

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

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

Студент гр. 0381

Березовская В. В.

Преподаватель

Ефремов М. А.

Санкт-Петербург

2022

### **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.

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

### **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 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 освобождении памяти, занимаемой

резидентом. Затем осуществляется выход по функции 4Ch прерывания int 21h.

Для того, чтобы проверить установку прерывания, можно поступить следующим образом. Прочитать адрес, записанный в векторе прерывания. Предположим, что этот адрес указывает на точку входа в установленный резидент. На определенном, известном смещении в теле резидента располагается сигнатура, некоторый код, который идентифицирует резидент. Сравнив известное значение сигнатуры с реальным кодом, находящимся в резиденте, можно определить, установлен ли резидент. Если значения совпадают, то резидент установлен. Длину кода сигнатуры должна быть достаточной, чтобы сделать случайное совпадение маловероятным.

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) Сохраняет стек прерванной программы (регистры SS и SP) в рабочих переменных и восстановить при выходе.
- 2) Организовать свой стек.
- 3) Сохранить значения регистров в стеке при входе и восстановить их при выходе.
- 4) 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.
- 5) Функция прерывания должна содержать только переменные, которые она использует

**Шаг 2.** Запустите отлаженную программу и убедитесь, что резидентный обработчик прерывания 1Ch установлен. Работа прерывания должна отображаться на экране, а 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.

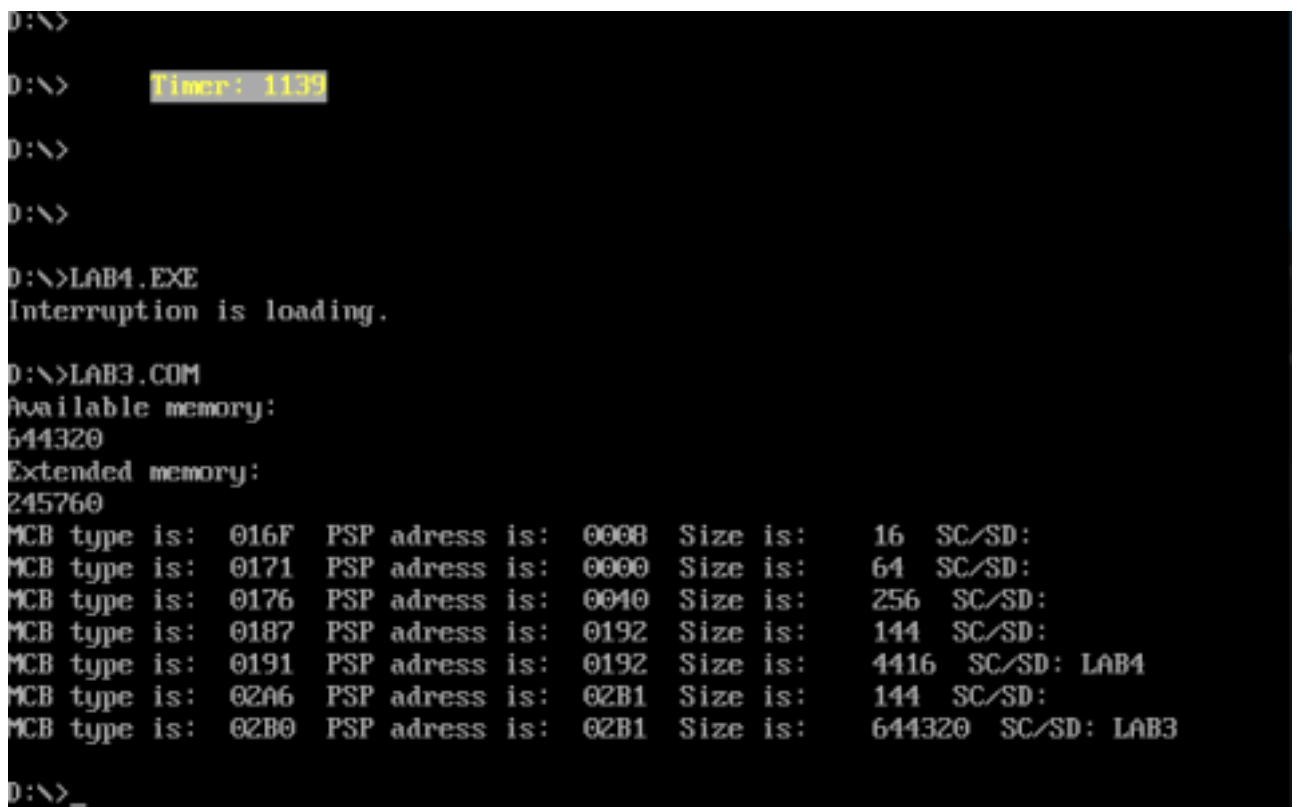
**Шаг 3.** Запустите отлаженную программу еще раз и убедитесь, что программа определяет установленный обработчик прерываний. Полученные результаты поместите в отчет.

**Шаг 4.** 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.

### **Выполнение работы.**

**Шаг 1.** Был написан и отлажен программный модуль типа .EXE, который выполняет, выполняет требуемый функционал.

**Шаг 2.** Программа была отлажена и запущена. Проверено размещение прерывания в памяти(5ая строчка в таблице МСВ).



```
D:\>
D:\> Timer: 1139
D:\>
D:\>
D:\> LAB4.EXE
Interruption is loading.
D:\> LAB3.COM
Available memory:
644320
Extended memory:
245760
MCB type is: 016F PSP adress is: 0008 Size is: 16 SC/SD:
MCB type is: 0171 PSP adress is: 0000 Size is: 64 SC/SD:
MCB type is: 0176 PSP adress is: 0040 Size is: 256 SC/SD:
MCB type is: 0187 PSP adress is: 0192 Size is: 144 SC/SD:
MCB type is: 0191 PSP adress is: 0192 Size is: 4416 SC/SD: LAB4
MCB type is: 02A6 PSP adress is: 02B1 Size is: 144 SC/SD:
MCB type is: 02B0 PSP adress is: 02B1 Size is: 644320 SC/SD: LAB3
D:\>_
```

Рисунок 1 – Демонстрация корректной работы резидентного обработчика прерываний

**Шаг 3.** Программа была повторно запущена, чтобы удостовериться, что программа определяет установленный обработчик прерываний.

```

D:\>
Timer: 2900
D:\>
D:\>LAB4.EXE
Interruption is loading.
D:\>LAB3.COM
Available memory:
644320
Extended memory:
245760
MCB type is: 016F PSP adress is: 0000 Size is: 16 SC/SD:
MCB type is: 0171 PSP adress is: 0000 Size is: 64 SC/SD:
MCB type is: 0176 PSP adress is: 0040 Size is: 256 SC/SD:
MCB type is: 0187 PSP adress is: 0192 Size is: 144 SC/SD:
MCB type is: 0191 PSP adress is: 0192 Size is: 4416 SC/SD: LAB4
MCB type is: 02A6 PSP adress is: 02B1 Size is: 144 SC/SD:
MCB type is: 02B0 PSP adress is: 02B1 Size is: 644320 SC/SD: LAB3
D:\>LAB4.EXE
Interruption has already loaded.
D:\>

```

Рисунок 2 – Демонстрация корректного определения установленного обработчика прерывания при повторном запуске программы.

**Шаг 4.** Программа была запущена с ключом выгрузки, чтобы убедиться, что резидентный обработчик прерывания выгружен и память, занятая резидентом освобождена.

```

D:\>LAB4.EXE
Interruption has already loaded.
D:\>LAB4.EXE \un
Interruption has already loaded.
D:\>LAB4.EXE /un
Interruption was unloaded.
D:\>LAB3.COM
Available memory:
648912
Extended memory:
245760
MCB type is: 016F PSP adress is: 0000 Size is: 16 SC/SD:
MCB type is: 0171 PSP adress is: 0000 Size is: 64 SC/SD:
MCB type is: 0176 PSP adress is: 0040 Size is: 256 SC/SD:
MCB type is: 0187 PSP adress is: 0192 Size is: 144 SC/SD:
MCB type is: 0191 PSP adress is: 0192 Size is: 648912 SC/SD: LAB3
D:\>

```

Рисунок 3 – Демонстрация корректной выгрузки резидентного обработчика прерываний.

## **Выводы.**

Построен собственный обработчик прерываний сигналов таймера. Получены дополнительные знания о работе с памятью (резидентный обработчик может быть загружен и выгружен из памяти).

## **Вопросы.**

### ***1. Как реализован механизм прерывания от часов?***

Прерывание INT 1Ch вызывается обработчиком аппаратного прерывания от таймера INT 08h приблизительно 18,2 раза в секунду.

Сначала запоминаются значения регистров, определяется смещение по номеру источника прерывания в таблице векторов (2 байта в IP, два – в CS).

Вызывается обработчик прерывания по сохраненному адресу.

В конце управление передается обратно от обработчика прерывания к прерванной программе.

### ***2. Какого типа прерывания использовались в работе?***

1Ch – аппаратное прерывание, 10h и 21h – программное прерывание, вызываемое командой int.