

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

**ОТЧЕТ**  
**по лабораторной работе № 4**  
**по дисциплине «Операционные системы»**  
**Тема: «Обработка стандартных исключений»**

Студент гр. 8381

\_\_\_\_\_

Гоголев Е.Е.

Преподаватель

\_\_\_\_\_

Ефремов М.А.

Санкт-Петербург

2020

### **Цель работы.**

Изучение обработки прерываний сигналов таймера, которые генерируются аппаратурой примерно 18.2 раза в секунду, вызывая прерывание с соответствующим вектором.

### **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 по 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 по соответствующему значению параметра в командной строке /un. Выгрузка прерывания состоит в восстановлении стандартного вектора прерываний и освобождения памяти, занимаемой резидентом. Затем осуществляется выход по функции 4Ch прерывания int 21h.

Для того, чтобы проверить установку прерывания, можно поступить следующим образом. Прочитать адрес, записанный в векторе прерывания. Предположим, что этот адрес указывает на точку входа в установленный резидент. На определённом, известном смещении в теле резидента располагается сигнатура, некоторый код, который идентифицирует резидент. Сравнив известное значение сигнатуры с реальным кодом, находящимся в резиденте, можно определить, установлен ли резидент. Если значения

совпадают, то резидент установлен. Длина кода сигнатуры должна быть достаточной, чтобы сделать случайное совпадение маловероятным.

Программа должна содержать код устанавливаемого прерывания в виде удалённой процедуры. Этот код будет работать после установки при возникновении прерывания. Он должен выполнять следующие функции:

1. Сохранить значение регистров в стеке при входе и восстановить их при выходе.
2. При выполнении тела процедуры накапливать общее суммарное число прерываний и выводить на экран. Для вывода на экран следует использовать прерывание `int 10h`, которое позволяет непосредственно выводить информацию на экран.

**Шаг 2.** Запустите отлаженную программу и убедитесь, что резидентный обработчик прерывания `1Ch` установлен. Работа прерывания должна отображаться на экране, а также необходимо проверить размещение прерывания в памяти. Для этого запустите программу ЛР 3, которая отображает карту памяти в виде списка блоков МСВ. Полученные результаты поместите в отчёт.

**Шаг 3.** Запустите отлаженную программу ещё раз и убедитесь, что программа определяет установленный обработчик прерываний. Полученные результаты поместите в отчёт.

**Шаг 4.** Запустите отлаженную программу с ключом выгрузки и убедитесь, 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. Полученные результаты поместите в отчёт.

### **Необходимые сведения для составления программы.**

Резидентные обработчики прерываний — это программные модули, которые вызываются при возникновении прерываний определённого типа (сигнал таймера, нажатие клавиши и т.д.), которым соответствуют

определённые вектора прерывания. Когда вызывается прерывание, процессор переключается на выполнение кода обработчика, а затем возвращается на выполнение прерванной программы. Адрес возврата в прерванную программу (CS:IP) запоминается в стеке вместе с регистром флагов. Затем в CS:IP загружается адрес точки входа программы обработки прерывания и начинает выполняться его код. Обработчик прерывания должен заканчиваться инструкцией IRET (возврат из прерывания).

Вектор прерывания имеет длину 4 байта. В первом хранится значение IP, во втором — CS. Младшие 1024 байта памяти содержат 256 векторов. Вектор для прерывания 0 начинается с ячейки 0000:0000, для прерывания 1 — с ячейки 0000:0004 и т.д.

Для установки написанного прерывания используется функция 25H прерывания 21H, которая устанавливает вектор прерывания на указанный адрес.

Программа, выгружающая обработчик прерываний, должна восстанавливать оригинальные векторы прерываний. Функция 35H прерывания 21H позволяет восстановить значение вектора прерывания, помещая значение сегмента в ES, а смещение в BX.

Для того, чтобы оставить процедуру прерывания резидентной в памяти, следует воспользоваться функцией DOS 31H прерывания int 21H. Эта функция оставляет память, размер которой указывается в качестве параметра, занятой, а остальную память освобождает, и осуществляет выход в DOS.

Вывод на экран информации обработчиком прерываний осуществляется с помощью функций прерывания 10H.

### **Описание программы.**

В результате выполнения лабораторной работы была написана программа, описание функций которой представлено ниже.

- INTERRUPT - резидентный обработчик прерывания, сохраняет и запоминает количество обработанных прерываний;
- LOAD - загрузка прерывания 1Ch;
- UNLOAD - выгрузка прерывания 1Ch;
- INT\_CHECK - проверка того, установлено ли прерывание
- TAIL\_CHECK - проверка хвоста командной строки
- FLAGS\_CHECK - проверка состояния программы (загружено ли прерывание и хвост командной строки)
- PRINT\_STRING - вывод строки из DX и перевода строки на экран;

### Ход работы

Написание исходного кода производилось в редакторе Visual Studio Code на базе операционной системы Windows 10, сборка и отладка производились в эмуляторе DOSBox.

```
F:\GOGOLE~4>int.exe
SUCCESS. Interruption loaded
85 interruptions
```

Рисунок 1 — Вывод программы INT.EXE после первого запуска

```
F:\GOGOLE~4>exec.com
--- Available memory (kilobytes): 640
--- Extended memory (kilobytes): 15360
----- MCB blocks -----
| Address | Type | PSP Address | Size(kb) | Name |
|-----|-----|-----|-----|-----|
| 016F | 4D | 0008 | 16 | 
| 0171 | 4D | 0000 | 64 | 
| 0176 | 4D | 0040 | 256 | 
| 0187 | 4D | 0192 | 144 | 
| 0191 | 4D | 0192 | 720 | INT |
| 01BF | 4D | 01CA | 144 | 
| 01C9 | 5A | 01CA | 648016 | EXEC |
1736 interruptions
```

Рисунок 2 — Вывод программы EXEC.COM после выполнения INT.EXE

Как видно из рисунка, процедура прерывания осталась резидентной в памяти.

```
F:\GOGOLE~4>int.exe
SUCCESS. Interruption loaded
93 interruptions
F:\GOGOLE~4>int.exe
ERROR. Already loaded
181 interruptions
```

Рисунок 3 — Вывод программы INT.EXE при повторном запуске

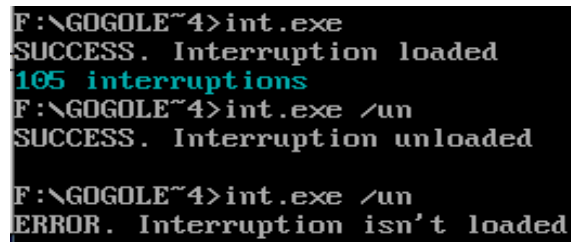
На рисунке 3 показано, что при повторном запуске программа выводит сообщение о том, что резидентный обработчик уже загружен.

```
F:\GOGOLE~4>int.exe
ERROR. Already loaded
1955 interruptions
F:\GOGOLE~4>int.exe /un
SUCCESS. Interruption unloaded

F:\GOGOLE~4>exec.com
--- Available memory (kilobytes): 640
--- Extended memory (kilobytes): 15360
----- MCB blocks -----
| Address | Type | PSP Address | Size(kb) | Name      |
|-----|-----|-----|-----|-----|
| 016F | 4D | 0008 | 16 | 
| 0171 | 4D | 0000 | 64 | 
| 0176 | 4D | 0040 | 256 | 
| 0187 | 4D | 0192 | 144 | 
| 0191 | 5A | 0192 | 648912 | EXEC      |
```

Рисунок 4 - Вывод программы EXEC.COM после выполнения INT.EXE с ключом выгрузки

Из рисунка 4 видно, что после выгрузки резидентного обработчика из памяти вся занятая им память была освобождена.



```
F:\GOGOLE~4>int.exe
SUCCESS. Interruption loaded
105 interruptions
F:\GOGOLE~4>int.exe /un
SUCCESS. Interruption unloaded
F:\GOGOLE~4>int.exe /un
ERROR. Interruption isn't loaded
```

Рисунок 5 — Вывод программы int.exe при повторном запуске с ключом выгрузки

Как видно из рисунка 5, при выгрузке резидентного обработчика было выведено сообщение, а также при запросе повторной выгрузки было показано, что резидентный обработчик не загружен.

### **Вывод.**

В результате выполнения данной лабораторной работы была изучена работа прерываний от системного таймера, прерывания 10h, а механизм загрузки и выгрузки резидентных обработчиков прерываний а также управление ходом программы через хвост командной строки.

## **Контрольные вопросы.**

### **Как реализован механизм прерывания от часов:**

Аппаратное прерывание `int 8h` вызывается 18.2 раза в секунду. Стандартный обработчик этого прерывания увеличивает счётчик и вызывает другое прерывание — `1Ch`. По умолчанию оно указывает на команду `IRET`. Во время выполнения этих двух прерываний не вызываются другие.

### **Какого типа прерывания использовались в программе:**

В программе использовались по большей части программные прерывания, такие как `int 21h` и `10h`. Написанный обработчик применялся к асинхронному аппаратному прерыванию, `1Ch`, прерыванию от таймера.