

**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(ка) гр. 0381

Ионина К.С.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

### **Цель работы.**

В архитектуре компьютера существуют стандартные прерывания, за которыми закреплены определенные вектора прерываний. Вектор прерываний 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 № 4 предлагается построить обработчик прерываний сигналов таймера. Эти сигналы 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.

### **Задание.**

**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 освобождении 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 определенном, 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. Длину кода сигнатуры должна быть достаточной, чтобы сделать случайное совпадение маловероятным.

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

- 1) Сохраняет стек прерванной программы (регистры SS и SP) в рабочих переменных и восстановить при выходе.
- 2) Организовать свой стек.
- 3) Сохранить значения регистров в стеке при входе и восстановить их при выходе.
- 4) 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.
- 5) Функция прерывания должна содержать только переменные, которые она использует.

**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 отчет.

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

**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 отчет.

**Шаг 5.** Ответьте на контрольные вопросы.

### **Выполнение работы.**

Сначала были объявлены строки для вывода информации:

```
ALREADY_LOAD DB 'Interrupt already loaded', 0DH, 0AH, '$'
```

```
SET_STR DB 'Interrupt set', 0DH, 0AH, '$'
```

```
RESTORE_STR DB 'Interruption restored', 0DH, 0AH, '$'
```

Далее была реализована процедура USER\_CHECK, которая проверяет, был ли установлен пользовательский обработчик прерываний. В регистр AX записывается значение слова SIGNATURE и происходит сравнение с нужным значением. Если значения совпадают, в AL заносится 1, иначе 0.

Процедура STAND\_INTERRUPT восстанавливает стандартное прерывание. Вызов происходит только в том случае, если модуль запущен с параметром «/un».

Процедура CONTROL\_END на основе значения, лежащего со смещением 81h в PSP, проверяет наличие параметра «/un» в конце запускающей программы.

Процедура INTERRUPTION вызывается, если пользовательское прерывание не установлено. В итоге на экран выводится сообщение «Interrupt set». В случае, если пользовательское прерывание уже установлено, происходит проверка на наличие параметра «/un». Если параметр есть, на экран выводится «Interrupt already loaded», иначе «Interruption restored».

Процедура PRINT выводит в консоль строку.

Процедура COUNTER\_UPDATE обновляет счётчик прерываний в строке STR\_COUNT.

### **Тестирование.**

При первом вызове программа устанавливает пользовательское прерывание. Счётчик отображается в левом верхнем углу экрана.

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
Count of interrupts: 0206er Version 5.10
Copyright (C) Microsoft Corp 1981, 1988. All rights reserved.

Object filename [lr4.OBJ]:
Source listing [NUL.LST]: gfd
Cross-reference [NUL.CRF]:

47838 + 453277 Bytes symbol space free

0 Warning Errors
0 Severe Errors

F:\>link lr4.obj

Microsoft (R) Overlay Linker Version 3.64
Copyright (C) Microsoft Corp 1983-1988. All rights reserved.

Run File [LR4.EXE]:
List File [NUL.MAP]: fgds
Libraries [.LIB]:

F:\>lr4.exe
User interrupt set
```

Рисунок 1. Вызов lr4.exe.

При запуске лабораторной работы №3 счетчик продолжает работать, программа из текущей лабораторной работы загружена в память.

```
F:\>lr3.com

Size of available memory:      64 byte
Size of extended memory: 15728640 byte
MCB: 4Dh  Address: 016Fh  Owner: MS DOS  Area size: 16 byte
SC/SD:
MCB: 4Dh  Address: 0171h  Owner: Free    Area size: 64 byte
SC/SD:
MCB: 4Dh  Address: 0176h  Owner: 0040h   Area size: 256 byte
SC/SD:
MCB: 4Dh  Address: 0187h  Owner: 0192h   Area size: 144 byte
SC/SD:
MCB: 4Dh  Address: 0191h  Owner: 0192h   Area size: 2624 byte
SC/SD: LR4
MCB: 4Dh  Address: 0236h  Owner: 0241h   Area size: 144 byte
SC/SD:
MCB: 5Ah  Address: 0240h  Owner: 0241h   Area size: 646112 byte
SC/SD: LR3

F:\>_
```

Рисунок 2. Вызов lr3.com.

При повторном запуске программы на экран выводится сообщение о том, что обработчик уже установлен.

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
Count of interrupts: 1497
F:\>lr3.com

Size of available memory:      64 byte
Size of extended memory: 15728640 byte
MCB: 4Dh  Address: 016Fh  Owner: MS DOS  Area size: 16 byte
SC/SD:
MCB: 4Dh  Address: 0171h  Owner: Free    Area size: 64 byte
SC/SD:
MCB: 4Dh  Address: 0176h  Owner: 0040h   Area size: 256 byte
SC/SD:
MCB: 4Dh  Address: 0187h  Owner: 0192h   Area size: 144 byte
SC/SD:
MCB: 4Dh  Address: 0191h  Owner: 0192h   Area size: 2624 byte
SC/SD: LR4
MCB: 4Dh  Address: 0236h  Owner: 0241h   Area size: 144 byte
SC/SD:
MCB: 5Ah  Address: 0240h  Owner: 0241h   Area size: 646112 byte
SC/SD: LR3

F:\>lr4.exe
Interrupt already loaded
```

Рисунок 3. Вызов lr4.exe.

При вызове программы с параметром «/un» на экран выводится сообщение о восстановлении стандартного прерывания, очищается память, счётчик прерываний пропадает с экрана консоли.

```
F:\>lr4.exe /un
Interrupt already loaded
Interruption restored

F:\>lr3.com

Size of available memory:      64 byte
Size of extended memory: 15728640 byte
MCB: 4Dh  Address: 016Fh  Owner: MS DOS  Area size: 16 byte
SC/SD:
MCB: 4Dh  Address: 0171h  Owner: Free    Area size: 64 byte
SC/SD:
MCB: 4Dh  Address: 0176h  Owner: 0040h   Area size: 256 byte
SC/SD:
MCB: 4Dh  Address: 0187h  Owner: 0192h   Area size: 144 byte
SC/SD:
MCB: 5Ah  Address: 0191h  Owner: 0192h   Area size: 648912 byte
SC/SD: LR3
```

Рисунок 4. Вызов lr4.exe /un + lr3.com

## **Вывод.**

В ходе выполнения лабораторной работы был построен обработчик прерываний сигналов таймера.

## **Контрольные вопросы по лабораторной работе №4.**

*1) Как реализован механизм прерывания от часов?*

Принимается сигнал прерывания (приходит ~каждые 54мс), запоминаются содержимые регистров, по номеру источника прерывания в таблице векторов определяется смещение, запоминается адрес 2 байта в IP и 2 байта в CS. Далее выполняется прерывание по сохранённому адресу и далее восстанавливается информация прерванного процесса и управление возвращается прерванной программе.

*2) Какого типа прерывания использовались в работе?*

В работе использовались int 10h и 21h – программные прерывания и 1Ch – аппаратное прерывание.