

**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

\_\_\_\_\_

Васильев А.И.

Преподаватель

\_\_\_\_\_

Ефремов М.А.

Санкт-Петербург

2019

### **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.

Таблица 1 – Описание функций.

<b>Название функции</b>	<b>Назначение</b>
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
ROUT	пользовательский обработчик прерываний, считающий и печатающий количество его вызовов
outputBP	функция вывода строки по адресу ES:BP
outputAL	функция вывода символа из AL
WriteStr	вывод строки на экран
SET_RESIDENT	установка резидента
SET_ROUT	выставление в вектор прерывания пользовательского обработчика ROUT
UNLOAD_ROUT	выгрузка пользовательского

	обработчика прерывания, восстановление старого обработчика
CHECK_TAIL	проверка хвоста командной строки
IS_LOADED	действия при загруженном резиденте

Таблица 2. Описание структур данных.

Название функции	Тип	Назначение
SIGNATURE_CHECK	db	строка для проверки сигнатуры
CHECK_MESSAGE	db	строка, информирующая о том, что резидент был загружен ранее
LOAD_MESSAGE	db	строка, информирующая о том, что загрузка резидента прошла успешно
UNLOAD_MESSAGE	db	строка, информирующая о том, что резидент был успешно удален
UNLOAD_ERROR	db	строка, информирующая о том, что при удалении резидента произошла ошибка
UNLOAD_TAIL	db	хвост командной строки, при котором резидент необходимо удалить
TAIL	db	хвост командной строки, при котором резидент необходимо загрузить
KEEP_PSP	dw	сохранение PSP
KEEP_SS	dw	сохранение регистров
KEEP_SP	dw	
KEEP_CS	dw	
KEEP_IP	dw	сохранение IP
SIGNATURE	db	сигнатура для проверки
COUNTER	dw	счетчик
COUNT_MESSAGE	db	сообщение счетчика

## Результат работы программы.

```
C:\>LR4.EXE
Rout was successfully loaded.

C:\>L3_1.COM
ROUT CALLED: 0374
Available memory: 644288 B
Extended memory: 15360 KB
Address Owner Size Name
016F 0008 16
0171 0000 64
0176 0040 256
0187 0192 144
0191 0192 4448 LR4
02A8 02B3 4144
02B2 02B3 644288 L3_1

C:\>_
```

Рисунок 1 – Результат выполнения программы lr4.exe и l3\_1.com, загрузка резидента

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

Рисунок 2 – Результат выполнения программы lr4.exe, попытка повторно загрузить резидент

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

C:\>lr4.exe /un
Rout was successfully unloaded.

C:\>L3_1.COM
Available memory: 648912 B
Extended memory: 15360 KB
Address Owner Size Name
016F 0008 16
0171 0000 64
0176 0040 256
0187 0192 144
0191 0192 648912 L3_1

C:\>_
```

Рисунок 3 – Результат выполнения программы lr4.exe, выгрузка резидента

```
C:\>lr4.exe /un
Rout is not loaded.

C:\>_
```

Рисунок 4 – Результат выполнения программы lr4.exe, попытка повторно выгрузить резидент

### **Выводы.**

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

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

1) Как реализован механизм прерывания от часов?

Сначала содержимое регистров сохраняется, затем определяется источник прерывания, по номеру которого определяется смещение в таблице векторов прерывания. Данные вектора прерывания помещаются в CS : IP. Управление передаётся по адресу CS:IP и происходит выполнение обработчика. В конце происходит возврат управления прерванной программе. Прерывания генерируются системным таймером.

2) Какого типа прерывания использовались в работе?

В работы использовались пользовательское прерывание по таймеру 1Ch и программные прерывания 21h и 10h.