

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

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

Студент гр. 7381

\_\_\_\_\_

Дорох С.В.

Преподаватель

\_\_\_\_\_

Ефремов М. А.

Санкт-Петербург

2019

## **Цель работы.**

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

В лабораторной работе № 4 предлагается построить обработчик прерываний сигналов таймера. Эти сигналы генерируются аппаратурой через определённые интервалы времени и, при возникновении такого сигнала, возникает прерывание с определённым значением вектора. Таким образом, управление будет передано функции, чья точка входа записана в соответствующий вектор прерывания.

## **Постановка задачи:**

**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 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 о 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.

**Шаг 2.** Далее необходимо запустить отлаженную программу и

убедиться, что резидентный обработчик прерывания 1Ch установлен. Работа прерывания должна отображаться на экране, а также необходимо проверить размещение прерывания в памяти. Для этого нужно запустить программу ЛР 3, которая отображает карту памяти в виде с писка блоков МСВ.

**Шаг 3.** Затем необходимо запустить отлаженную программу еще раз и убедиться, что программа определяет установленный обработчик прерываний.

**Шаг 4.** Далее нужно запустить отлаженную программу с ключом выгрузки и убедиться, 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.

### **Необходимые сведения для составления программы:**

Резидентные обработчики прерываний - это программные модули, которые вызываются при возникновении прерываний определенного типа (сигнал таймера, нажатие клавиши и т.д.), которым соответствуют определенные вектора прерывания. Когда вызывается прерывание, процессор переключается на выполнение кода обработчика, а затем возвращается на выполнение прерванной программы. Адрес возврата в прерванную программу (CS:IP) запоминается в стеке вместе с регистром флагов. Затем в CS:IP загружается адрес точки входа программы обработки прерывания и начинает выполняться его код. Обработчик прерывания должен заканчиваться инструкцией IRET (возврат из прерывания).

Вектор прерывания имеет длину 4 байта. В первом хранится значение IP, во втором - CS. Младшие 1024 байта памяти содержат 256 векторов. Вектор для прерывания 0 начинается с ячейки 0000:0000, для прерывания 1 - с ячейки 0000:0004 и т.д.

Обработчик прерывания - это отдельная процедура, имеющая следующую структуру:

ROUT PROC FAR

PUSH AX ; сохранение изменяемых регистров

.....  
<действия по обработке прерывания>

```
POP AX ; восстановление регистров
MOV AL, 20H
OUT 20H,AL
IRET
ROUT ENDP
```

Две последние строки необходимы для разрешения обработки прерываний с более низкими уровнями, чем только что обработанное. Для установки написанного прерывания в поле векторов прерываний используется функция 25H прерывания 21H, которая устанавливает вектор прерывания на указанный адрес.

```
PUSH DS
MOV DX, OFFSET ROUT ; смещение для процедуры в DX
MOV AX, SEG ROUT ; сегмент процедуры
MOV DS, AX ; помещаем в DS
MOV AH, 25H ; функция установки вектора
MOV AL, 1CH ; номер вектора
INT 21H ; меняем прерывание
POP DS
```

Программа, выгружающая обработчик прерываний должна восстанавливать оригинальные векторы прерываний. Функция 35 прерывания 21H позволяет восстановить значение вектора прерывания, помещая значение сегмента в ES, а смещение в BX. Программа должна содержать следующие инструкции:

```
; -- хранится в обработчике прерываний
KEEP_CS DW 0 ; для хранения сегмента
KEEP_IP DW 0 ; и смещения прерывания
; -- в программе при загрузке обработчика прерывания
MOV AH, 35H ; функция получения вектора
MOV AL, 1CH ; номер вектора ШТЕ 21P
MOV KEEP_IP, BX ; запоминание смещения
MOV KEEP_CS, ES ; и сегмента
; -- в программе при выгрузке обработчика прерываний CLI
PUSH DS
```

```

MOV DX, KEEP_IP
MOV AX, KEEP_CS
MOV DS, AX
MOV AH, 25H
MOV AL, 1CH
INT 21H    ; восстанавливаем вектор
POP DS
STI

```

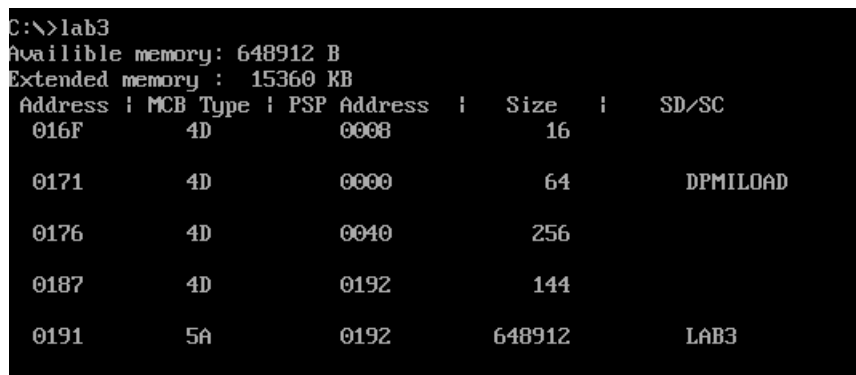
Для того, чтобы оставить процедуру прерывания резидентной в памяти, следует воспользоваться функцией DOS 31h прерывания 21h. Эта функция оставляет память, размер которой указывается в качестве параметра, занятой, а остальную память освобождает и осуществляет выход в DOS.

Функция 31h int 21h использует следующие параметры:

AH - номер функции 31h;  
 AL - код завершения программы;  
 DX - размер памяти в параграфах, требуемый резидентной программе.  
 Пример обращения к функции:  
 MOV DX, OFFSET LAST\_BYTE ; размер в байтах от начала сегмента  
 MOV CL, 4 ; перевод в параграфы  
 SHR DX, CL  
 INC DX ; размер в параграфах  
 MOV AH, 31h  
 INT 21h

### Ход работы.

- 1) Память до загрузки в нее резидентного прерывания(lab3.com).



```

C:\>lab3
Available memory: 648912 B
Extended memory : 15360 KB

```

Address	MCB Type	PSP Address	Size	SD/SC
016F	4D	0008	16	
0171	4D	0000	64	DPMILOAD
0176	4D	0040	256	
0187	4D	0192	144	
0191	5A	0192	648912	LAB3

- 2) Проверка памяти после загрузки в нее резидентного прерывания.

```
C:\>lab4
Resident is download

C:\>lab3
Available memory: 648144 B
Extended memory : 15360 KB
Address | MCB Type | PSP Address | Size | SD/SC
016F    4D    0008        16
0171    4D    0000        64    DPMILOAD
0176    4D    0040       256
0187    4D    0192       144
0191    4D    0192       592    LAB4
01B7    4D    01C2       144
01C1    5A    01C2    648144    LAB3
```

- 3) Попытка загрузки повторного резидентного прерывания и выгрузка резидентного прерывания.

```
C:\>lab4
Resident is already download

C:\>lab4 /un
Vector restored
```

- 4) Проверка памяти после выгрузки резидентного прерывания.

```
C:\>lab4 /un
Vector restored

C:\>lab3
Available memory: 648912 B
Extended memory : 15360 KB
Address | MCB Type | PSP Address | Size | SD/SC
016F    4D    0008        16
0171    4D    0000        64    DPMILOAD
0176    4D    0040       256
0187    4D    0192       144
0191    5A    0192    648912    LAB3
```

## Контрольные вопросы.

1. Как реализован механизм прерывания от часов?

Сначала сохраняется содержимое регистров, потом определяется источник прерывания, по номеру которого определяется смещение в

таблице векторов прерывания, сохраняется в CS : IP, передаётся управление по адресу CS:IP и происходит выполнение обработчика, и в конце происходит возврат управления прерванной программе. Прерывания генерируются системным таймером с частотой 18,206 Гц.

2. Какого типа прерывания использовались в работе?

В работе использовались прерывания 21h, 10h, 1Ch.

### **Выводы.**

В результате выполнения данной лабораторной работы были исследованы организация и управление прерываниями, был построен обработчик прерывания от сигналов таймера.