

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

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

Студент гр. 7383

Зуев Д.В.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2019

## **Постановка задачи.**

### **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.

### **Реализация задачи:**

В данной работе были написаны и использовались готовые следующие функции:

TETR\_TO\_HEX – вспомогательная функция для функции BYTE\_TO\_HEX

BYTE\_TO\_HEX – переводит байт AL в два символа шестнадцатеричного числа в AX.

WRD\_TO\_HEX – переводит шестнадцатиразрядное число, расположенное в AX в шестнадцатеричную систему счисления.

PRINT – вызывает прерывание программы для печати строки.

ROUT – пользовательский обработчик прерывания, загружаемый в память, выводит счетчик срабатывания таймера.

getCurs – функция, читающая позицию и размер курсора.

setCurs – функция, устанавливающая позицию курсора.

outputBP – функция вывода строки по адресу ES:BP на экран.

LOAD\_ROUT – функция, загружающая обработчик прерывания в память.

KEEP\_ROUT – функция, оставляющая обработчик прерывания резидентным в памяти.

UNLOAD\_ROUT – функция, выгружающая обработчик прерываний из памяти.

CHECK – функция, проверяющая сперва параметр выгрузки обработчика прерывания из памяти, потом сигнатуру пользовательского прерывания и в зависимости от этих данных выводит сообщение и загружает или выгружает обработчик прерывания из памяти.

Результаты работы программы представлены на рисунках 1 – 4.

```
C:\>lab4.exe
Resident interrupt is loaded
The interrupt is triggered by 021Ctimes
C:\>lab3_1.com
Amount of available memory is 647424 b
Size of extended memory is 15360 Kb
Chain of MCB is
Address | Type | PSP owner | Size | Name
016F    D4h    0008h      16
0171    D4h    0000h      64
0176    D4h    0040h     256
0187    D4h    0192h     144      P
0191    D4h    0192h    1312    LAB4      =
01E4    D4h    01EFh     1144      P
01EE    A5h    01EFh   647424  LAB3_1    =
```

Рисунок 1 – Результат загрузки обработчика прерывания в память

```
C:\>lab4.exe
Resident interrupt is already loaded
```

Рисунок 2 – Результат повторной загрузки обработчика прерывания в память

```

C:\>lab4.exe /un
Resident interrupt is unloaded

C:\>lab3_1.com
Amount of available memory is 648912 b
Size of extended memory is 15360 Kb
Chain of MCB is
Address | Type | PSP owner | Size | Name
016F    D4h    0008h      16
0171    D4h    0000h      64
0176    D4h    0040h     256
0187    D4h    0192h     144      P
0191    A5h    0192h    648912  LAB3_1  =

```

Рисунок 3 – Результат выгрузки обработчика прерывания из памяти

```

C:\>lab4.exe /un
Resident interrupt is not loaded

```

Рисунок 4 – Результат повторной выгрузки обработчика прерывания из памяти

## Выводы.

В ходе выполнения данной лабораторной работы была написана программа, проверяющая наличие пользовательского прерывания в памяти и параметр выгрузки обработчика. В зависимости от этих данных загружает или выгружает обработчик прерывания из памяти и выводит соответствующее сообщение.

## Контрольные вопросы.

- 1) Как реализован механизм прерывания от часов?

*При каждом такте системного таймера (каждые 55 микросекунд; приблизительно 18.2 раз в секунду) сохраняется значение регистров, определяется источник прерывания, по номеру которого определяется смещение в таблице векторов прерываний, данные вектора прерывания помещаются в CS:IP, после передается управление по адресу CS:IP, происходит обработка прерывания и возврат управления прерванной программе.*

2) Какого типа прерывания использовались в работе?

*В работе использовались аппаратные прерывания (int 1Ch) и программные прерывания (int 10h, int 21h).*