

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе №5
по дисциплине «Операционные системы»
ТЕМА: СОПРЯЖЕНИЕ СТАНДАРТНОГО И ПОЛЬЗОВАТЕЛЬСКОГО
ОБРАБОТЧИКОВ ПРЕРЫВАНИЙ

Студентка гр. 0382

Рубежова Н.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

Исследование возможности встраивания пользовательского обработчика прерываний в стандартный обработчик от клавиатуры. Пользовательский обработчик прерывания получает управление по прерыванию (int 09h) при нажатии клавиши на клавиатуре. Он обрабатывает скан-код и осуществляет определенные действия, если скан-код совпадает с определенными кодами, которые он должен обрабатывать. Если скан-код не совпадает с этими кодами, то управление передается стандартному прерыванию.

Задание.

Шаг 1. Для выполнения лабораторной работы необходимо написать и отладить программный модуль типа .EXE, который выполняет такие же функции, как в программе ЛР 4, а именно:

- 1) Проверяет, установлено ли пользовательское прерывание с вектором 09h.
- 2) Если прерывание не установлено то, устанавливает резидентную функцию для обработки прерывания и настраивает вектор прерываний. Адрес точки входа в стандартный обработчик прерывания находится в теле пользовательского обработчика. Осуществляется выход по функции 4Ch прерывания int 21h.
- 3) Если прерывание установлено, то выводится соответствующее сообщение и осуществляется выход по функции 4Ch прерывания int 21h.

Выгрузка прерывания по соответствующему значению параметра в командной строке /un. Выгрузка прерывания состоит в восстановлении стандартного вектора прерываний и освобождении памяти, занимаемой резидентом. Затем осуществляется выход по функции 4Ch прерывания int 21h.

Для того чтобы проверить установку прерывания, можно поступить следующим образом. Прочитать адрес, записанный в векторе прерывания. Предположим, что этот адрес указывает на точку входа в установленный резидент. На определенном, известном смещении в теле резидента

располагается сигнатура, некоторый код, который идентифицирует резидент. Сравнив известное значение сигнатуры с реальным кодом, находящимся в резиденте, можно определить, установлен ли резидент. Если значения совпадают, то резидент установлен. Длину кода сигнатуры должна быть достаточной, чтобы сделать случайное совпадение маловероятным.

Программа должна содержать код устанавливаемого прерывания в виде удаленной процедуры. Этот код будет работать после установки при возникновении прерывания. Он должен выполнять следующие функции:

- 1) Сохранить значения регистров в стеке при входе и восстановить их при выходе.
- 2) При выполнении тела процедуры анализируется скан-код.
- 3) Если этот код совпадает с одним из заданных, то требуемый код записывается в буфер клавиатуры.
- 4) Если этот код не совпадает ни с одним из заданных, то осуществляется передача управления стандартному обработчику прерывания.

Шаг 2. Запустите отлаженную программу и убедитесь, что резидентный обработчик прерывания 09h установлен. Работа прерывания проверяется введением различных символов, обрабатываемых установленным обработчиком и стандартным обработчиком.

Шаг 3. Также необходимо проверить размещение прерывания в памяти. Для этого запустите программу ЛР 3, которая отображает карту памяти в виде списка блоков МСВ. Полученные результаты поместите в отчет.

Шаг 4. Запустите отлаженную программу еще раз и убедитесь, что программа определяет установленный обработчик прерываний. Полученные результаты поместите в отчет.

Шаг 5. Запустите отлаженную программу с ключом выгрузки и убедитесь, что резидентный обработчик прерывания выгружен, то есть сообщения на экран не выводятся, а память, занятая резидентом освобождена. Для этого также следует запустить программу ЛР 3. Полученные результаты поместите в отчет.

Шаг 6. Ответьте на контрольные вопросы.

Выполнение работы.

Шаг 1. При реализации программы были написаны следующие процедуры, включая те, которые были написаны в ЛР4:

USR_INTERRUPT – пользовательский обработчик прерывания, который при вызове проверяет, нажаты ли клавиши LEFT SHIFT, Z и X, сравнивая данные из порта 60h со скан-кодами этих клавиш, если совпадает, то заменяет их на символы '!', '@', '#' соответственно.

IS_SET – процедура проверки, что пользовательский обработчик прерывания установлен. Результат проверки обрабатывается в процедуре MAIN и выводится соответствующее сообщение: «User interrupt has already set», если обработчик уже установлен, в обратном случае вызывается USR_INT_SET.

CHECK_COMMAND – проверка на наличие в командной строке параметра /un.

USR_INT_SET – установка пользовательского обработчика прерывания. Сохраняет оригинальный вектор прерывания(получили через функцию 35h прерывания int 21h), устанавливает пользовательский обработчик прерывания(через функцию 25h прерывания int 21h) и для того, чтобы оставить процедуру прерывания резидентной в памяти, использует функцию 31h прерывания int 21h.

USR_INT_UNLOAD – процедура выгрузки обработчика прерывания.

MAIN – основная процедура, в ней вызываются вышеописанные процедуры в зависимости от введенных пользователем команд и вызываются сопутствующие сообщения.

В отличие от ЛР4 была изменена процедура USR_INTERRUPT, пользовательский обработчик прерывания. Вектор прерывания изменен с 1Ch на 09h.

Шаг 2. Запустим программу. Выводится сообщение, что обработчик прерывания был установлен. Нажмем последовательно клавиши: LEFT SHIFT, Z, X, C, V. Вывод программы см. на рисунке ниже.

```
C:\>lr5
Interrupt is set.
C:\>abcd !@#_
```

Рисунок 1 – Первый запуск программы lr5.exe и последовательное нажатие клавиш A-B-C-D-SPACE-LEFT SHIFT-Z-X

Видим, что при нажатии клавиш LEFT SHIFT, Z, X, печатаются символы ‘!’, ‘@’, ‘#’, остальные выведенные символы соответствуют нажатым клавишам. Следовательно, пользовательский обработчик прерывания работает корректно.

Шаг 3. Запустим модуль var1.com, который покажет, загружен ли обработчик прерывания в память. Вывод программы см. на рисунке ниже.

```
C:\>var1
Available memory: 647712 bytes
Extended memory: 15728640 bytes
1_MCB Address:016F PSP_address: 0008 Size: 16 SD/SC:
2_MCB Address:0171 PSP_address: 0000 Size: 64 SD/SC:
3_MCB Address:0176 PSP_address: 0040 Size: 256 SD/SC:
4_MCB Address:0187 PSP_address: 0192 Size: 144 SD/SC:
5_MCB Address:0191 PSP_address: 0192 Size: 1024 SD/SC: LR5
6_MCB Address:01D2 PSP_address: 01DD Size: 1144 SD/SC:
7_MCB Address:01DC PSP_address: 01DD Size:647712 SD/SC: VAR1
```

Рисунок 2 – Запуск модуля var1.com для проверки загрузки обработчика прерывания в память

Видим, что обработчик прерывания, действительно, загружен в память (это подтверждает 5ый MCB блок).

Шаг 4. Запустим модуль lr5.exe повторно. Программа вывела сообщение, что обработчик прерываний уже установлен. Вывод программы см. на рисунке ниже.

```
C:\>lr5
Interrupt is set.

C:\>lr5
Interrupt has already set.
```

Рисунок 3 – Повторный запуск программы

Шаг 5. Выгрузим обработчик прерывания, выполнив команду запуска модуля lr5.exe с ключом '/un'. Программа вывела сообщение о том, что он был выгружен. И запустим модуль var1.com, чтобы удостовериться, что память была освобождена. Вывод программы см. на рисунке ниже.

```
C:\>lr5 /un
Interrupt is not set.

C:\>var1
Available memory: 648912 bytes
Extended memory: 15728640 bytes
1_MCB Address:016F PSP_address: 0008 Size: 16 SD/SC:
2_MCB Address:0171 PSP_address: 0000 Size: 64 SD/SC:
3_MCB Address:0176 PSP_address: 0040 Size: 256 SD/SC:
4_MCB Address:0187 PSP_address: 0192 Size: 144 SD/SC:
5_MCB Address:0191 PSP_address: 0192 Size:648912 SD/SC: VAR1

C:\>abcd zx_
```

Рисунок 4 – Выгрузка обработчика прерывания, проверка состояния памяти и проверка обработки клавиш

Видим, что обработчик, действительно, был выгружен и его нет в памяти. При вводе той же последовательности клавиш A-B-C-D-SPACE-LEFT SHIFT-Z-X, выводятся соответствующие им символы, что характерно для стандартного обработчика прерывания, следовательно, стандартный обработчик прерывания был восстановлен. Модуль lr5.exe работает корректно.

Ответы на контрольные вопросы.

1. Какого типа прерывания использовались в работе?

int 09H – аппаратное прерывание клавиатуры

int 16H — для работы с сервисом клавиатуры, тоже аппаратное

int 21H — программное прерывание для работы с DOS

2. Чем отличается скан-код от кода ASCII?

Скан-код — код, присвоенный каждой клавише, с помощью которого драйвер клавиатуры распознает, какая клавиша была нажата/отжата. Скан-код

может быть назначен, как клавишам, обозначающим символы, так и «служебным» клавишам, например, CAPS-LOCK, ENTER и др. ASCII-код — численное представление (кодировка) символов из таблицы ASCII. Является именно кодировкой символа.

Выводы.

В результате работы был реализован пользовательский обработчик прерывания, который получает управление по прерыванию (int 09h) при нажатии клавиш LEFT SHIFT, Z, X на клавиатуре. Он обрабатывает скан-код нажатой клавиши, если скан-код совпадает с кодами ранее перечисленных клавиш, то записывает в буфер символы '!', '@', '#' соответственно. Если скан-код не совпадает с этими кодами, то управление передается стандартному прерыванию.