-й строчке.

```
00 ba ba 04 00 2c 00 00 00 54 a4 b0 3f 01 00 00 00 00
```

Скан-код 0BAh (VK_OEM_1), при нажатом шифте (4) приводит к вызову 000000013FB0A454 = «до свидания». Если же ни одна из комбинаций не проходит, то в последней строчке срабатывает код от 0 до FF, и вызывается n/n 000000014039530C, часть которой мы видели выше, где всё идёт по обычному плану, и символ попадает в имя листа.

Ну вот. Казалось бы, всё понятно. Программисты не учли раскладку, просто отсекли некоторые скан-коды и в результате ошибка. Теперь осталось только выяснить, как таблица заполняется. Нигде в файлах офиса её нет, значит она генерируется на этапе исполнения. Снова повторяется мутороное скитание по многовложенным кодам, которое я пропущу. В этом процессе конечно excel пришлось много раз перезапускать.

И вот, в очередной раз остановившись где-то в недрах mso.dll, я с удивлением вижу, что в этой таблице какие-то совсем другие числа! Так я наконец узнал страшную тайну microsoft excel.

Самое читаемое

Сейчас | Неделя | Месяц

Как мы боролись с парсерами 39
Маргарет Гамильтон: «Пацаны, я вас на Луну отправлю» 1
Порог вхождения в Angular 2 — теория и практика 97
6 секретов Bitbucket 10
Microsoft может интегрировать подсистему Linux в новый выпуск Windows 10 128
Использование кодовой базы проекта Chromium в качестве SDK для разработки кроссплатформенных приложений 3
История создания Chatto 2
Критическая уязвимость коммутаторов Cisco Nexus 3000 Series и 3500 Platform позволяет получить к ним удаленный доступ 3
Обзор физики в играх Sonic. Части 3 и 4: прыжки и вращение 3
Requests и Responses в CodeIgniter 4 8

```
003981A80  0d 00 00 00 0e 00 00 00 0d 0d 00 00 00 00 00 00 54 55 39 40 01 00 00 00 00
003981A98  00 00 b3 03 00 00 00 00 2b 2b 00 00 00 00 00 00 54 55 39 40 01 00 00 00 00
003981AB0  00 00 64 03 00 00 00 00 1b 1b 00 00 00 00 00 00 b4 58 39 40 01 00 00 00 00
003981AC8  00 00 b3 03 00 00 00 00 03 03 00 00 00 00 b8 00 b4 58 39 40 01 00 00 00 00
003981AE0  00 00 78 03 00 00 00 00 09 09 00 00 00 00 00 00 54 55 39 40 01 00 00 00 00
003981AF8  00 00 b3 03 00 00 00 00 36 36 04 00 2c 00 00 00 54 a4 b0 3f 01 00 00 00 00
003981B10  00 00 78 03 00 00 00 00 38 38 04 00 2c 00 00 00 54 a4 b0 3f 01 00 00 00 00
003981B28  00 00 b3 03 00 00 00 00 37 37 04 00 2c 00 00 00 54 a4 b0 3f 01 00 00 00 00
003981B40  00 00 97 03 00 00 00 00 34 34 0c 00 2c 00 00 00 54 a4 b0 3f 01 00 00 00 00
003981B58  00 00 b3 03 00 00 00 00 36 36 0c 00 2c 00 00 00 54 a4 b0 3f 01 00 00 00 00
003981B70  00 00 97 03 00 00 00 00 32 32 0c 00 2c 00 00 00 54 a4 b0 3f 01 00 00 00 00
003981B88  00 00 b3 03 00 00 00 00 71 71 00 00 2c 00 00 00 54 a4 b0 3f 01 00 00 00 00
003981BA0  00 00 98 03 00 00 00 00 00 ff 00 00 00 00 00 00 0c 53 39 40 01 00 00 00 00
```

Как видим, здесь теперь не 14, а 13 строк, и в середине другие скан-коды (выделены зеленым). А именно, в шестой строке, там где раньше была буква Ж, теперь shift-6. То есть то же самое двоеточие, только из русской раскладки. То же самое и с остальными клавишами. Вот теперь действительно стало всё понятно. Довольно быстро выяснилось, что таких таблиц тут не одна, а целых 43 штуки (для каждой области экрана, отдельно для основного поля, поля ввода формул и т.д.). И все они заполняются один раз, при первом нажатии клавиши, а заполнение зависит от выбранной **именно в этот момент** раскладки и больше никогда не меняется.

То есть если после запуска excel при первом нажатии любой клавиши, пусть даже стрелки вниз, выбрана английская раскладка, буквы 'Ж' вы больше не увидите. И наоборот, если раскладка была русская, больших 'Ж' будет сколько угодно, а запрещённым «назначается» shift-6, и, кстати, shift-7 тоже. И после этого назвать лист, скажем, «рога & копыта» уже не получится, хотя символ '&' вроде как разрешён.

Ещё раз напомню, что всё вышесказанное относится к вводу текста в названии листа, а для остальных областей экрана excel таблицы другие, некоторые по 40 и более строк. О том, что там может происходить при нажатии клавиш, лучше не думать.

Тем не менее, исправить эту ошибку можно, и даже проще, чем ожидалось. Оказалось, что исходным материалом для заполнения всех этих таблиц служит текстовая (!) таблица комбинаций клавиш, расположенная в файле XLINTL32.DLL, лежащего в одной из папок офиса. Его часть как раз изображена на КДПВ.

Вот так выглядит фрагмент, касающийся названия листа:

```
~Sh+~Alt+~Ctrl+Return~Sh+~Alt+~Ctrl+Execute~Sh+~Alt+~Ctrl+Escape~Sh+~Alt+~Ctrl+Cancel
~Sh+~Alt+~Ctrl+Tab:*?Ctrl+;Ctrl+:Ctrl+'Ctrl+"F2Default
```

Что же делает excel? Он разбирает части этой строки и делает из неё ту самую таблицу, подбирая такие скан-коды, которые приводили бы к вводу нужных символов, с учётом раскладки. Представляете? Он анализирует текстовое представление, чтобы сделать из него таблицу скан-кодов, чтобы потом, при нажатии клавиш, сравнивать полученный код с каждой строкой и вызывать соответствующую процедуру. Заполнять все 43 таблицы при каждом нажатии естественно не годится. Поэтому это делается один раз. Так что программисты не забыли про раскладки, а провели с ними огромную работу. Только одно они не учли — во время работы excel их можно переключать.

Кстати, теперь ясно, почему в русском варианте не 14 строк, а 13. Одна из запрещённых комбинаций Ctrl+' невозможна в русской раскладке, потому что апостроф в ней в принципе нет, поэтому и скан-кода для него не находится.

Вернёмся однако к ошибке. В середине текста видны подряд те самые 3 символа :*? для запрещения. Чтобы всё исправить, достаточно в файле XLINTL32.DLL заменить эти 3 символа :*? на 3 звёздочки, потому что звёздочка на обоих раскладках в одном месте. Это можно сделать с помощью любого двоичного редактора или даже FAR, т.к. он позволяет менять текстовую часть двоичного файла по F4, при этом не испортив его.

После этого excel перестанет отбрасывать нужные скан-коды, и можно будет при любой раскладке вводить большую Ж, '&', '^' и запятую, при этом действительно запрещённые ':' и '?' всё равно не пройдут 2 проверки, которые описаны в начале статьи. Короче всё будет хорошо. И в следующий раз, когда будете набирать какой-нибудь текст в ворде или другом приложении офиса, постарайтесь не думать о том, что там происходит внутри.

реверсинг, excel, древние баги

+225 | 97,4k | 314

@ID_Daemon карма 343,0 рейтинг 0,0

Похожие публикации

+16 10 правил хорошего тона при описании багов
42,9k | 269 | 65

+8 Создание Excel файла из селекта с параметрами при помощи чистого PL/SQL, как альтернатива Oracle*Reports
11,4k | 101 | 11

+5 Букмарклеты в Internet Explorer 11: формат хранения, лимиты и негласные правила, коварный баг
3,3k | 14 | 6

Комментарии (87)

sim-dev 6 августа 2015 в 19:57 +28

Я подспудно всегда чувствовал, что если вопросами искусственного интеллекта займется Microsoft, восстания машин нам не избежать. Не по злому умыслу, а…

edogs 6 августа 2015 в 20:55 +21

И оно провалится при команде: «1. Убить все[вы; ивши[» :)

Arkham 10 августа 2015 в 10:09 0

А что за прикол с "[вы; ивши["?

Ubuntovod 10 августа 2015 в 10:51 0

«Убить всех выживших» — на традиционной 101-кнопочной клавиатуре символы "[" и «х» находятся на одной клавише. Аналогично ";" и «ж». Такое может вылезти при фильтрации по коду кнопки, а не по символу.

Arkham 11 августа 2015 в 03:07 *(комментарий был изменён)* 0

Спасибо, а то подумал может ещё какая пасхалка офиса :)

Eternalko 7 августа 2015 в 03:46 *(комментарий был изменён)* +7

Когда винда наконец-то увидит и распознает мой сетевой принто-сканер, тогда я начну за это опасаться.
Я буду держать всех в курсе и скажу если «началось».

Godless 7 августа 2015 в 10:01 0

Да пусть хотя бы локальный Olivetti PR2E увидит…

Muzzy0 12 августа 2015 в 08:54 +1

Да ладно… Если у них получится искуственный интеллект, то он пройдёт первую же проверку — не будет работать.
Что он, дурной?

ivlis 6 августа 2015 в 20:01 *(комментарий был изменён)* +2

MSO то ладно, а вот представьте сколько таких багов во подсистемах безопасности и шифрования.

Интересно зачем некорректный символ проверяется 3 раза? И ну и по сканкоду, это сильно конечно.

MaximChistov 6 августа 2015 в 20:12 +14

> Интересно зачем некорректный символ проверяется 3 раза?

Разные индусы эти куски кода писали, у каждого было тз такие символы не пропускать, а вмержились они все одновременно, как вариант,

MacIn 7 августа 2015 в 00:53 +2

> MSO то ладно, а вот представьте сколько таких багов во подсистемах безопасности и шифрования.

Это же совсем разные команды. Прикладники — сами по себе, «ядерщики» — сами и т.д.

greenkaktus 7 августа 2015 в 17:36 0

let char = 'Ж'

// если не присвоилось, то
if !char {

}

SomebodyElse 6 августа 2015 в 20:08 +11

И эти люди ~~запрещают мне ковыряться в носу~~ учать меня как нужно писать программы :)

goodbear 6 августа 2015 в 21:53 +56

Когда смотришь вопросы на собеседовании — не иначе гениев набирают.
Когда смотришь на код — и куда они всех этих гениев дели…

sim-dev 7 августа 2015 в 10:36 +4

Ну как куда? В отборщиков гениев — в менеджеры.

matiouchkine 7 августа 2015 в 10:52 0

У меня есть небезосновательная гипотеза, что лид продукта в то время, когда писался этот код, спустя некоторое время основал FogCreek и StackOverflow. И вообще, кажется, мужик довольно неглупый.

bitterman 7 августа 2015 в 11:49 0

а как связана разработка VBA с разработкой всего экселя? Или VBA — отдельный продукт, а эксель — отдельный?

Что обсуждают

Сейчас | Вчера | Неделя

Пробрасываем роуты Angular 2 через роутер Laravel 5 18

Маргарет Гамильтон: «Пацаны, я вас на Луну отправлю» 1

Разложение матрицы афинного преобразования 2

Порог вхождения в Angular 2 — теория и практика 97

Как мы боролись с парсерами 39

matiouchkine 7 августа 2015 в 11:58 0

А как связан ваш вопрос с моим комментарием?

bitterman 7 августа 2015 в 11:59 0

наверное, тем, что Джоел Спольски отвечал всё-таки за VBA?

matiouchkine 7 августа 2015 в 13:15 0

Да вроде нет.

> I left the company in 1994, assuming Bill had completely forgotten me, until I noticed a short interview with Bill Gates in the Wall Street Journal, in which he mentioned, almost in passing, something along the lines of how hard it was to recruit, say, ★a good program manager for Excel★. They don't just grow on trees, or something.
— www.joelonsoftware.com/items/2006/06/16.html

bitterman 7 августа 2015 в 13:46 0

1. как соотносится «лид продукта» и program manager? Сколько менеджеров работают над одним продуктом и как из них называется ответственный за продукт в целом? Который «лид продукта»?

2. в этой же статье отчётливо написано, что будучи «program manager» он разрабатывал подсистему VBA для экселя. VBA < Excel < Microsoft Office < Microsoft, при этом код, отвечающий за букву Ж во страницах экселя никоим образом с Джоэлом не связан.

wizardsd 6 августа 2015 в 23:02 +10

Одни примеры с MSDN чего стоят. Ни RAII, ни exception-safety нет, а сколько коду с этих примеров скопировано в продакшн…

Wedmer 6 августа 2015 в 23:30 +14

… внутри самого MicroSoft.

ishevchuk 6 августа 2015 в 20:10 +12

Читая статьи о реверс-инжиниринге Майкрософтских программ иногда кажется, что не видя исходный код такое раскопать невозможно...(а это говорит о мастерстве автора :)).

Или может намного проще понять в чем проблема НЕ имея исходный код, чем бегать по сотням классов/функций в IDE и читать код?

oYASo 7 августа 2015 в 00:46 +10

Некоторый код проще понять в ассемблере, серьезно.

m08pvv 7 августа 2015 в 10:58 +1

Особенно если компилятор проделал огромную работу по оптимизации «индусского» кода.

iago 7 августа 2015 в 16:15 0

Какой-нибудь крэш одинаково легко. А такую работу, как проделал автор, конечно проще было бы проделать через IDE с сорцами. Он вообще большой умница!

divanikus 6 августа 2015 в 20:26 0

В 2016-м превью вроде бы проблемы нет. Вводиться в любой раскладке. Есть подозрения что могли пофиксить с переходом xslx.

divanikus 6 августа 2015 в 20:28 +4

Ха, а в постановке с форума действительно не работает:
Эта старинная хохма достигается следующим макаром:
— в ENG-раскладке попытаться ввести двоеточие (:)
— сказать «ой» (ибо не получится)
— переключиться в RUS-раскладку
— попытаться ввести «Ж»
— опять — «ой»

ID_Daemon 6 августа 2015 в 20:33 +8

Читайте внимательно статью. Всё дело в том, какая раскладка у вас будет выбрана **в момент** первого нажатия на клавишу.

divanikus 6 августа 2015 в 20:38 +1

Да, признаю, так не работает.

ID_Daemon 6 августа 2015 в 20:39 +27

В общем, спасибо за информацию, что в 2016-м баг по-прежнему есть ;)

domix32 7 августа 2015 в 11:50 +10

Ловите человека из будущего!

xapienz 4 января 2016 в 15:02 0

Привет из 2016!

JC_Piligrim 9 августа 2015 в 22:55 +3

Спортивный дайджест с собой взял? :)

dax 6 августа 2015 в 20:36 0

После таких статей начинаешь понимать, почему Микрософт не любит open source. Иной исходник может неслабо так подмочить репутацию.

creker 6 августа 2015 в 21:38 +13

Я просто оставлю это здесь
github.com/dotnet
github.com/aspnet
github.com/Microsoft
Первое, что по памяти нашел.

zed91 7 августа 2015 в 07:08 +3

Каждый опенсорсник считает нужным пнуть мс за легаси код, этого не исправить

505abc 7 августа 2015 в 08:37 −3

1) Исправить
2) Не пускать в продакшен такой код. Не пофиксить баг и просто забить, я ума не приложу, как такое можно сделать. Хотя справидливости ради хочеться отметить, что не плохо бы знать частоту появления этого бага. В этой статье эти данный представлены расплывчато.

Ubuntovod 7 августа 2015 в 08:51 +1

1) Исправить-то можно, а нужно ли? Когда софт работает идеально — нет повода «пинать».
2) Ошибка не критическая, скорее просто неприятная. К тому же на продажи исправление такого бага не влияет совершенно никак — а соответственно нет повода вкладывать в это дело ни цента. Microsoft может себе позволить не исправлять ошибку, встречающуюся «раз на миллион».

xel 7 августа 2015 в 11:13 +1

Вы когда пишете код/делаете концепт предусматриваете rtl-языки?
По опыту работы американцы где-то также относятся к другим локалям, европейцы -к другим таймзонам: вроде понятно, что оно существует и даже будет где-то использоваться, но внимания к этому по остаточному принципу.
Вспомните баг в Netscape Navigator — стоило в javascript использовать букву «я», как браузер сходил с ума. Ну а что? Удобно же 255-й код использовать для своих нужд, он всё равно «где-то в неиспользуемой части» ascii-таблицы.

Archon 7 августа 2015 в 16:00 *(комментарий был изменён)* +2

Среднему американскому программисту вообще не приходит в голову, что у клавиатуры может переключаться раскладка. Как раз этим, например, можно объяснить решение Эпла включить всем по умолчанию отдельную *раскладку* для смайликов. Для американца этот интерфейс прост и понятен: нажали кнопку, включился режим смайликов, нажали ещё раз, он выключился. А про народы, вынужденные писать в двух раскладках, никто и не подумал.

Gendalph 8 августа 2015 в 12:58 +2

Вам и M$ передают привет страны с тремя раскладками, а также отдельные их жители с четырьмя.

Muzzy0 12 августа 2015 в 08:58 0

Вы меня опередили про 3 раскладки

fuCtor 7 августа 2015 в 19:31 +5

> По опыту работы американцы где-то также относятся к другим локалям

Где-то видел, на одном англоязычном форуме, перевод в UTF-8 путем приписывания 0x00 спереди, мол и так работает же.

Muzzy0 12 августа 2015 в 08:58 0

> Вы когда пишете код/делаете концепт предусматриваете rtl-языки?

Вы даже не представляете, на какую больную мозоль наступили…

ploop 6 августа 2015 в 20:42 +3

А у майкрософта есть какой-нибудь багтрекер или нечто подобное?
Автору спасибо, люблю такие детективы. Главное доступно и интересно написано.

PastorGL 6 августа 2015 в 21:52 0

Есть suggestion box, excel.uservoice.com

Багтрекер, о котором почему-то никто не знает, хотя он вполне публичный, много лет был на MS connect, но его сейчас постепенно прикрывают.

gotch 6 августа 2015 в 22:36 +6

А знаете почему?

Muxto 7 августа 2015 в 09:52 +13

Почему?

gotch 10 августа 2015 в 15:16 0

Вероятно потому, что ваше мнение и ваши усилия в направлении улучшения продукта больше не нужны.

MacIn 10 августа 2015 в 15:36 0

Это неверно.
Запросы на форумах TechNet от специалистов, а не конечных пользователей (у меня тут зависло, почините) обрабатываются пасущимися там MSFT.
У меня смешанный опыт: иногда получалось достучаться, иногда нет.

gotch 11 августа 2015 в 10:44 0

Есть ли у вас примеры, что в форуме получен ответ или решение, недоступное простому обывателю, внимательно читающему библиотеку Technet?

MacIn 11 августа 2015 в 14:31 *(комментарий был изменён)* +1

Да. Я дважды получал помощь: один раз оказалось, что в SDK не включены кое-какие .h файлы, я указал на это, они извинились за ошибку и включили их в следующий выпуск.
В другой раз мне нужно было решить одну хитрую задачу с WinJobs, мне подсказали нестандартный трюк. Насколько я помню, это были как раз MSFT участники.
А вот ответа по поводу OLE интерфейсов в RichEdit компоненте ответа получить не смог нигде. Ни через багтрекер, ни через форумы технет, ни через твиттер их техподдержки.

gotch 13 августа 2015 в 18:29 0

Дайте ссылки, интересно. Все ваши сообщения есть в вашем профиле, activity.

BlackRaven86 7 августа 2015 в 03:28 +11

> Тем не менее, исправить эту ошибку можно, и даже проще, чем ожидалось.

Кстати, позавчера LibreOffice 5 вышел… На всякий случай, вдруг кому :)

idiv 7 августа 2015 в 08:42 0

Это похоже на ошибку Автокада версии до 2009 вроде. Там нельзя было использовать в названии слоя букву Б. К сожалению, но помню, можно ли было сделать копировать-вставить. Причем у них в багах это висело лет 7-8, пока исправили (это важно, так как каждые 3 года они меняют формат файла и в целом вносят много изменений в код, а тут 3 поколения не исправляли).

ef_end_y 7 августа 2015 в 09:54 +6

MS видимо не понимают, что название состоит из символов, а не нажатых клавиш. Надо проверять само название на валидность, а оно может быть сформировано как с клавиатуры, как еще кучей разных способов, включая ситуацию, когда кто-то залезает в исходник документа и там поменяет символ в обход excel.

Представляете сколько избыточного и тупого кода в других частях системы? Например, проверка клавиш при сохранении файла. И эта проверка может быть своя в каждом продукте. Понятно куда деваются гигабайты на системе виндовс

ploop 7 августа 2015 в 10:07 *(комментарий был изменён)* +1

> Надо проверять само название на валидность

Тоже спорный вопрос. Зачем ставить ограничение на **название**? Ладно в ФС, и то там минимум ограничений, слеш и какое-нибудь двоеточие, остальное от лукавого, но внутри документа???

khim 7 августа 2015 в 10:21 −1

Вы когда-нибудь работали с Excel или вообще никогда? Название листа там вполне может фигурировать как часть формулы! Если всякие двоеточия будут допустимы в имени — как это всё будет работать, по вашему?

ploop 7 августа 2015 в 10:52 +3

> Вы когда-нибудь работали с Excel или вообще никогда?

По минимуму.

> Название листа там вполне может фигурировать как часть формулы!

Почему бы в синтаксисе не предусмотреть квотирование? Пример из комментa ниже "[Файл]Лист!$A$8" может выглядеть вполне как "[Файл]'Лист: *красненький*, \для Мариванны\'$A$8"
Эти проблемы давно решены.

el777 7 августа 2015 в 10:34 0

Потому что потом вам надо будет делать ссылку на ячейку вида "[Файл]Лист!$A$8". Если у вас будут левые символы, то ссылка не распарсится и что-то обязательно упадет в другом месте или посчитает неверный результат.

Собственно та же самая проблема, что и с названиями файлов — если они сами по себе, то без разницы, а если нужно использовать их где-то, то уже важно название.

DmitryAnatolich 16 августа 2015 в 00:27 0

Ой, да ну что за проблема! В T-SQL, значит, можно экранировать [Пробелы и прочую ересь] квадратными скобками в именах колонок, таблиц и прочего, а в Excel проблема так же распарсить?

khim 7 августа 2015 в 10:19 −4

MS всё прекрасно понимает. Если бы вы дочитали статью до конца, то увидели бы, что символы они **тоже** проверяют — чуть позже. А тут они хотели сделать «как лучше». Проблема в том, что две раскладки бывают только в ограниченном числе стран: там Греция, Россия, Израиль… если людей из этих стран в команде нет, то может не найтись никого, кто бы осознавал не только то, что в природе бывают раскладки, но и то, что в природе бывают люди, которые их во время работы регулярно меняют — и их довольно много…

P.S. Причём тут именно важно иметь людей именно из этих стран. Эмигранты часто смиряются с тем, что раскладка на клавиатуре не написана и пользуются всякими translit.ru, если им нужно вводить русский текст.

vlivyur 7 августа 2015 в 10:37 +6

Я б сказал что двух раскладок НЕ бывает в ограниченном числе стран, а остальные минимум с двумя и живут. Вот только MS из США, а там как раз она одна.

khim 7 августа 2015 в 11:28 −8

Опять этот великорусский шовинизм, LOL. «Остальные» — это хто? США и Европа (включая Турцию), Латинская Америка и Китай, Япония и Австралия — все они живут с одной раскладкой. Да-да, даже страны, где пишут иероглифами. Там ввод текста осуществляется совсем особым образом — но именно поэтому проверка скан-кодов никому не мешает. Так что, увы и ах, но это именно страны с двумя кодировками — исключение. Потому с ними и возникают проблемы то в Linux, то в Windows. А вовсе не потому, что софт только в Штатах разрабатывают.

iroln 7 августа 2015 в 12:42 *(комментарий был изменён)* +1

А это что?
en.wikipedia.org/wiki/German_keyboard_layout

> The German keyboard layout is a QWERTZ keyboard layout commonly used in Germany and Austria

khim 7 августа 2015 в 13:41 0

Ммм. Не понял вопроса. Это клавиатура, которой пользуются в германии. QWERTZ, да. QWERTY ни при этом, разумеется, не пользуют, раскладки не переключают, Office и другие подобные «сумасшедшие» программы их не напрягают. В чём проблема?

KReal 7 августа 2015 в 13:44 0

В немецкой раскладке прекрасно можно писать на английском, я гарантирую это.

EvilFox 7 августа 2015 в 13:48 *(комментарий был изменён)* −1

Там если начать ввод на обычной раскладке, а потом переключиться на эту дополнительную то тот знак что находится где «:» тоже нельзя будет ввести. Вряд ли только он вводится так же часто как Ж.

khim 7 августа 2015 в 14:07 0

Там нет никакой «основной» и «дополнительной». Все многочисленные символы, которые вы там видите вводятся с разными всякими AltGr'ами в основной раскладке и Office со своими трюками это прекрасно поддерживает.

EvilFox 7 августа 2015 в 13:44 *(комментарий был изменён)* +1

Как-то слишком самоуверено.

> Япония

Как вы быстро лишили японцев хираганы и катаканы. У них есть ввод каной. Поэтому в принципе ромадзи там могло и не быть. Просто ~~ромадзи навязали~~ IME очень наворочен и позволяет задать клавишу на переключение внутренней раскладки с кана на ромазди и обратно. У нас к слову тоже есть умельцы которые впихнули в одну раскладку и латиницу и кириллицу и сделали переключение по caps look (при этом не сломав верхний регистр по Shift, но сломав типичную проверку орфографии). В общем японцам тут чуть больше повезло. А так объективно их нельзя выписывать из списка двух-раскладочных. Сюда ещё можно добавить Корею. Как у китайцев я не знаю.
На память приходит что ещё свои раскладки есть у:

- Армении
- Грузии
- Тайланда

Про прочие кириллические — Украину, Беларусию, Казахстан и т. п. думаю нет смысла писать?

Одна раскладка только у стран где принята в основе латиница (доп знаки они набирают через AltGr) и то как видно в случае Германией есть исключение.

khim 7 августа 2015 в 14:05 −3

> Поэтому в принципе ромадзи там могло и не быть.

Не могло. Компьютеры японцы получили с Запада и хотя изначально они тоже были двураскладочниками, но латинница там была изначально. Каны не было — это да.

> Как вы быстро лишили японцев хираганы и катаканы.

Вот именно они и вводились на второй раскладке на какой-нибудь MSX. Но в современных системах (хоть Windows, хоть ChromeOS) японцы — однораскладочники.

> А так объективно их нельзя выписывать из списка двух-раскладочных. Сюда ещё можно добавить Корею. Как у китайцев я не знаю.

У всех трёх есть переключение режимов **внутри** одной раскладки. Собственно IME есть пошла из Китая — причём она в ранние времена была программно-аппаратным комплексом, на продаже которого поднялась одна небезизвестная компания. Microsoft свою версию разработал вроде бы независимо, но «по образу и подобию». И, опять-таки: поскольку всё это надстраивалось над QWERTY, то латинница там была в основе изначально.

У меня просто есть знакомый, занимавшийся разработкой ChromeOS. Так вот: **первой** страной, которая потребовала замораживаться с двумя раскладками была Россия. Греки и евреи были уже позже. А до этого они «окучили» большую часть мира, включая Японию, часть Латинскую Америки и Африки.

Возможно в какой-то другой, альтернативной, вселенной развитие компьютерной техники могло пойти иначе и там бы Япония и Китай попали бы в список «многораскладочников». Но, увы, на нашей планете этого не случилось и мы с вам попадаем в жалкое меньшинство о котором редко кто задумывается.

Я не говорю, что это хорошо — это просто неизбежно.

sup 10 августа 2015 в 19:54 0

Странам, набирающим символы через AltGr тоже можно свинью подложить, использовав в программе шорткаты Ctrl + Alt + something.
Мало кто из русских программистов знает, что таким образом можно заблокировать ввод некоторых символов туркам, полякам и т.д. 8)

EvilFox 7 августа 2015 в 12:43 0

Какое-то нелепое оправдание. Если продукт идёт на рынок других стран, он должен разрабатываться с учётом их особенностей и тестироваться в том числе в их условиях.

khim 7 августа 2015 в 13:48 0

Microsoft не раскрывает подробную статистику по продажам в разных странах, но есть основания считать, что вряд ли он получает в двураскладочных странах более нескольких процентов продаж.

Если продукт идёт на рынок стран, которые приносят вам 1% дохода, то вы выделите на адаптацию ресурсов ровно столько, сколько он заслуживает. Никто не будет **разрабатывать** продукт специально для такого рынка. Его доработают — как смогут.

Более того, если вам придётся жизнь людей с этого однопроцентного рынка сильно ухудшить из-за того, что какая-то фишка облегчит при этом жизнь 99% ваших потребителей — это нужно делать особо не задумываясь, так как улучшение продаж на, скажем, 10% на двадцатипроцентном рынке скомпенсирует вам возможные потери с ликвой. Простая математика.

Очевидно тестировщики этот баг не воспроизвели, а если и воспроизвели — то не обратили внимание.

EvilFox 7 августа 2015 в 14:25 +3

> Microsoft не раскрывает подробную статистику по продажам в разных странах

> есть основания считать что вряд ли он получает в двураскладочных странах более нескольких процентов продаж

Взаимоисключающие параграфы.
Ваши домыслы ничем не подкреплены.

khim 7 августа 2015 в 15:21 0

Если вы считаете, что единственные данные, на основании которых можно что-то оценивать — это бумажки, издаваемые Microsoft'ом, то мне вас жаль. Это для компании «рога и копыта», продающей два с половиной компьютера в день невозможно точно узнать сколько точно и кому она продала, но тут мы всё-таки про лидера рынка говорим! Есть данные об объёмах рынка, есть данные о распространённости пиратской продукции, есть данные о ценах в разных странах, наконец! Это вполне достаточно для того, чтобы примерно оценить объёмы и перспективность рынков.

muon 10 августа 2015 в 14:11 *(комментарий был изменён)* 0

И давно ли Microsoft стал лидером рынка? По Вашей ссылке он упоминается два раза — 5% в тексте и четвёртое место в графике. Давайте определимся, о рынке чего идёт речь.

А потом определимся, относить ли к «однораскладочным» странам Китай и Индию, с её месивом культур и национальностей.

Muzzy0 12 августа 2015 в 09:11 0

> Израиль…

Кто ещё шовинист…

Допустим, что нас тут в Израиле мало и поэтому с локализацией заморачиваются по минимуму — и не только в программном обеспечении. Но есть ещё двоюродные братья (арабы) коих куда больше. А для них актуально всё то же, что и для нас: своя раскладка клавиатуры, RTL. И даже больше: у них количество символов поболее, ибо есть всякие лигатуры и обязательные огласовки (в иврите огласовки опциональны, а реально ли вводить теамим — я даже не знаю).
Кроме того, у арабов даже цифры свои

Maccimo 7 августа 2015 в 11:19 0

В одной из древних, ещё до поглощения Adobe-ом, версий среды разработки Macromedia Flash было невозможно использовать букву «я» ни в коде скрипта, ни даже в комментариях.
Приходилось в таких случаях обходить баг при помощи escape-последовательность \377.

khim 7 августа 2015 в 11:35 0

Был такой редактор, одно время очень популярный: MultiEdit. В нём та же проблема была. Причём в DOS это никому не мешало (там символ с кодом 255 — это «неразрывный пробел», который мало кого волновал в те годы), а в Windows (где туда попала полезная буква «я») — это стало просто катастрофой.

VAE 7 августа 2015 в 12:55 −3

интересно, минусует один и тот же пользователь?

alexs0ff 7 августа 2015 в 13:06 +2

Еще один workAround — фиксится CapsLock+ж+CapsLock

toxicdream 7 августа 2015 в 22:09 0

Хорошо что все комментарии прочитал…
А то сейчас бы дубль был.

Dinir102 7 августа 2015 в 20:10 −3

Ещё заметил, что блочится "?". Так вот, берём *Английский* язык и пытаемся поставить знак **вопроса** (Shift+/). Вопросительный знак не ставится. Переключаемся на *Русскую* клавиатуру и пытаемся поставить **запятую** той-же комбинацией. **Поздравляю! Вы получили ещё одну нерабочую кнопку :D**

ID_Daemon 7 августа 2015 в 20:22 +3

Это уже есть в статье, и включено в исправление ошибки, описанное в последних 2 абзацах.

04.03.2016 • 3 отклика • **Цена договорная**

Landing Page для детской школы IT-образования

04.03.2016 • 7 откликов • **Цена договорная**

ERP для компании сервис-инженеров

04.03.2016 • 3 отклика

Зарегистрироваться **Разместить заказ**