

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. 8383

Мирсков А.А.

Преподаватель

Губкин А.Ф.

Санкт-Петербург

2020

Цель работы.

Построить обработчик прерываний сигналов таймера. Изучить способы загрузки и выгрузки резидентной программы в память.

Выполнение работы.

Шаг 1. Был написан и отлажен программный модуль типа .EXE, который выполняет 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 21h.
- 3) Если прерывание установлено, то выводится соответствующее сообщение и осуществляется выход по функции 4Ch прерывания 21h.
- 4) Выгрузка прерывания по соответствующему значению параметра в командной строке /un. Затем осуществляется выход по функции 4Ch прерывания 21h.

Программа содержит код устанавливаемого прерывания в виде удаленной процедуры. Он выполняет следующие функции:

- 1) Сохраняет значения регистров в стеке при входе и восстанавливает при выходе
- 2) При выполнении тела процедуры накапливает общее суммарное число прерываний и выводит на экран.

Карта памяти до запуска программы представлена на рис. 1.

```
OWNER ADDRESS 0008  
SIZE 16  
  
4D  
OWNER ADDRESS 0000  
SIZE 64  
  
4D  
OWNER ADDRESS 0040  
SIZE 256  
  
4D  
OWNER ADDRESS 0192  
SIZE 144  
  
5A  
OWNER ADDRESS 0192  
SIZE 648912  
LAB3
```

Рисунок 1 — Карта памяти до запуска

Шаг 2. Результат запуска программы представлен на рис. 2. На рис. 3 изображена карта памяти, на которой видно, что прерывание разместилось в памяти.

```
C:\>lab4      Interrupt number 195  
C:\>
```

Рисунок 2 — Результат запуска программы

```

4D
OWNER ADDRESS 0008
SIZE 16

4D
OWNER ADDRESS 0000
SIZE 64

4D
OWNER ADDRESS 0040
SIZE 256

4D
OWNER ADDRESS 0192
SIZE 144

4D
OWNER ADDRESS 0192
SIZE 1072
LAB4

4D
OWNER ADDRESS 01E0
SIZE 144

5A
OWNER ADDRESS 01E0
SIZE 647664
LAB3

```

Рисунок 3 - Карта памяти после запуска

Шаг 3. Программа была запущена ещё раз. На рис. 4 видно, что программа определила, что обработчик уже установлен и вывела соответствующее сообщение.

```

C:\>lab4
Already loaded Interrupt number 4569
C:\>_

```

Рисунок 4 — Результат повторного запуска программы

Шаг 4. Программа была запущена с ключом выгрузки. На рис. 5 видно, что сообщение с количеством прерываний не появилось. На рис. 6 видно, что память, занятая резидентом освободилась.

```

C:\>lab4 /un
C:\>_

```

Рисунок 5 — Результат запуска программы с ключом /un

```
4D
OWNER ADDRESS 0008
SIZE 16

4D
OWNER ADDRESS 0000
SIZE 64

4D
OWNER ADDRESS 0040
SIZE 256

4D
OWNER ADDRESS 0192
SIZE 144

5A
OWNER ADDRESS 0192
SIZE 648912
LAB3
```

Рисунок 6 — Карта памяти после выгрузки

Ответы на контрольные вопросы.

1) Как реализован механизм прерывания от часов?

Компьютер содержит устройство, называемое системным таймером. Это устройство вырабатывает прерывание INT 8h приблизительно 18,2 раза в секунду.

При инициализации BIOS устанавливает свой обработчик для прерывания таймера. Этот обработчик каждый раз увеличивает на 1 текущее значение четырехбайтовой переменной, располагающейся в области данных BIOS по адресу 0000:046Ch - счетчик тиков таймера. Если этот счетчик переполняется (прошло более 24 часов с момента запуска таймера), в ячейку 0000:0470h заносится 1.

Другое действие, которое выполняет обработчик прерывания таймера - вызов прерывания INT 1Ch. После инициализации системы вектор INT 1Ch указывает на команду IRET, т.е. ничего не выполняется. Программа может установить собственный обработчик этого прерывания для того чтобы выполнять какие-либо периодические действия.

2) Какого типа прерывания использовались в работе?

Програмные 21h и 10h. Аппаратное 1Ch.

Выводы.

В ходе лабораторной работы был построен обработчик прерываний сигналов таймера. Были изучены способы загрузки и выгрузки резидентной программы в память.