

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. 0382

Азаров М.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.

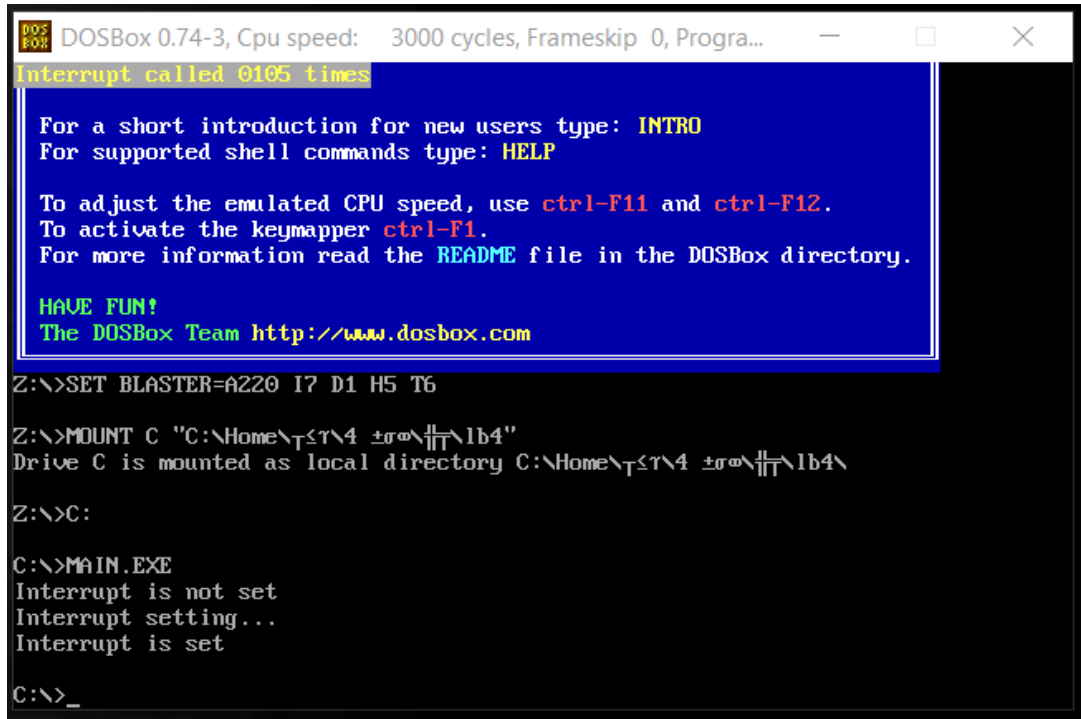
Шаг 5. Ответьте на контрольные вопросы.

Ход работы

1. Для выполнения первого шага понадобилось разработать модуль типа .EXE , который содержит следующие процедуры:

Название процедуры	Что делает
Interrupt	Обработчик прерывания, который накапливает общее суммарное число прерываний и выводит на экран.
Check_set_inter	Проверяет было ли заменено прерывание на пользовательское или нет.
Load_interrupt	Загружает пользовательское прерывание Interrupt вместо стандартного прерывания таймера.
Check_param_un	Проверяет был ли указан параметр /un при загрузке модуля
Unload_interrupt	Выгружает пользовательский модуль.
Main	Главная процедура выполняющая поставленную задачу.

2. После запуска, программа действительно заменяет обработчик прерывания 1Ch на пользовательский. Об этом свидетельствует вывод на экране в левом верхнем углу:



```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
Interrupt called 0105 times

For a short introduction for new users type: INTRO
For supported shell commands type: HELP

To adjust the emulated CPU speed, use ctrl-F11 and ctrl-F12.
To activate the keymapper ctrl-F1.
For more information read the README file in the DOSBox directory.

HAVE FUN!
The DOSBox Team http://www.dosbox.com

Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>MOUNT C "C:\Home\T\4 \u0440\u0430\u0431\u0430\u0442\u0447\u0438\u043a\u0438\u043b\u0430"
Drive C is mounted as local directory C:\Home\T\4 \u0440\u0430\u0431\u0430\u0442\u0447\u0438\u043a\u0438\u043b\u0430\

Z:\>C:

C:\>MAIN.EXE
Interrupt is not set
Interrupt setting...
Interrupt is set

C:\>_
```

Проверим размещение прерывания в памяти, с помощью программы разработанной в лабораторной №3:

До замены прерывания и вызова модуля:

```
Size available memory: 648912 bytes
Size extended memory: 245760 bytes

MCB #01
Address MCB: 016F
Address PSP owner: 0008
Size: 16 bytes
SC/SD:

MCB #02
Address MCB: 0171
Address PSP owner: 0000
Size: 64 bytes
SC/SD:

MCB #03
Address MCB: 0176
Address PSP owner: 0040
Size: 256 bytes
SC/SD:

MCB #04
Address MCB: 0187
Address PSP owner: 0192
Size: 144 bytes
SC/SD:

MCB #05
Address MCB: 0191
Address PSP owner: 0192
Size: 648912 bytes
SC/SD: INFO
```

После замены прерывания:

```
Size available memory: 647808 bytes
Size extended memory: 245760 bytes

MCB #01
Address MCB: 016F
Address PSP owner: 0008
Size: 16 bytes
SC/SD:

MCB #02
Address MCB: 0171
Address PSP owner: 0000
Size: 64 bytes
SC/SD:

MCB #03
Address MCB: 0176
Address PSP owner: 0040
Size: 256 bytes
SC/SD:

MCB #04
Address MCB: 0187
Address PSP owner: 0192
Size: 144 bytes
SC/SD:

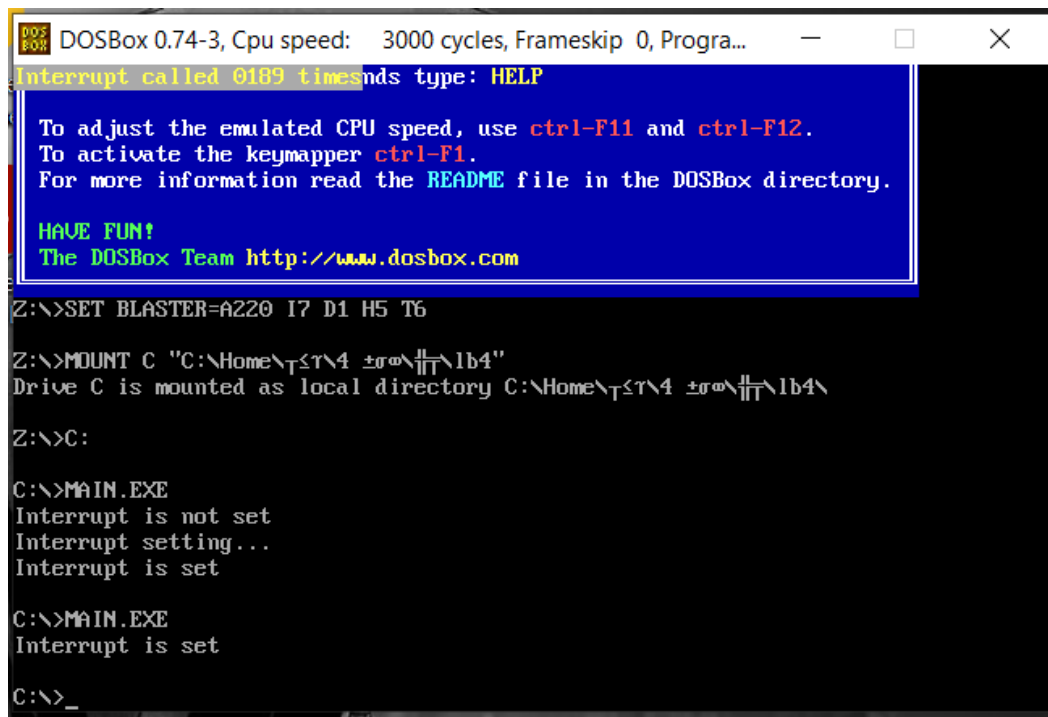
MCB #05
Address MCB: 0191
Address PSP owner: 0192
Size: 928 bytes
SC/SD: MAIN

MCB #06
Address MCB: 01CC
Address PSP owner: 01D7
Size: 144 bytes
SC/SD:

MCB #07
Address MCB: 01D6
Address PSP owner: 01D7
Size: 647808 bytes
SC/SD: INFO
```

Как видим добавилось два новых блока памяти (один для MAIN, другой для пользовательского прерывания).

3. При повторном запуске программы, программа определяет установленный обработчик прерываний и выводит соответствующее сообщение.



```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
Interrupt called 0189 timesnds type: HELP
To adjust the emulated CPU speed, use ctrl-F11 and ctrl-F12.
To activate the keymapper ctrl-F1.
For more information read the README file in the DOSBox directory.
HAVE FUN!
The DOSBox Team http://www.dosbox.com

Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>MOUNT C "C:\Home\T\4 \u\l\4"
Drive C is mounted as local directory C:\Home\T\4 \u\l\4\

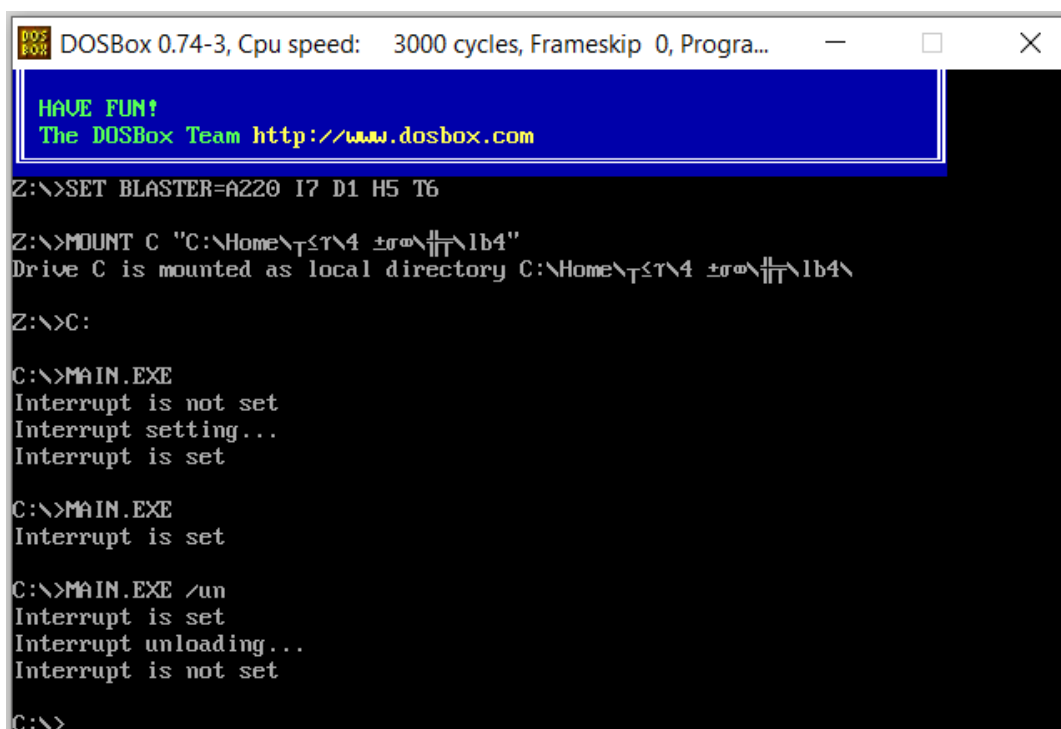
Z:\>C:

C:\>MAIN.EXE
Interrupt is not set
Interrupt setting...
Interrupt is set

C:\>MAIN.EXE
Interrupt is set

C:\>_
```

4. Запустим программу с ключом выгрузки /un и убедимся что обработчик прерывания таймера заменился на стандартный, об этом свидетельствует прекращение вывода сообщений на экран:



```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
HAVE FUN!
The DOSBox Team http://www.dosbox.com

Z:\>SET BLASTER=A220 I7 D1 H5 T6

Z:\>MOUNT C "C:\Home\T\4 \u\l\4"
Drive C is mounted as local directory C:\Home\T\4 \u\l\4\

Z:\>C:

C:\>MAIN.EXE
Interrupt is not set
Interrupt setting...
Interrupt is set

C:\>MAIN.EXE
Interrupt is set

C:\>MAIN.EXE /un
Interrupt is set
Interrupt unloading...
Interrupt is not set

C:\>
```

Проверим, что память занятая пользовательским прерыванием была освобождена, для этого еще раз запустим программу из лабораторной №3:

```
Size available memory: 648912 bytes
Size extended memory: 245760 bytes
```

```
MCB #01
Address MCB: 016F
Address PSP owner: 0008
Size: 16 bytes
SC/SD:
```

```
MCB #02
Address MCB: 0171
Address PSP owner: 0000
Size: 64 bytes
SC/SD:
```

```
MCB #03
Address MCB: 0176
Address PSP owner: 0040
Size: 256 bytes
SC/SD:
```

```
MCB #04
Address MCB: 0187
Address PSP owner: 0192
Size: 144 bytes
SC/SD:
```

```
MCB #05
Address MCB: 0191
Address PSP owner: 0192
Size: 648912 bytes
SC/SD: INFO
```

Как видим все вернулось в изначальное состояние.

Ответы на контрольные вопросы.

1) Как реализован механизм прерывания от часов?

Ответ: При каждом «тике» таймера определяется источник прерывания (по номеру источника прерывания определяется смещение в таблице векторов прерываний), из таблицы определяется адрес обработчика прерывания, передаётся управление по этому адресу, затем происходит обработка прерывания и возврат управления прерванной программе.

2) Какого типа прерывания использовались в работе?

Ответ: В работе использовались аппаратные прерывания (int 1Ch) и программные прерывания (int 21h, int 10h).

Вывод.

В ходе работы были изучены механизмы работы прерываний. Разработана программа которая заменяет стандартный обработчик прерывания таймера 1Ch , на пользовательский, который в свою очередь накапливает общее суммарное число прерываний и выводит их на экран. Также если запустить программу с ключом /un , то программа обратно вернет обработчик прерывания по умолчанию.