МИНОБРНАУКИ РОССИИ

Федеральное государственное автономное образовательное учреждения высшего образования

«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт компьютерных технологий и информационной безопасности

Кафедра математического обеспечения и применения ЭВМ

  

**ЛАБОРАТОРНАЯ РАБОТА № 1**

по дисциплине

**«Операционные системы и системное программное обеспечение»**

на тему:

**«Исследование обмена сообщениями в ОС Windows»**

*Вариант № 2*

Выполнил:

Студент группы

КТбо2-8

Нестеренко П. А.

Проверила:

ассистент кафедры

МОП ЭВМ

Альминене Т. А.

Оценка

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г.

Таганрог 2020

# **1 ЦЕЛЬ РАБОТЫ**

**1.1 Дидактическая цель работы**

Целью данной работы является ознакомление с механизмом сообщений в операционной системе Windows, способами генерации и обработки сообщений, а также приобретение навыков исследования потока сообщений с помощью программы Spy++.

**1.2 Практическая цель работы**

Выполнение работы заключается в получении протокола сообщений, возникающих в ситуации, определенной вариантом задания. Для получения протокола следует настроить Spy++ подходящим образом и осуществить заданные действия. Можно не отслеживать сообщения, адресованные посторонним окнам, сообщения IME и сообщения о перемещении мыши. Все оставшиеся сообщения следует прокомментировать и понимать, в ответ на какое действие пользователя появляется то или иное сообщение (или группа сообщений).  
 В тех случаях, когда протокол содержит серии однотипных сообщений, можно оставить только два-три первых из серии, а остальные заменить строкой из точек.Вместе с сообщениями для окна верхнего уровня следует отслеживать также сообщения для всех его дочерних окон.  
 Информацию о сообщениях Windows, их смысле и параметрах, можно найти в справочных данных системы программирования (Visual Studio, Delphi, C++ Builder) или, если есть возможность, в базе данных MSDN (Microsoft Developers Network).

.

# **2 ВАРИАНТ ЗАДАНИЯ**

**Задание.** Получить и прокомментировать протокол сообщений, посылаемых окну редактора Notepad («Блокнот») при выполнении копирования/вставки фрагмента текста через буфер обмена.

Пример выполнялся в системе Windows 10. Для других версий Windows поток сообщений может несколько отличаться.

# **3 ХОД РАБОТЫ**

Сначала запустим **Notepad**, создадим новый файл и оставим пустым. Затем запустим Spy++ и выполним команду **Spy+Log Messages**. Откроется диалоговое окно **Message Options**. На вкладке Windows оттащим **Finder Tool** на окно Notepad. При этом можно обнаружить, что Notepad содержит, как минимум, два окна: основное окно программы и вложенное в него (дочернее) окно с панелью редактора. На самом деле, имеется еще окно строки состояния, которое может быть видимым или нет, в зависимости от настройки Notepad.

Выберем главное окно, но включим переключатель **Child Windows**, чтобы получать сообщения для всех окон Notepad. На вкладке **Messages** для большей наглядности примера оставим все сообщения.

Полезно перейти теперь в окно списка окон (через команду меню Spy++ **Windows**), на всякий случай обновить список (командой контекстного меню **Refresh**) и найти в списке окна Notepad. Это можно сделать либо просмотром всего списка, либо с помощью команды **Search+Find Window**. В нашем случае список содержит главное окно Notepad с хэндлом , дочернее окно редактирования (хэндл 0001572) и окно строки состояния (00010574). Разумеется, значения хэндлов при других запусках Notepad могут быть другими.

Вернемся в окно протокола сообщений. Там уже могли появиться записи в ходе выполнения команд **Refresh** и **Find Window**. Очистим протокол, удалив все сообщения командой **Messages+Clear Log**.

Теперь скопируем строку “Привет мир” в панель редактора, используя сочетание клавиш **Ctrl+v**. В данном случае в протокол успело попасть 79 сообщений.

|  |  |
| --- | --- |
| <00001> 001B0572 P | WM\_KEYDOWN nVirtKey:VK\_CONTROL cRepeat:1 ScanCode:1D fExtended:0 fAltDown:0 fRepeat:0 fUp:0 |
| <00002> 001B0572 P | WM\_KEYDOWN nVirtKey:'V' cRepeat:1 ScanCode:2F fExtended:0 fAltDown:0 fRepeat:0 fUp:0 |
| <00003> 009705E0 S | WM\_INITMENU hmenuInit:00E7066F |
| <00004> 001B0572 S | EM\_GETSEL lpdwStart:0000002486DBEF10 lpdwEnd:0000002486DBEF14 |
| <00005> 001B0572 R | EM\_GETSEL wStart:0 wEnd:0 lpdwStart:86DBEF10 (0) lpdwEnd:86DBEF14 (0) |
| <00006> 001B0572 S | WM\_GETTEXTLENGTH |
| <00007> 001B0572 R | WM\_GETTEXTLENGTH cch:0 |
| <00008> 001B0572 S | EM\_CANUNDO |
| <00009> 001B0572 R | EM\_CANUNDO fCanUndo:True |
| <00010> 009705E0 R | WM\_INITMENU |
| <00011> 009705E0 S | WM\_INITMENUPOPUP hmenuPopup:0D0D04A9 uPos:1 fSystemMenu:False |
| <00012> 009705E0 R | WM\_INITMENUPOPUP |
| <00013> 009705E0 S | WM\_COMMAND wNotifyCode:1 (sent from an accelerator) wID:770 |
| <00014> 009705E0 R | WM\_COMMAND |
| <00015> 009705E0 S | WM\_UNINITMENUPOPUP |
| <00016> 009705E0 R | WM\_UNINITMENUPOPUP |
| <00017> 001B0572 P | WM\_PASTE |
| <00018> 009705E0 S | WM\_NOTIFY idCtrl:15 pnmh:0000002486DBF5F8 |
| <00019> 009705E0 R | WM\_NOTIFY |
| <00020> 001B0572 S | WM\_STYLECHANGING wStyleType:GWL\_EXSTYLE | GWL\_STYLE lpss:86DBF080 |
| <00021> 001B0572 R | WM\_STYLECHANGING |
| <00022> 001B0572 S | WM\_STYLECHANGED wStyleType:GWL\_EXSTYLE | GWL\_STYLE lpss:86DBF080 |
| <00023> 001B0572 R | WM\_STYLECHANGED |
| <00024> 001B0572 S | WM\_STYLECHANGING wStyleType:GWL\_EXSTYLE | GWL\_STYLE lpss:86DBF080 |
| <00025> 001B0572 R | WM\_STYLECHANGING |
| <00026> 001B0572 S | WM\_STYLECHANGED wStyleType:GWL\_EXSTYLE | GWL\_STYLE lpss:86DBF080 |
| <00027> 001B0572 R | WM\_STYLECHANGED |
| <00028> 001B0572 S | WM\_STYLECHANGING wStyleType:GWL\_EXSTYLE | GWL\_STYLE lpss:86DBF080 |
| <00029> 001B0572 R | WM\_STYLECHANGING |
| <00030> 001B0572 S | WM\_STYLECHANGED wStyleType:GWL\_EXSTYLE | GWL\_STYLE lpss:86DBF080 |
| <00031> 001B0572 R | WM\_STYLECHANGED |
| <00032> 001B0572 S | WM\_STYLECHANGING wStyleType:GWL\_EXSTYLE | GWL\_STYLE lpss:86DBF080 |
| <00033> 001B0572 R | WM\_STYLECHANGING |
| <00034> 001B0572 S | WM\_STYLECHANGED wStyleType:GWL\_EXSTYLE | GWL\_STYLE lpss:86DBF080 |
| <00035> 001B0572 R | WM\_STYLECHANGED |
| <00036> 009705E0 S | WM\_COMMAND wNotifyCode:EN\_UPDATE wID:15 hwndCtl:001B0572 |
| <00037> 009705E0 R | WM\_COMMAND |
| <00038> 009705E0 S | WM\_CTLCOLOREDIT hdcEdit:63010D24 hwndEdit:001B0572 |
| <00039> 009705E0 R | WM\_CTLCOLOREDIT hBrush:00100075 |
| <00040> 001B0572 S | WM\_IME\_NOTIFY dwCommand:IMN\_SETCOMPOSITIONWINDOW dwCommand:0000000B dwData:00000000 |
| <00041> 001B0572 R | WM\_IME\_NOTIFY |
| <00042> 001B0572 S | WM\_GETTEXTLENGTH |
| <00043> 001B0572 R | WM\_GETTEXTLENGTH cch:10 |
| <00044> 001B0572 S | EM\_GETSEL lpdwStart:0000002486DBED90 lpdwEnd:0000002486DBED94 |
| <00045> 001B0572 R | EM\_GETSEL wStart:10 wEnd:10 lpdwStart:86DBED90 (10) lpdwEnd:86DBED94 (10) |
| <00046> 001B0572 S | EM\_GETRECT lprc:0000002486DBED38 |
| <00047> 001B0572 R | EM\_GETRECT lprc:0000002486DBED38 (40,0)-(34,0) |
| <00048> 001B0572 S | EM\_CHARFROMPOS x:4 y:0 |
| <00049> 001B0572 R | EM\_CHARFROMPOS wCharIndex:0 wLineIndex:0 |
| <00050> 001B0572 S | WM\_GETFONT |
| <00051> 001B0572 R | WM\_GETFONT hfont:600A0E1C ("Consolas" 11pt) |
| <00052> 001B0572 S | EM\_LINEFROMCHAR ich:10 |
| <00053> 001B0572 R | EM\_LINEFROMCHAR iLine:0 |
| <00054> 001B0572 S | EM\_POSFROMCHAR wCharIndex:10 |
| <00055> 001B0572 R | EM\_POSFROMCHAR x:65535 y:65535 |
| <00056> 001B0572 S | EM\_POSFROMCHAR wCharIndex:9 |
| <00057> 001B0572 R | EM\_POSFROMCHAR x:76 y:0 |
| <00058> 001B0572 S | EM\_LINELENGTH ich:0 |
| <00059> 001B0572 R | EM\_LINELENGTH cchLine:10 |
| <00060> 009705E0 S | WM\_COMMAND wNotifyCode:EN\_CHANGE wID:15 hwndCtl:001B0572 |
| <00061> 009705E0 R | WM\_COMMAND |
| <00062> 009705E0 S | WM\_NOTIFY idCtrl:15 pnmh:0000002486DBF5B0 |
| <00063> 009705E0 R | WM\_NOTIFY |
| <00064> 001B0572 P | WM\_KEYUP nVirtKey:VK\_CONTROL cRepeat:1 ScanCode:1D fExtended:0 fAltDown:0 fRepeat:1 |
| <00065> 001B0572 S | message:0x1512 [User-defined:WM\_USER+4370] wParam:00000000 lParam:00000000 |
| <00066> 001B0572 R | message:0x1512 [User-defined:WM\_USER+4370] lResult:0000000A |
| <00067> 001B0572 S | message:0x1513 [User-defined:WM\_USER+4371] wParam:0000000A lParam:00000000 |
| <00068> 001B0572 R | message:0x1513 [User-defined:WM\_USER+4371] lResult:00000000 |
| <00069> 001B0572 S | message:0x1514 [User-defined:WM\_USER+4372] wParam:00000000 lParam:00000000 |
| <00070> 001B0572 R | message:0x1514 [User-defined:WM\_USER+4372] lResult:00000000 |
| <00071> 00560414 S | SB\_SETTEXTW iPart:1 uType:0 szText:0000002486DBF530 |
| <00072> 00560414 S | WM\_PAINT hdc:00000000 |
| <00073> 00560414 S | WM\_ERASEBKGND hdc:63010D24 |
| <00074> 00560414 R | WM\_ERASEBKGND fErased:True |
| <00075> 00560414 R | WM\_PAINT |
| <00076> 00560414 R | SB\_SETTEXTW fSucceeded:True |
| <00077> 001B0572 S | EM\_GETMODIFY |
| <00078> 001B0572 R | EM\_GETMODIFY fModified:True |
| <00079> 001B0572 P | WM\_KEYUP nVirtKey:'V' cRepeat:1 ScanCode:2F fExtended:0 fAltDown:0 fRepeat:1 fUp:1 |

**Комментарий к протоколу**:

1-2 строки. WM\_KEYDOWN. Система фиксирует нажатие несистемных клавиш **“ctrl”** и **‘v’**. (сочетания без alt).

3-16. Меню редактирования становится активным, но не отображается, об этом говорит сообщение WM\_INITMENUPOPUP, после проведения необходимых операций, сообщение WM\_UNINITMENUPOPUP говорит нам, что меню было закрыто. Меню отвечает за вызов функции вставки, ей соответсвует сообщение M\_PASTE.

17-19. Вызов функции вставки. Сообщение **WM\_NOTIFY** отправляется стандартным органом управления своему родительскому окну, когда произошло некое событие или орган управления требует некоторой информации. В нашем случае происходит перерисовка окна редактора с вставленной строкой!!!!!

20-35. Перерисовка стилей окна WM\_STYLECHANGING, определяющих аспекты внешнего вида и поведения окна, например добавить горизонтальную или вертикальную линию прокрутки. Заканчивается сообщением WM\_STYLECHANGED.

36. WM\_COMMAND wNotifyCode: EN\_UPDATE wID:15 hwndCtl: 001B0572 сообщает, что элемент редактирования собирается перерисовать себя. Этот код уведомления отправляется после того, как элемент управления отформатировал текст, но до его отображения.

37-76 Перерисовка окна, задаются шрифты(WM\_GETFONT), устанавливаются цвета текста и фона (WM\_CTLCOLOREDIT). Далее создаётся так называемый “прямоугольник форматирования” - ограничивающий прямоугольник (EM\_GETRECT), задаются его размеры и координаты на экране. В конце процесса перерисовки там отобразится текст. После этого отправляется запрос на перерисовку окна (WM\_PAINT). Удаляется задний фон (WM\_ERASEBKGND) перед конечной вставкой текста (SB\_SETTEXTW).

77-79 Флаг EM\_GETMODIFY fModified:True сигнализирует, что редактирование прошло успешно. После чего была отжата клавиша ”v”. Аналогично в строке 64 была отжат “Ctrl”.