

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра математического обеспечения и применения ЭВМ**

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

Студент гр. 7383

\_\_\_\_\_

Александров Р.А.

Преподаватель

\_\_\_\_\_

Ефремов М.А.

Санкт-Петербург

2019

### **Цель работы.**

Изучение внутренней структуры обработчиков прерываний, методов прямой взаимодействия с ними и построение резидентного обработчика прерываний сигналов таймера.

### **Постановка задачи.**

Таблица 1 – Сведения о функциях программы

Функции программы	Описание функций
PRINT	Вывод строки в консоль
ROUT	Обработчик прерывания
outputBP	Вывод строки по адресу ES:BP на экран
CHECK_ROUT	Проверяет, установлен ли обработчик прерывания
SET_ROUT	Устанавливает обработчик прерывания
DEL_ROUT	Удаляет обработчик прерывания
SAVE_STAND	Сохраняет адрес стандартного обработчика
TETR_TO_HEX	Перевод числа из 2 с/с в 16 с/с
BYTE_TO_HEX	Перевод байта из 2 с/с в 16 с/с
WRD_TO_HEX	Перевод слова из 2 с/с в 16 с/с
BYTE_TO_DEC	Перевод байта из 2 с/с в 10 с/с
BEGIN	Начало работы программы

Таблица 2 – Сведения о структурах данных программы

Название	Тип	Назначение
SET_INTERRUPT	db	Setup interrupt
DEL_INTERRUPT	db	Uninstall interrupt
ALREADY_SET	db	Interrupt is already set
NOT_SET	db	Interrupt is not set

Таблица 3 – Последовательность действий, выполняемых утилитой

Номер действия	Что делает
1	Устанавливает резидентный обработчик прерывания
2	Выгружает резидентный обработчик прерывания
3	Проверяет, что обработчик установлен

### Ход работы программы.

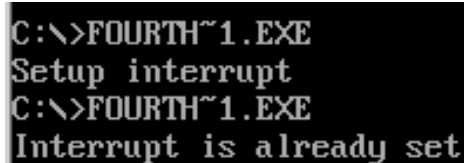
1. На рис. 1 представлен результат загрузки обработчика прерывания.



```
C:\>FOURTH~1.EXE
Setup interrupt
```

Рисунок 1 – Загрузка обработчика

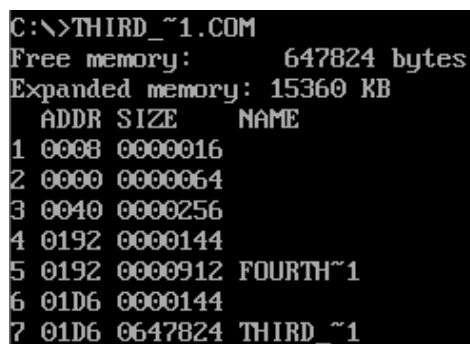
2. На рис. 2 представлен результат повторной загрузки обработчика прерывания.



```
C:\>FOURTH~1.EXE
Setup interrupt
C:\>FOURTH~1.EXE
Interrupt is already set
```

Рисунок 2 – Повторная загрузка обработчика

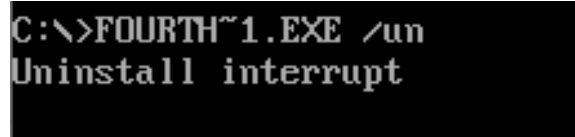
3. На рис. 3 видно размещение обработчика в памяти. Используется загрузочный модуль third\_lab\_1.com из л/р №3.



```
C:\>THIRD_~1.COM
Free memory:      647824 bytes
Expanded memory: 15360 KB
  ADDR  SIZE  NAME
  ----  -
1 0008 0000016
2 0000 0000064
3 0040 0000256
4 0192 0000144
5 0192 0000912 FOURTH~1
6 01D6 0000144
7 01D6 0647824 THIRD_~1
```

Рисунок 3 – Размещение в памяти

4. На рис. 4 представлен результат выгрузки обработчика прерывания.



```
C:\>FOURTH~1.EXE /un
Uninstall interrupt
```

Рисунок 4 – Выгрузка обработчика прерывания

### **Выводы.**

В ходе лабораторной работы был исследован механизм работы резидентного обработчика прерывания, создан и протестирован загрузочный модуль, реализующий исследуемую функциональность ОС.

## **ПРИЛОЖЕНИЕ А**

### **ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ**

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

Сначала сохраняется содержимое регистров, потом определяется источник прерывания, по номеру которого определяется смещение в таблице векторов прерывания, сохраняется в CS: IP, передаётся управление по адресу CS:IP и происходит выполнение обработчика, и в конце происходит возврат управления прерванной программе. Аппаратное прерывание от таймера происходит каждые 55 мс.

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

Аппаратные прерывания (1Ch); прерывания функций DOS (21h); прерывания функций BIOS (10h).