

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МОЭВМ

ОТЧЕТ

по лабораторной работе №4
по дисциплине «Операционные системы»
Тема: «Обработка стандартных прерываний»

Студент гр. 8381

Сосновский Д.Н.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2020

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 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.

Основные теоретические положения.

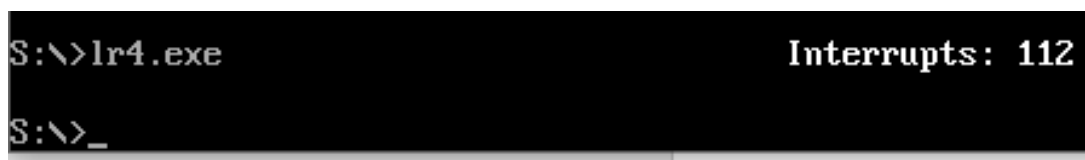
Резидентные обработчики прерываний — это программные модули, которые вызываются при возникновении прерываний определенного типа (сигнал таймера, нажатие клавиши и т.д.), которым соответствуют определённые вектора прерывания. Когда вызывается прерывание, процессор переключается на выполнение кода обработчика, 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.

Ход работы.

Мною был разработан .EXE модуль, 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 int21h.

Результат работы этой программы представлен на рисунке 1.



```
S:\>lr4.exe                               Interrupts: 112
S:\>_
```

Рисунок 1 - работа программы LR4.exe

Если повторно запустить программу, то выведется сообщение о том, что прерывание уже установлено. А если запустить программу с хвостом «/un», то произойдёт выгрузка резидентного обработчика прерываний. Результаты выполнения этих примеров приведены на рисунках 2, 3 и 4.



```
S:\>lr4.exe
S:\>lr4.exe                               Interrupts: 44
Interruption has been already loaded.      Interrupts: 69
S:\>_
```

Рисунок 2 - повторный запуск

```
S:\>lr4.exe Interrupts: 44
Interruption has been already loaded. Interrupts: 276

S:\>lr4.exe /un
```

Рисунок 3 - запуск с хвостом "/un" при загруженном резидентным обработчике

```
S:\>lr4.exe /un

S:\>lr4.exe /un
Interruption is not loaded.

S:\>
```

Рисунок 4 - запуск с хвостом “/un” при не загруженном резидентном обработчике

Далее запустим программу LR3_1.com после запуска LR4.exe. Результат приведён на рисунке 5.

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
Number: 2
Owner: free
Area size(b): 64

Number: 3
Owner: 0040
Area size(b): 256

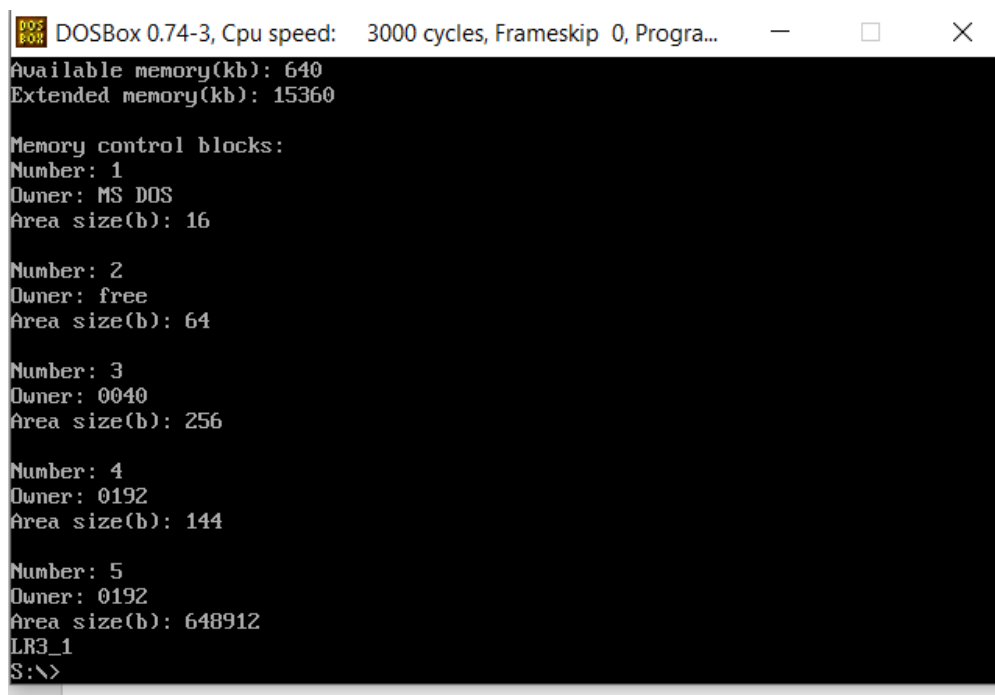
Number: 4
Owner: 0192
Area size(b): 144

Number: 5
Owner: 0192
Area size(b): 4560
LR4
Number: 6
Owner: 02BA
Area size(b): 144

Number: 7
Owner: 02BA
Area size(b): 644176 Interrupts: 123
LR3_1
S:\>_
```

Рисунок 5 - результат работы LR3_1.com после запуска LR4.exe

Далее был выгружен резидентный обработчик и снова запущена LR3_1.com. Результат работы представлен на рисунке 6.

A screenshot of a DOSBox window. The title bar reads "DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...". The window contains a black background with white text. The text displays memory statistics: "Available memory(kb): 640", "Extended memory(kb): 15360", and a list of "Memory control blocks" with five entries. Each entry shows a "Number", an "Owner", and an "Area size(b)". The last entry is labeled "LR3_1". The prompt "S:\>" is visible at the bottom.

```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
Available memory(kb): 640
Extended memory(kb): 15360

Memory control blocks:
Number: 1
Owner: MS DOS
Area size(b): 16

Number: 2
Owner: free
Area size(b): 64

Number: 3
Owner: 0040
Area size(b): 256

Number: 4
Owner: 0192
Area size(b): 144

Number: 5
Owner: 0192
Area size(b): 648912
LR3_1
S:\>
```

Рисунок 6 - запуск LR3_1 после выгрузки обработчика

Исходя из этого, видно, что память, которую занимал обработчик, освобождена, а также что сообщения обработчика не выводятся.

Ответы на контрольные вопросы.

1) Как реализован механизм прерывания от часов?

Прерывание системного таймера `int 8h` срабатывает примерно 18,2 раз в секунду (точное значение – $1193180/65536$). При инициализации BIOS устанавливает свой обработчик для прерывания таймера. Этот обработчик при срабатывании увеличивает на 1 текущее значение счётчика срабатываний таймера. Этот счётчик расположен по адресу `0000:046Ch`. Если это поле переполнено, то в ячейку `0000:0470h` заносится единица. В конце работы обработчик прерывания таймера вызывает прерывание `int 1Ch`, в которой может храниться пользовательское прерывание по таймеру. Если пользовательское прерывание не установлено в `1Ch`, то там хранится команда `IRET`. Во время работы обработчика `int 8h` и `int 1Ch` запрещены все аппаратные прерывания, а после завершения их работы осуществляется возврат к продолжению исполнению кода, который был прерван этим обработчиком.

2) Какого типа прерывания использовались в работе?

Были использованы: прерывания системы DOS (int 21h), аппаратное прерывание таймера.

Вывод.

В ходе выполнения данной лабораторной работы была написана программа, которая позволяет загружать и выгружать резидентный обработчик прерывания от системного таймера в память.