

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

**ОТЧЕТ**  
**по лабораторной работе №4**  
**по дисциплине «Операционные системы**  
**Тема: Обработка стандартных прерываний**

Студент гр. 9381

Щеглов Д.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

### **Цель работы.**

В архитектуре компьютера существуют стандартные прерывания, за которыми закреплены определенные вектора прерываний. Вектор прерываний хранит адрес подпрограммы обработчика прерываний. При возникновении прерывания, аппаратура компьютера передает управление по соответствующему адресу вектора прерывания. Обработчик прерываний получает управление и выполняет соответствующие действия.

### **Постановка задачи.**

Требуется написать и отладить .EXE модуль, который выполняет следующие функции:

- 1) Проверяет, установлено ли пользовательское прерывание с вектором 1Ch.
- 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 освобождения памяти, занимаемой резидентом. Затем осуществляется выход по функции 4Ch прерывания int 21h.

Программа должна содержать код устанавливаемого прерывания в виде удаленной процедуры. Код должен:

- 1) Сохранять значения регистров в стеке при входе и восстановить их при выходе.
- 2) При выполнении тела процедуры накапливать общее суммарное число прерываний и выводить на экран. Для вывода на экран следует использовать прерывание int 10h, которое позволяет непосредственно выводить информацию на экран.

Процедуры в программе:

ROUT\_INTERRUPT – пользовательское прерывание

OUTPUTTING\_STRING\_TO\_CONSOLE – вывод строки на консоль.

LOADING\_KEY – проверка есть ли параметр /un в командной строке.

IS\_LOADED – проверяет, есть ли прерывание в памяти.

LOADING\_INTERRUPT – загрузка прерывания в память.

UNLOADING\_INTERRUPT – выгрузка прерывания из памяти.

### **Выполнение работы.**

Сначала был написан код для .EXE модуля. Модуль работает следующим образом: если он запущен без ключа /un, то он установит прерывание в память. Далее, при попытках снова установить это прерывание будет выводиться соответствующая строка. Это можно видеть на рисунке 1.



```
C:\> Interrupts: 1381
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>
C:\>lab4.exe
Interrupt was succesfully loaded.
C:\>
```

Рис. 1: Запуск .EXE модуля

Также, прерывание можно выгрузить, запустив программу с ключом /un. Если запустить программу так же еще раз, после выгрузки, то выведется соответствующее сообщение. Это представлено на рисунке 2.

```

C:\>lab4.exe /un
There was interrupt to unload. As a result, interrupt was unloaded.

C:\>lab4.exe /un
There was no interrupt to unload.

C:\>

```

Рис. 2: Выгрузка прерывания

Далее на рисунках 3-6 представлены соответственно: расположение прерывания в памяти, расположение прерывания при попытке повторной загрузки прерывания, расположение прерывания при выгрузке и расположение прерывания при повторной выгрузке соответственно.

```

C:\>
Interrupts: 0201
C:\>
C:\>
C:\>
C:\>lab4.exe
Interrupt was succesfully loaded.

C:\>lab3_2.com
Number of accessed memory: 648240 bytes
Size of extended memory: 245760 bytes
MCB 1: Segment address of PSP: 0008 Size of area: 16 SD/SC:
MCB 2: Segment address of PSP: 0000 Size of area: 64 SD/SC:
MCB 3: Segment address of PSP: 0040 Size of area: 256 SD/SC:
MCB 4: Segment address of PSP: 0192 Size of area: 144 SD/SC:
MCB 5: Segment address of PSP: 0192 Size of area: 496 SD/SC: LAB4
MCB 6: Segment address of PSP: 01BC Size of area: 144 SD/SC:
MCB 7: Segment address of PSP: 01BC Size of area: 768 SD/SC: LAB3_2
MCB 8: Segment address of PSP: 0000 Size of area: 647456 SD/SC: 80E1L=?

C:\>

```

Рис. 3: Расположение прерывания в памяти

```

MCB 1: Segment address of PSP: 0008 Size of area: 16 SD/SC:
MCB 2: Segment address of PSP: 0000 Size of area: 64 SD/SC:
MCB 3: Interrupts: 0827s of PSP: 0040 Size of area: 256 SD/SC:
MCB 4: Segment address of PSP: 0192 Size of area: 144 SD/SC:
MCB 5: Segment address of PSP: 0192 Size of area: 496 SD/SC: LAB4
MCB 6: Segment address of PSP: 01BC Size of area: 144 SD/SC:
MCB 7: Segment address of PSP: 01BC Size of area: 768 SD/SC: LAB3_2
MCB 8: Segment address of PSP: 0000 Size of area: 647456 SD/SC: 80E|L=?

C:\>lab4.exe
Interrupt is already loaded.

C:\>lab3_2.com
Number of accessed memory: 648240 bytes
Size of extended memory: 245760 bytes
MCB 1: Segment address of PSP: 0008 Size of area: 16 SD/SC:
MCB 2: Segment address of PSP: 0000 Size of area: 64 SD/SC:
MCB 3: Segment address of PSP: 0040 Size of area: 256 SD/SC:
MCB 4: Segment address of PSP: 0192 Size of area: 144 SD/SC:
MCB 5: Segment address of PSP: 0192 Size of area: 496 SD/SC: LAB4
MCB 6: Segment address of PSP: 01BC Size of area: 144 SD/SC:
MCB 7: Segment address of PSP: 01BC Size of area: 768 SD/SC: LAB3_2
MCB 8: Segment address of PSP: 0000 Size of area: 647456 SD/SC: 80E|L=?

C:\>_

```

Рис. 4: Расположение прерывания в памяти при повторной загрузке

```

Number of accessed memory: 648240 bytes
Size of extended memory: 245760 bytes
MCB 1: Segment address of PSP: 0008 Size of area: 16 SD/SC:
MCB 2: Segment address of PSP: 0000 Size of area: 64 SD/SC:
MCB 3: Segment address of PSP: 0040 Size of area: 256 SD/SC:
MCB 4: Segment address of PSP: 0192 Size of area: 144 SD/SC:
MCB 5: Segment address of PSP: 0192 Size of area: 496 SD/SC: LAB4
MCB 6: Segment address of PSP: 01BC Size of area: 144 SD/SC:
MCB 7: Segment address of PSP: 01BC Size of area: 768 SD/SC: LAB3_2
MCB 8: Segment address of PSP: 0000 Size of area: 647456 SD/SC: 80E|L=?

C:\>lab4.exe /un
There was interrupt to unload. As a result, interrupt was unloaded.

C:\>lab3_2.com
Number of accessed memory: 648912 bytes
Size of extended memory: 245760 bytes
MCB 1: Segment address of PSP: 0008 Size of area: 16 SD/SC:
MCB 2: Segment address of PSP: 0000 Size of area: 64 SD/SC:
MCB 3: Segment address of PSP: 0040 Size of area: 256 SD/SC:
MCB 4: Segment address of PSP: 0192 Size of area: 144 SD/SC:
MCB 5: Segment address of PSP: 0192 Size of area: 768 SD/SC: LAB3_2
MCB 6: Segment address of PSP: 0000 Size of area: 648128 SD/SC:

C:\>_

```

Рис. 5: Расположение прерывания при выгрузке

```

C:\>lab3_2.com
Number of accessed memory: 648912 bytes
Size of extended memory: 245760 bytes
MCB 1: Segment address of PSP: 0008 Size of area: 16 SD/SC:
MCB 2: Segment address of PSP: 0000 Size of area: 64 SD/SC:
MCB 3: Segment address of PSP: 0040 Size of area: 256 SD/SC:
MCB 4: Segment address of PSP: 0192 Size of area: 144 SD/SC:
MCB 5: Segment address of PSP: 0192 Size of area: 768 SD/SC: LAB3_2
MCB 6: Segment address of PSP: 0000 Size of area: 648128 SD/SC:

C:\>lab4.exe /un
There was no interrupt to unload.

C:\>lab3_2.com
Number of accessed memory: 648912 bytes
Size of extended memory: 245760 bytes
MCB 1: Segment address of PSP: 0008 Size of area: 16 SD/SC:
MCB 2: Segment address of PSP: 0000 Size of area: 64 SD/SC:
MCB 3: Segment address of PSP: 0040 Size of area: 256 SD/SC:
MCB 4: Segment address of PSP: 0192 Size of area: 144 SD/SC:
MCB 5: Segment address of PSP: 0192 Size of area: 768 SD/SC: LAB3_2
MCB 6: Segment address of PSP: 0000 Size of area: 648128 SD/SC: 80E-L=?

C:\>_

```

Рис. 6: Расположение прерывания при повторной выгрузке

### Ответы на вопросы.

- 1) Как реализован механизм прерывания от часов?

При каждом тике таймера сохраняется состояние регистров, определяется источник прерывания, после чего первые два байта перемещаются в IP, вторые два – в CS. После запускается обработчик прерывания, который производит обработку прерывания, после чего возвращает управление прерванной программе.

- 2) Какого типа прерывания использовались в работе?

Пользовательское прерывание int 21h с вектором 1Ch и аппаратные прерывания int 10h и int 21h.

### Выводы.

Был написан обработчик прерываний сигналов таймера.