**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. 6383 |  | Базаров И.А |
| Преподаватель |  | Губкин А. Ф. |

Санкт-Петербург

2018

**Постановка задачи**

1. 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 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 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.

1. Описание функций:

|  |  |
| --- | --- |
| Название | Назначение |
| INTERRUPTION | Функция обработчика прерывания |
| IS\_INTERRUPTION\_SET | Проверка установки резидента |
| CHECK\_COMMAND\_PROMT | Проверка команды '/un' |
| LOAD\_INTERRUPTION | Загрузка резидента |
| UNLOAD\_INTERRUPTION | Выгрузка резидента |

1. Описание данных:

|  |  |  |
| --- | --- | --- |
| Название | Тип | Назначение |
| M\_INT\_NOT\_SET | Db | Резидент не загружен |
| M\_INT\_RESTORED | Db | Резидент загружен |
| M\_INT\_ISLOADED | Db | Резидент уже загружен |
| M\_INT\_ISLOADED0 | Db | Резидент был выгружен |

1. Общие сведения:

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

Обработчик прерывания - это отдельная процедура, имеющая следующую структуру:

ROUT PROC FAR

PUSH AX ; сохранение изменяемых регистров

.....................................

<действия по обработке прерывания>

POP AX ; восстановление регистров

MOV AL, 20H

OUT 20H,AL

IRET

ROUT ENDP

Две последние строки необходимы для разрешения обработки прерываний с более низкими уровнями, чем только что обработанное. Для установки написанного прерывания в поле векторов прерываний используется функция 25H прерывания 21H, которая устанавливает вектор прерывания на указанный адрес.

PUSH DS

MOV DX, OFFSET ROUT ; смещение для процедуры в DX

MOV AX, SEG ROUT ; сегмент процедуры

MOV DS, AX ; помещаем в DS

MOV AH, 25H ; функция установки вектора

MOV AL, 1CH ; номер вектора

INT 21H ; меняем прерывание

POP DS

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

; -- хранится в обработчике прерываний

KEEP\_CS DW 0 ; для хранения сегмента

KEEP\_IP DW 0 ; и смещения прерывания

; -- в программе при загрузке обработчика прерывания

MOV AH, 35H ; функция получения вектора

MOV AL, 1CH ; номер вектора ШТЕ 21Р

MOV KEEP\_IP, BX ; запоминание смещения

MOV KEEP\_CS, ES ; и сегмента

; -- в программе при выгрузке обработчика прерываний CLI

PUSH DS

MOV DX, KEEP\_IP

MOV AX, KEEP\_CS

MOV DS, AX

MOV AH, 25H

MOV AL, 1CH

INT 21H ; восстанавливаем вектор

POP DS

STI

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

Функция 31h int 21h использует следующие параметры:

AH - номер функции 31h;

AL - код завершения программы;

DX - размер памяти в параграфах, требуемый резидентной программе.

Пример обращения к функции:

MOV DX, OFFSET LAST\_BYTE ; размер в байтах от начала сегмента

MOV CL,4 ; перевод в параграфы

SHR DX,CL

INC DX ; размер в параграфах

MOV AH,31h

INT 21h

**Ход работы:**

1. Состояние памяти до загрузки резидента (используем модуль, разработанный в третьей лабораторной работе).

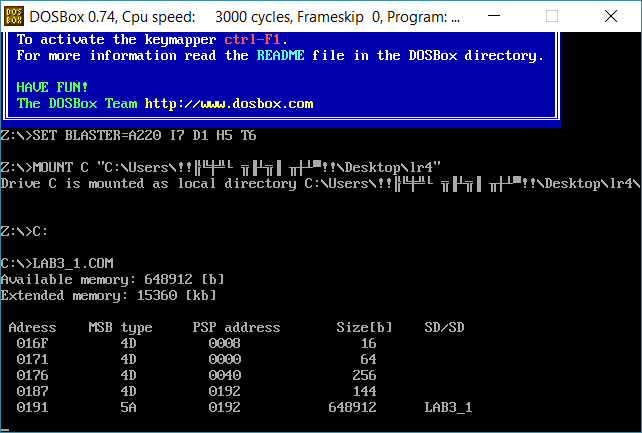


Рисунок 1

1. Загрузка резидента в память.

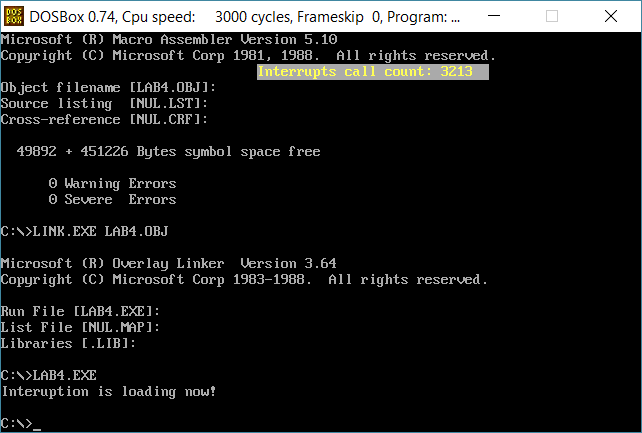


Рисунок 2

1. Попытка повторной загрузки резидента в память.

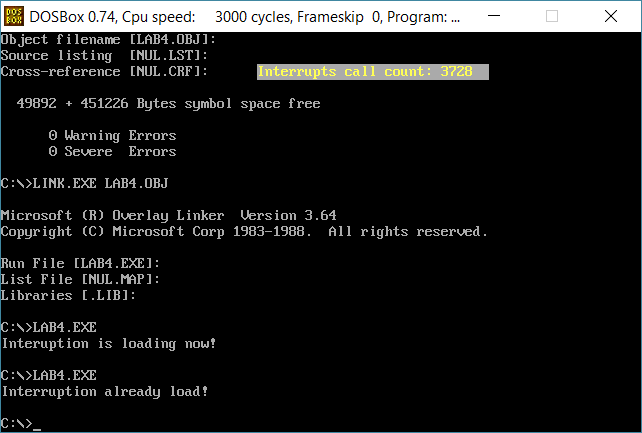


Рисунок 3

1. Состояние памяти при загрузке в неё резидента.

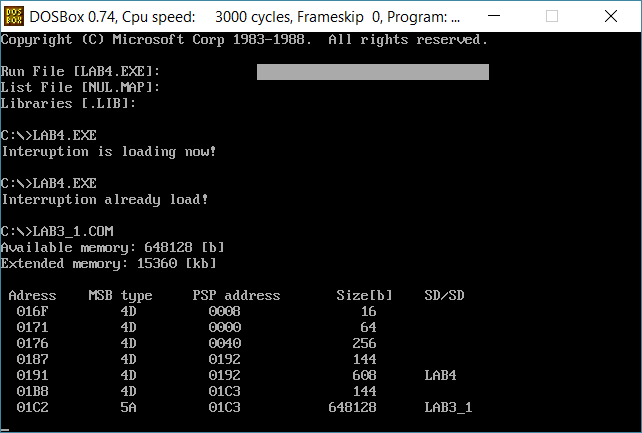


Рисунок 4

1. Запускаем отложенную программу с ключом /un, тем самым выгружаем резидент и смотрим состояние памяти после выгрузки резидента.

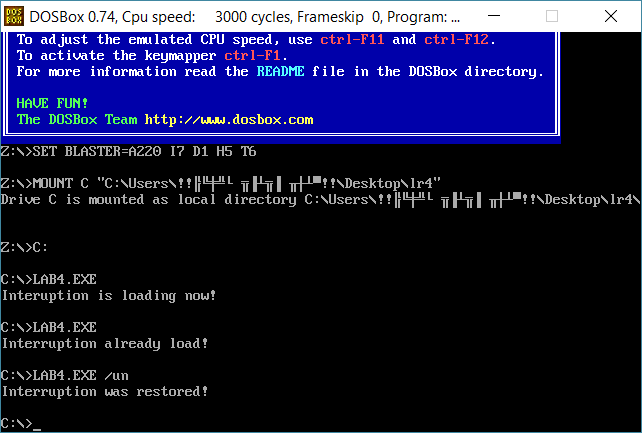


Рисунок 5

**Ответы на контрольные вопросы:**

1. *Как реализован механизм прерывания от часов?*

Сначала сохраняется содержимое регистров, потомопределяется источник прерывания, по номеру которого определяется смещение в таблице векторов прерывания, сохраняется в CS : IP, передаётся управление по адресу CS:IP и происходит выполнение обработчика, и в конце происходит возврат управления прерванной программе. Аппаратное прерывание от таймера происходит каждые 55 мс.

1. *Какого типа прерывания использовались в работе?*

Аппаратные прерывания (1Сh); прерывания функций DOS (21h); прерывания функций BIOS (10h).

**Заключение:**

В ходе лабораторной работы был построен обработчик прерывания от сигналов таймера. Изучены дополнительные функции работы с памятью: установка программы-резидента и его выгрузка из памяти.