

**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. 7383

\_\_\_\_\_

Ласковенко Е.А.

Преподаватель

\_\_\_\_\_

Ефремов М.А.

Санкт-Петербург

2019

### **Цель работы.**

В архитектуре компьютера существуют стандартные прерывания, за которыми закреплены определенные вектора прерываний. Вектор прерывания 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 предлагается построить обработчик прерываний сигналов таймера. Эти сигнал 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.

Таблица 1. Описание функций.

<b>Название функции</b>	<b>Назначение</b>
ROUT	пользовательский обработчик прерывания
BYTE_TO_HEX	перевод байта в AL в два числа в 16-ой с/с в AX, в AL старшая цифра, в AH младшая
TETR_TO_HEX	вспомогательная функция для работы функции BYTE_TO_HEX
WRD_TO_HEX	перевод в 16с/с 16-ти разрядного числа, в AX - число, DI - адрес последнего символа
PRINT_STR	вывод строки на экран
CHECK_ROUT	проверка, установлен ли пользовательский вектор прерывания
SET_ROUT	выставление в вектор прерывания пользовательского обработчика ROUT
MAKE_RESIDENT	установка обработчика как резидента
UNLOAD_ROUT	выгрузка пользовательского обработчика прерывание,

	восстановление старого обработчика
MAIN	головная функция, содержащая в себе все необходимые обращения к функциям

Таблица 2. Описание структур данных.

Название	Тип	Назначение
CHECK_UNIQUE	db	строка, содержащая сигнатуру, идентифицирующая резидент
ROUT_LOAD_MESSAGE	db	строка, информирующая об успешной загрузке обработчика
ROUT_CHECK_MESSAGE	db	строка, информирующая о том, что обработчик уже установлен
UNLOAD_MESSAGE	db	строка, информирующая о том, что обработчик еще не загружен
UNLOAD_ROUT_MESSAGE	db	строка, информирующая об успешной выгрузке обработчика
UNLOAD_TAIL	db	строка, содержащая хвост командной строке на выгрузку
TAIL	db	строка, в которую будет записан хвост командной строки
KEEP_CS	dw	переменная, хранящая старый адрес сегмента кода
KEEP_IP	dw	переменная, хранящая старый адрес смещения обработчика
UNIQUE	db	сигнатура,

		идентифицирующая резидент
KEEP_PSP		переменная, хранящая адрес PSP
KEEP_SP	dw	переменная, хранящая старый указатель стека
KEEP_SS	dw	переменная, хранящая старый адрес сегмента стека
COUNTER	dw	переменная - счетчик
COUNT_MESSAGE	db	строка, содержащая количество вызовов обработчика
ROUT_STACK	dw	область памяти под стек резидента
stack_ptr		указатель на стек резидента

### **Выполнение программы.**

Была написана программа, которая выполняет следующие действия:

- Проверяет, установлено ли пользовательское прерывание с вектором 1Ch
- Устанавливает резидентную функцию для обработки прерывания и настраивает вектор прерываний, если прерывание не установлено, и осуществляет выход по функции 4Ch прерывания int 21h
  - Если прерывание установлено, то выводится соответствующее сообщение и осуществляется выход по функции 4Ch прерывания int 21h
  - 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.

## Результат работы программы.

```
ROUT CALLED: 0078
C:\>LR4.EXE
Rout is loaded.

C:\>LR3_1.COM
Available memory: 644320 B
Extended memory: 15360 KB
MCB Chain:
ADDRESS  OWNER    SIZE      NAME
 016F    0008        16
 0171    0000        64  DPMILOAD
 0176    0040       256
 0187    0192       144
 0191    0192      4416  LR4
 02A6    02B1      4144
 02B0    02B1  644320  LR3_1

C:\>
```

Рисунок 1 – Результат первоначального выполнения программы lr4.exe без ключа выгрузки и lr3\_1.com для отображения блоков MCB

```
ROUT CALLED: 0444
C:\>LR3_1.COM
Available memory: 644320 B
Extended memory: 15360 KB
MCB Chain:
ADDRESS  OWNER    SIZE      NAME
 016F    0008        16
 0171    0000        64  DPMILOAD
 0176    0040       256
 0187    0192       144
 0191    0192      4416  LR4
 02A6    02B1      4144
 02B0    02B1  644320  LR3_1

C:\>LR4.EXE
Rout has been already loaded.

C:\>♥
```

Рисунок 2 – Результат повторного выполнения программы lr4.exe без ключа выгрузки

```
C:\>LR4.EXE
Rout has been already loaded.
```

```
C:\>LR4.EXE /un
Rout is unloaded.
```

```
C:\>LR3_1.COM
Available memory: 648912 B
Extended memory: 15360 KB
MCB Chain:
ADDRESS  OWNER   SIZE    NAME
  016F    0008     16
  0171    0000     64  DPMILOAD
  0176    0040    256
  0187    0192    144
  0191    0192  648912  LR3_1
```

```
C:\>
```

Рисунок 3 – Результат выполнения программы lr4.exe с ключом выгрузки и lr3\_1.com для отображения блоков MCB

```
C:\>LR4.EXE /un
User rout has not been loaded yet.
```

```
C:\>_
```

Рисунок 4 – Результат повторного выполнения программы lr4.exe с ключом выгрузки

### **Выводы.**

В процессе выполнения данной лабораторной работы был построен пользовательский обработчик прерываний сигналов таймера.

### **Ответы на контрольные вопросы.**

1) Как реализован механизм прерывания от часов?

При каждом такте часов происходит следующее: сохраняется содержимое регистров, определяется источник прерывания, по номеру источника прерывания определяется смещение в таблице векторов прерываний, далее эти данные помещаются в CS:IP, запускается обработчик прерывания по этому адресу (CS:IP), происходит возврат управления прерванной программе.

2) Какого типа прерывания использовались в работе?

В работе были использованы пользовательское прерывание 1Ch и программные прерывания int 10h и int 21h.