

**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.**

Студент гр. 7381

\_\_\_\_\_

Лукашев Р.С.

Преподаватель

\_\_\_\_\_

Ефремов М.А.

Санкт-Петербург

2019

## **Цель работы.**

Необходимо реализовать обработчик прерываний сигналов таймера. Сигналы таймера генерируются машиной через некоторые 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 о 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.

### Ход работы.

- 1) Состояние таблицы MCB до загрузки прерывания представлено на рис. 1.

```
C:\>LAB3_1.COM
Accesible memory size - 648912 B
Extended memory size - 15360 KB
MCB table:
Address|  Type|  Owner|  Size| Name
016F|    4D|  0008|    16|
0171|    4D|  0000|    64| DPMILOAD
0176|    4D|  0040|   256|
0187|    4D|  0192|   144|
0191|    5A|  0192| 648912| LAB3_1
```

Рисунок 1 – состояние памяти до загрузки прерывания

При загрузке резидента на экран выводится количество вызовов прерывания (см. рис. 2).

```
C:\>LAB_4.EXE
Loading resident                                Interruption count: 0078
```

Рисунок 2 – Результат загрузки прерывания

3. Состояние таблицы MCB после загрузки прерывания представлено на рис.

```
C:\>LAB3_1.COM
Accesible memory size - 647632 B
Extended memory size - 15360 KB
MCB table:
Address|  Type|  Owner|  Size| Name
016F|    4D|  0008|    16|
0171|    4D|  0000|    64| DPMILOAD
0176|    4D|  0040|   256|
0187|    4D|  0192|   144|
0191|    4D|  0192|  1104| LAB_4
01D7|    4D|  01E2|   144|
01E1|    5A|  01E2| 647632| LAB3_1
Interruption count: 1504
```

Рисунок 3 – состояние памяти после загрузки резидента.

- 2) При попытке повторной загрузки прерывания выводится сообщение о том, что резидент был уже загружен (см. рис. 4).

```
C:\>LAB_4.EXE
Resident has been loaded already          Interrupt count: 4186
```

Рисунок 4 – попытка повторной загрузки прерывания

3) Выгрузка резидента показана на рисунке 5.

```
C:\>LAB_4.EXE /un
Unloading resident
```

Рисунок 5 – выгрузка резидента

Память действительно освободилась, как можно увидеть на рисунке 6.

```
C:\>LAB3_1.COM
Accesible memory size -    648912 B
Extended memory size -    15360 KB
MCB table:
Address!   Type!   Owner!   Size!   Name
  016F!     4D!     0008!     16!
  0171!     4D!     0000!     64!  DPMILOAD
  0176!     4D!     0040!    256!
  0187!     4D!     0192!    144!
  0191!     5A!     0192!  648912! LAB3_1
```

Рисунок 6 – состояние памяти после выгрузки резидента

### Ответы на контрольные вопросы.

1. Как реализован механизм прерывания от часов?
  - Прерывание по таймеру вызывается автоматически при каждом тике аппаратных часов раз в 55 миллисекунд. Сохраняется изначальное содержимое регистров, затем определяется источник прерывания, и по его номеру определяется смещение в таблице векторов прерывания. Далее происходит сохранение CS:IP, после чего управление передается по этому адресу, затем происходит выполнение обработчика прерывания. После выполнения обработчика происходит возврат управления прерванной программе.
2. Какого типа прерывания использовались в работе?
  - Аппаратные прерывания (1Ch), прерывания функций DOS (21h), и прерывания функций BIOS (10h).

## **Вывод.**

В ходе лабораторной работы был построен обработчик прерывания от сигналов таймера. Изучены дополнительные функции работы с памятью: установка программы-резидента и его выгрузка из памяти.