

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

**ОТЧЕТ**  
**по лабораторной работе №5**  
**по дисциплине «Операционные сети»**  
**Тема: Сопряжение стандартного и пользовательского**  
**обработчиков прерываний**

Студентка гр. 0381

Сарычева А.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

## **Цель работы.**

Исследование возможности встраивания пользовательского обработчика прерываний в стандартный обработчик от клавиатуры

## **Задание.**

Шаг 1. Для выполнения лабораторной работы необходимо написать и отладить программный модуль типа .EXE, который выполняет следующие функции:

1) Проверяет, установлено ли пользовательское прерывание с вектором 09h.

2) Устанавливает резидентную функцию для обработки прерывания и настраивает вектор прерываний, если прерывание не установлено, и 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.

4) 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.

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.

### **Выполнение работы.**

В файле lb5.asm написан исходный код .EXE модуля. В начале данного модуля прописаны строки для вывода запрашиваемой информации, для вывода которых была создана процедура OUTPUT.

Написана процедура CHECK\_TAIL, которая проверяет был ли введен параметр /un при запуске программы с помощью побайтного сравнения ячеек памяти, отвечающих за хвост командной строки и его размер с необходимой строкой UNLOAD\_MESSAGE.

Процедура CHECK\_INSTALL проверяет установлено ли пользовательское прерывания с помощью функции получения вектора 35h int 21h и сравнения поля вектора и строки SIGNATURE по необходимому адресу в этом векторе, если они равны, то процедура в регистре cl сохраняет 1, иначе 0.

Процедура UNLOAD, если пользовательское прерывание установлено, выгружает его, восстанавливая старый обработчик прерывания с помощью функции 25h int 21, при этом освобождая занятые блоки памяти с помощью

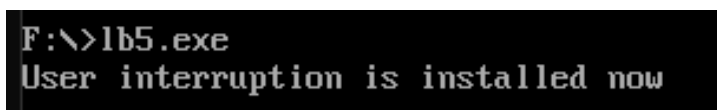
функции 49h int 21. В конце выводится сообщение о выполненном действии с помощью процедуры OUTPUT.

Далее написана процедура DON'T\_UNLOAD, которая сначала с помощью CHECK\_INSTALL проверяет установлен ли обработчик прерывания, если да, то просто выводится сообщение, что он уже установлен, иначе устанавливает резидентную функцию для обработки прерывания и настраивает вектор прерываний с помощью функций 31h и 25h int 21h. В конце выводится сообщение о установке обработчика прерывания.

Процедура NEW\_INTERRUPTION является реализованным прерыванием, которое имеет свой стек, в который сохраняются значения всех используемых далее регистров для их восстановления при выходе. После загрузки в память, прерывание начинает анализировать скан-коды вводимых символов. Так, если скан-код нажатой клавиши совпадает с заданным в программе – Alt(левый), то в буфер клавиатуры записывается символ «S», иначе происходит передача управления стандартному обработчику прерывания.

В головной процедуре происходит вызов процедуры CHECK\_TAIL, от результата которой вызывается либо процедура UNLOAD, либо DONT\_UNLOAD, после которых происходит выход из программы с помощью функции 4Ch прерывания 21h.


Написанная программа при ее первом вызове, устанавливает обработчик прерывания и выводит об этом сообщение. (см. рис. 1).



```
F:\>lb5.exe
User interruption is installed now
```

Рисунок 1 – Запуск модуля lb5.exe

Так, после установки данного обработчика при нажатии клавиши Alt(левый), на экран записывается символ «S», при нажатие остальных клавиш выводятся запрашиваемые символы. (см. рис. 2).



```
F:\>jfndoSSSSuddunSSidspaSSSSSS
```

Рисунок 2 – Отображение символов после запуска модуля lb5.exe

При запуске программы лабораторной работы №3 видно, что программа из текущей лабораторной работы загружена в память и ей выделена необходимая память (см. рис. 3).

```
F:\>lb3.com
Amount of available memory: 64 bytes
Extended memory size: 585728 bytes
Memory control blocks:
1  Type: 4Dh  MSB adress: 016Fh  PSP adress: 0008h  MSB size: 16  text:
2  Type: 4Dh  MSB adress: 0171h  PSP adress: 0000h  MSB size: 64  text:
3  Type: 4Dh  MSB adress: 0176h  PSP adress: 0040h  MSB size: 256  text:
4  Type: 4Dh  MSB adress: 0187h  PSP adress: 0192h  MSB size: 144  text:
5  Type: 4Dh  MSB adress: 0191h  PSP adress: 0192h  MSB size: 928  text: LB5
6  Type: 4Dh  MSB adress: 01CCh  PSP adress: 01D7h  MSB size: 144  text:
7  Type: 5Ah  MSB adress: 01D6h  PSP adress: 01D7h  MSB size: 647808  text: LB3
```

Рисунок 3 – Запуск модуля lb3.com после загрузки обработчика прерывания

При повторной попытке загрузить созданный модуль программы выводится сообщения о том, что обработчик уже установлен, а клавиши продолжают отображаться в соответствие с заданным правилом (см. рис. 4).

```
F:\>lb5.exe
User interruption has already installed.
F:\>fvodmsSSSSSdosmiSSSdfo_
```

Рисунок 4 –Повторный запуск модуля lb5.exe

При попытке же запуска данного модуля с ключом /un, происходит восстановление стандартного вектора прерывания и освобождение памяти, занимаемой резидентом, после которого вывод соответствующего сообщения и завершение работы программы. Так, после завершения программы при попытке нажатия клавиши Alt(левый) ничего не происходит и символ «S» не отображается (см. рис. 5), а в освобождение памяти можно убедиться, повторно запустив программу лабораторной работы №3.

```

F:\>lb5.exe /un
Unload interruption

F:\>lb3.com
Amount of available memory: 64 bytes
Extended memory size: 585728 bytes
Memory control blocks:
 1 Type: 4Dh MSB adress: 016Fh PSP adress: 0008h MSB size: 16 text:
 2 Type: 4Dh MSB adress: 0171h PSP adress: 0000h MSB size: 64 text:
 3 Type: 4Dh MSB adress: 0176h PSP adress: 0040h MSB size: 256 text:
 4 Type: 4Dh MSB adress: 0187h PSP adress: 0192h MSB size: 144 text:
 5 Type: 5Ah MSB adress: 0191h PSP adress: 0192h MSB size: 648912 text: LB3

F:\>okumomufiupsu_

```

Рисунок 5 – Запуск модуля lb5.exe с ключом /un

### **Выводы.**

Была исследована возможность встраивания пользовательского обработчика прерываний в стандартный обработчик от клавиатуры.

### **Ответы на контрольные вопросы.**

1) Какого типа прерывания использовались в работе?

В работе использовались аппаратные прерывания: 09h и 16h, и программное прерывание int 21h.

2) Чем отличается скан код от кода ASCII?

Скан-код – уникальное число, однозначно определяющее нажатую клавишу. Скан-коды бывают двух видов: при нажатии клавиши генерируется так называемый Make-код, а при её отпуске Break-код. В отличие от скан-кода, ASCII код описывает не клавишу, а символ, который изображен на ней, так ASCII-код имеет код не для каждой клавиши.