

**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»**

Студентка гр. 8381

Бердникова А.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2020

## **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.

## **Основные теоретические положения.**

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 начинается 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
        INT 21H
        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 прерывания 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

## **Сведения о программе**

Процедуры программы:

- OUTPUT\_AL - Процедура вывода одного символа из AL в текущее положение курсора
- OUTPUT\_AL2 - Процедура вывода двух символов из AL в текущее положение курсора
- GET\_CURS - Процедура получения позиции курсора
- SET\_CURS - Процедура установки курсора
- INTERRUPT - Процедура с прерыванием
- INTERRUPT\_HANDLER\_REPLACEMENT - Процедура замены обработчика прерывания
- LOAD\_RESIDENT - Процедура загрузки резидента
- REMOVE\_RESIDENT - Процедура удаления резидента из памяти
- FREE\_MEMORY - Процедура освобождения памяти

## **Ход работы.**

1) В файле lab4.asm была написана требуемая программа, из этого файла был получен исполняемый модуль lab4.exe и запущен. Для проверки нахождения программы в памяти была запущена программа lab3\_1.com из лабораторной работы номер 3. Результат её работы и действия обработчика прерываний:

```

Amount of available memory (in bytes): 648912
Amount of extended memory (in kilobytes): 15360
-----
MS DOS area
Memory size (in bytes): 16
Last bytes:
-----
Free area
Memory size (in bytes): 64
Last bytes:
-----
Unknown possessor: 0040
Memory size (in bytes): 256
Last bytes:
-----
Unknown possessor: 0192
Memory size (in bytes): 144
Last bytes:
-----
Unknown possessor: 0192
Memory size (in bytes): 648912
Last bytes: LAB3_1

```

Рисунок 1 - Результат работы 3 л/р и действия обработчика прерываний

2) Lab4.exe запуск, результат работы:

```

C:\>lab4.exe
Resident has been load in memory!

```

Рисунок 2 - Результат выполнения lab4.exe

3) Повторный запуск лабораторной №3:

```

Free area
Memory size (in bytes): 64
Last bytes:
-----
Unknown possessor: 0040
Memory size (in bytes): 256
Last bytes:
-----
Unknown possessor: 0192
Memory size (in bytes): 144
Last bytes:
-----
Unknown possessor: 0192
Memory size (in bytes): 992
Last bytes: LAB4
-----
Unknown possessor: 01DB
Memory size (in bytes): 144
Last bytes:
-----
Unknown possessor: 01DB
Memory size (in bytes): 647744
Last bytes: LAB3_1

```

Рисунок 3 - Результат повторного выполнения 3 л/р

4) Lab4.exe, повторный запуск с ключом удаления “/un”, для проверка нахождения программы в памяти была запущена программа из лабораторной работы номер 3. Результат работы:

```
C:\>lab4.exe /un  
Resident has been removed!
```

Рисунок 4 - Запуск lab4.exe с ключом удаления “/un”

```
Amount of available memory (in bytes): 648912  
Amount of extended memory (in kilobytes): 15360  
-----  
MS DOS area  
Memory size (in bytes): 16  
Last bytes:  
-----  
Free area  
Memory size (in bytes): 64  
Last bytes:  
-----  
Unknown possessor: 0040  
Memory size (in bytes): 256  
Last bytes:  
-----  
Unknown possessor: 0192  
Memory size (in bytes): 144  
Last bytes:  
-----  
Unknown possessor: 0192  
Memory size (in bytes): 648912  
Last bytes: LAB3_1
```

Рисунок 4 - Результат повторного выполнения модуля lab4.exe с ключом удаления “/un”

## Ответы на контрольные вопросы:

### Лабораторная работа №3

1. Как реализован механизм прерывания от часов?

Ответ: 1. Увеличение счётчика системного таймера.

2. Проверка на наличие запрета прерываний этого типа.

3. Если запрет присутствует, то продолжается выполнение заданных команд.

4. Если запрет отсутствует, то осуществляется вызов обработчика прерываний

5. Устанавливается запрет на прерывания данного типа.

6. Запоминается контекст прерванного процесса.

7. Происходит обработка прерывания.

8. Происходит восстановление прерванного процесса, разрешаются прерывания данного типа.

9. Осуществляется возврат к заданной последовательности команд, туда, где был получен запрос прерывания.

2. Какого типа прерывания использовались в работе?

Ответ: Был построен обработчик прерываний сигналов таймера, также использовались прерывания для обращений к функциям DOS

### **Выводы.**

В ходе выполнения лабораторной работы был построен обработчик прерывания от сигналов таймера.