

**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. 7383

\_\_\_\_\_

Рудоман В.А.

Преподаватель

\_\_\_\_\_

Ефремов М.А.

Санкт-Петербург

2019

## **Цель лабораторной 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.

## **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 о 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 о 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.

Шаг 2. Далее необходимо запустить отлаженную программу и убедиться, что резидентный обработчик прерывания 1Ch установлен. Работа прерывания должна отображаться на экране, а также необходимо проверить размещение прерывания в памяти. Для этого нужно запустить программу ЛР 3, которая отображает карту памяти в виде с писка блоков MCB.

Шаг 3. Затем необходимо запустить отлаженную программу еще раз и убедиться, что программа определяет установленный обработчик прерываний.

Шаг 4. Далее нужно запустить отлаженную программу с ключом выгрузки и убедиться, 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.

Оформить отчёт и ответить на контрольные вопросы.

Процедуры, которые используются в программе.

ROUT	Функция обработчика прерывания.
IS_LOADED	Функция проверки загружен ли резидент в память.
IS_UNLOADED	Функция проверки ключа выгрузки резидента из памяти.
RES_LOAD	Функция загрузки резидента в память.
RES_UNLOAD	Функция выгрузки резидента из памяти.
PRINT	Функция печати на экран.

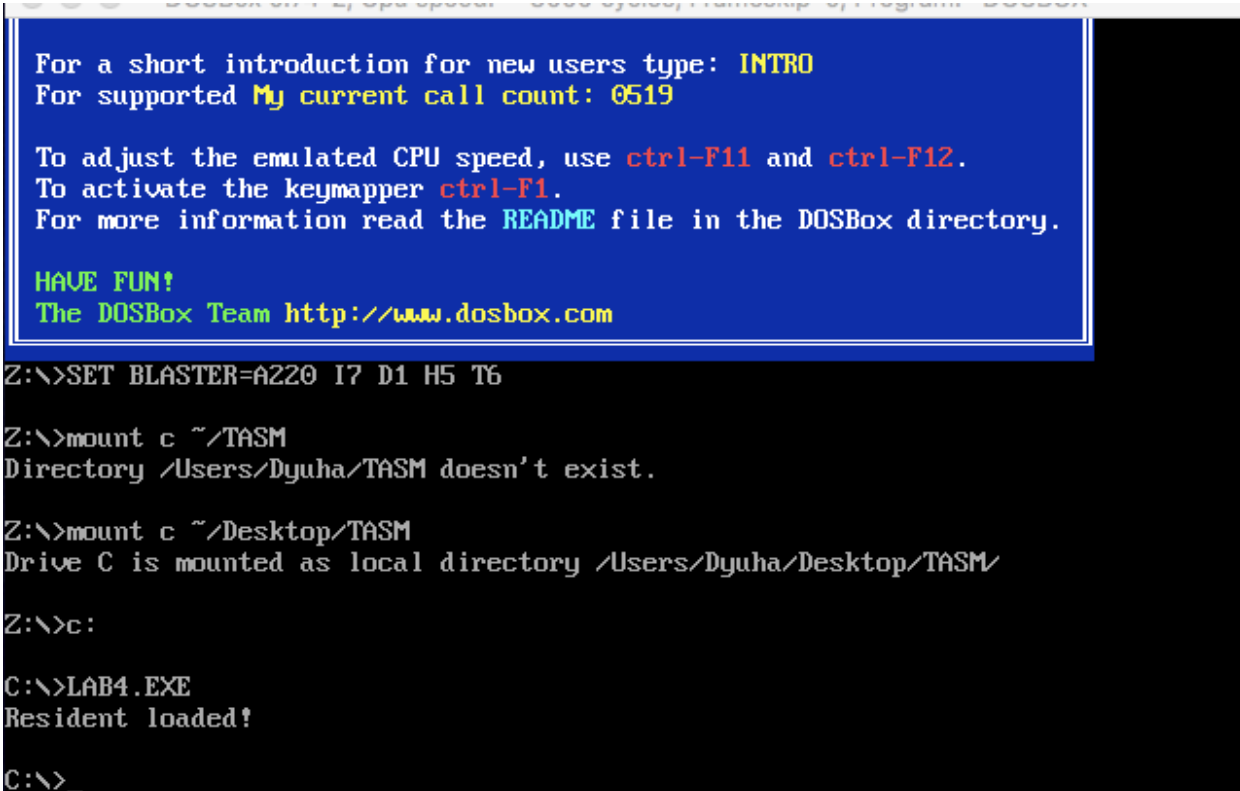
Переменные, которые используются в программе.

resNotSet	db	Применяется для вывода информации о том, что резидент не загружен в память.
resUnload	db	Применяется для вывода информации о том, что резидент выгружен из памяти.
resAlrSet	db	Применяется для вывода информации о том, что резидент уже загружен в память.
resLoad	db	Применяется для вывода информации о том, что резидент загружен в память.
COUNTER	db	Счетчик обработчика прерывания
INTER_ADR	db	Применяется для проверки того, загружен ли уже резидент в память или нет.
KEEP_IP	dw	Применяется для запоминания смещения вектора прерывания.
KEEP_CS	dw	Применяется для запоминания сегмента вектора прерывания.

## Ход выполнения работы

Результаты выполнения программы представлены на рисунках 1.1-1.6:

Рис.1.1



```
For a short introduction for new users type: INTRO  
For supported My current call count: 0519  
  
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 ~/TASM  
Directory /Users/Dyuha/TASM doesn't exist.  
  
Z:\>mount c ~/Desktop/TASM  
Drive C is mounted as local directory /Users/Dyuha/Desktop/TASM/  
  
Z:\>c:  
  
C:\>LAB4.EXE  
Resident loaded!  
  
C:\>_
```

```
DOSBox 0.74-2, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX

To adjust the emulated CPU speed, use ctrl-F11 and ctrl-F12.
To activate the My current call count: 0765
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 ~/TASM
Directory /Users/Dyuha/TASM doesn't exist.

Z:\>mount c ~/Desktop/TASM
Drive C is mounted as local directory /Users/Dyuha/Desktop/TASM/

Z:\>c:

C:\>LAB4.EXE
Resident loaded!

C:\>LAB4.EXE
Resident already loaded!

C:\>
```

Рис.1.2

```
DOSBox 0.74-2, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX

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 ~/TASM
Directory /Users/Dyuha/TASM doesn't exist.

Z:\>mount c ~/Desktop/TASM
Drive C is mounted as local directory /Users/Dyuha/Desktop/TASM/

Z:\>c:

C:\>LAB4.EXE
Resident loaded!

C:\>LAB4.EXE
Resident already loaded!

C:\>LAB4.EXE /un
Resident unloaded!

C:\>
```

Рис.1.3

```
DOSBox 0.74-2, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
The DOSBox Team http://www.dosbox.com

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

Z:\>mount c ~/TASM
Directory /Users/Dyuha/TASM doesn't exist.

Z:\>mount c ~/Desktop/TASM
Drive C is mounted as local directory /Users/Dyuha/Desktop/TASM/

Z:\>c:

C:\>LAB4.EXE
Resident loaded!

C:\>LAB4.EXE
Resident already loaded!

C:\>LAB4.EXE /un
Resident unloaded!

C:\>LAB4.EXE /un
Resident didnt loaded now!

C:\>
```

Рис.1.4

```
Resident already loaded!

C:\>LAB31.COM My current call count: 0452

Size of available memory: 648048 byte
Freeing mermory...
Suscess!
Size of extended memory: 245760 byte
MCB #01
Addr: 016F Owner: Area belongs to MS DOS Size: 16 byte Name:
MCB #02
Addr: 0171 Owner: Empty area Size: 64 byte Name:
MCB #03
Addr: 0176 Owner: 0040 Size: 256 byte Name:
MCB #04
Addr: 0187 Owner: 0192 Size: 144 byte Name:
MCB #05
Addr: 0191 Owner: 0192 Size: 688 byte Name: LAB4
MCB #06
Addr: 01BD Owner: 01CB Size: 144 byte Name:
MCB #07
Addr: 01C7 Owner: 01CB Size: 1840 byte Name: LAB31
MCB #08
Addr: 023B Owner: Empty area Size: 646192 byte Name:
Press any key...
```

Рис.1.5

```

C:\>LAB4.EXE /un
Resident unloaded! current call count: 0905

C:\>LAB31.COM

Size of available memory: 648912 byte
Freeing mermory...
Suscess!
Size of extended memory: 245760 byte
MCB #01
Addr: 016F      Owner: Area belongs to MS DOS Size: 16 byte Name:
MCB #02
Addr: 0171      Owner: Empty area Size: 64 byte Name:
MCB #03
Addr: 0176      Owner: 0040      Size: 256 byte Name:
MCB #04
Addr: 0187      Owner: 0192      Size: 144 byte Name:
MCB #05
Addr: 0191      Owner: 0192      Size: 1840 byte Name: LAB31
MCB #06
Addr: 0205      Owner: Empty area Size: 647056 byte Name: ||)@áá δ
Press any key...

```

Рис.1.6

### Контрольные вопросы

1. *Как реализован механизм прерывания от часов?*

При каждом такте таймера (часов) происходит(каждые 55 миллисекунд или приблизительно 18.2 раз в секунду): сначала сохраняется состояние регистров, затем определяется источник прерывания (по номеру источника прерывания определяется смещение в таблице векторов прерываний), далее эти данные сохраняются, т. е. - первые два байта помещаются в IP, второе два байта в CS, далее запускается обработчик прерывания по адресу CS:IP и обрабатывается прерывание, в конце происходит возврат управления прерванной программе

2. *Какого типа прерывания использовались в работе?*

Были использованы пользовательские прерывания, такие как int 10h(стандартного видеосервиса ROM-BIOS) и int 21h(сервис DOS) и аппаратные прерывания (1Ch(пользовательское прерывание по таймеру)).

### Вывод

В результате выполнения данной лабораторной работы были исследованы организация и управление прерываниями. Была написана программа, в которой построен обработчик прерываний сигналов таймера.