

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра математического обеспечения и применения ЭВМ**

**ОТЧЕТ**  
**по практической работе № 4**  
**по дисциплине «Операционные системы»**  
**Тема: Обработка стандартных прерываний**

Студент гр.

Диденко Д. В.

Преподаватель

Ефремов М. А.

Санкт-Петербург

2022

### **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 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.

### **Постановка задачи.**

Требуется написать и отладить программный модуль типа .EXE, который выполняет следующие функции:

- Проверяет, установлено ли пользовательское прерывание с вектором 1Ch.
- Устанавливает резидентную функцию для обработки прерывания и настраивает вектор прерываний, если прерывание не установлено, и осуществляет выход по функции 4Ch прерывания int 21h.
- Если прерывание установлено, то выводится соответствующее сообщение и осуществляется выход по функции 4Ch прерывания int 21h.
- 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.

Код пользовательского прерывания должен выполнять следующие функции:

- Сохранять значения регистров в стеке при входе и восстановить их при выходе.
- 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.

### **Выполнение работы.**

Были написаны строки для вывода информации:

- STR\_IST\_LOAD DB 'Iterrupt is not load', 0AH, 0DH,'\$'
- STR\_ALR\_LOAD DB 'Iterrupt is already loaded', 0AH, 0DH,'\$'
- STR\_SUC\_LOAD DB 'Iterrupt has been loaded', 0AH, 0DH,'\$'
- STR\_IS\_UNLOAD DB 'Iterrupt is unloaded', 0AH, 0DH,'\$'

Переменные для хранения флагов:

- flag db 0 – флаг удаления;
- flag\_load db 0 – флаг загрузки.

Переменные, хранящиеся в прерывании:

PSP dw ? – сохранение адреса PSP;

KEEP\_IP dw 0 – сохранение данных исходного прерывания;

KEEP\_CS dw 0 – сохранение данных исходного прерывания;

ITERRUPT\_ID dw 8f17h – уникальный идентификатор прерывания;

STR\_COUNTER db 'Number of my iterrups: 0000\$' – строка вывода кол-во прерываний;

KEEP\_SS dw ? – для работы стека прерывания;

KEEP\_SP dw ? – для работы стека прерывания;

KEEP\_AX dw ? – для работы стека прерывания;

ITERRUPT\_STACK dw 32 dup (?) – стек прерывания;

END\_IT\_STACK dw ? – конец стека прерывания;

Были составлены функции (см. табл.1)

Таблица 1 – функции в программе

Процедура	Описание
MY_ITERRUPT	Резидентное прерывание, которое загружается в память и выполняет накопление и вывод числа накопленных прерываний на экран.
WRITE_STRING	Вывод строки на экран
LOAD_FLAG	Проверка на наличия флага “/un”
IS_LOAD	Проверка на загрузку пользовательского прерывания в память
LOAD_ITERRAPT	Сохранение первоначального прерывания и загрузка пользовательского прерывания в память
UNLOAD_ITERRAPT	Выгрузка пользовательского прерывания из памяти, а также освобождение памяти и восстановление первоначальных прерываний
MAIN	Главная функция

В результате выполнения были получены следующие значения(рис.1-4):

```

Z:\>SNumber of my iterrups: 0419

Z:\>MOUNT C "C:\Users\Samsung\Desktop\Studing\OS\MASM"
Drive C is mounted as local directory C:\Users\Samsung\Desktop\Studing\OS\MASM\

Z:\>C:

C:\>lab4.exe
Iterrupt has been loaded

C:\>lab3_2

Size of accessed memory: 648224 byte
Size of extended memory: 245760 byte
MCB:01 Address: 016F PSP address: 0008 Size: 16 SD/SC:
MCB:02 Address: 0171 PSP address: 0000 Size: 64 SD/SC:
MCB:03 Address: 0176 PSP address: 0040 Size: 256 SD/SC:
MCB:04 Address: 0187 PSP address: 0192 Size: 144 SD/SC:
MCB:05 Address: 0191 PSP address: 0192 Size: 512 SD/SC: LAB4
MCB:06 Address: 01B2 PSP address: 01BD Size: 144 SD/SC:
MCB:07 Address: 01BC PSP address: 01BD Size: 832 SD/SC: LAB3_2
MCB:08 Address: 01F1 PSP address: 0000 Size: 647376 SD/SC:
C:\>

```

Рисунок 1 – прерывание загружено в память

```

MCB:0 Number of my iterrups: 1366s: 0000 Size: 64 SD/SC:
MCB:03 Address: 0176 PSP address: 0040 Size: 256 SD/SC:
MCB:04 Address: 0187 PSP address: 0192 Size: 144 SD/SC:
MCB:05 Address: 0191 PSP address: 0192 Size: 512 SD/SC: LAB4
MCB:06 Address: 01B2 PSP address: 01BD Size: 144 SD/SC:
MCB:07 Address: 01BC PSP address: 01BD Size: 832 SD/SC: LAB3_2
MCB:08 Address: 01F1 PSP address: 0000 Size: 647376 SD/SC:
C:\>lab4.exe
Iterrapt is already loaded

C:\>lab3_2

Size of accessed memory: 648224 byte
Size of extended memory: 245760 byte
MCB:01 Address: 016F PSP address: 0008 Size: 16 SD/SC:
MCB:02 Address: 0171 PSP address: 0000 Size: 64 SD/SC:
MCB:03 Address: 0176 PSP address: 0040 Size: 256 SD/SC:
MCB:04 Address: 0187 PSP address: 0192 Size: 144 SD/SC:
MCB:05 Address: 0191 PSP address: 0192 Size: 512 SD/SC: LAB4
MCB:06 Address: 01B2 PSP address: 01BD Size: 144 SD/SC:
MCB:07 Address: 01BC PSP address: 01BD Size: 832 SD/SC: LAB3_2
MCB:08 Address: 01F1 PSP address: 0000 Size: 647376 SD/SC:
C:\>

```

Рисунок 2 – повторная загрузка прерывания

```

C:\>lab4.exe /un
Iterrapt is unloaded

C:\>lab3_2

Size of accessed memory: 648912 byte
Size of extended memory: 245760 byte
MCB:01 Address: 016F PSP address: 0008 Size: 16 SD/SC:
MCB:02 Address: 0171 PSP address: 0000 Size: 64 SD/SC:
MCB:03 Address: 0176 PSP address: 0040 Size: 256 SD/SC:
MCB:04 Address: 0187 PSP address: 0192 Size: 144 SD/SC:
MCB:05 Address: 0191 PSP address: 0192 Size: 832 SD/SC: LAB3_2
MCB:06 Address: 01C6 PSP address: 0000 Size: 648064 SD/SC:
C:\>

```

Рисунок 3 – выгрузка прерывания

```

C:\>lab4.exe /un
Iterrapt is not load

C:\>lab3_2

Size of accessed memory: 648912 byte
Size of extended memory: 245760 byte
MCB:01 Address: 016F PSP address: 0008 Size: 16 SD/SC:
MCB:02 Address: 0171 PSP address: 0000 Size: 64 SD/SC:
MCB:03 Address: 0176 PSP address: 0040 Size: 256 SD/SC:
MCB:04 Address: 0187 PSP address: 0192 Size: 144 SD/SC:
MCB:05 Address: 0191 PSP address: 0192 Size: 832 SD/SC: LAB3_2
MCB:06 Address: 01C6 PSP address: 0000 Size: 648064 SD/SC:
C:\>

```

Рисунок 4 – повторная выгрузка прерывания

### **Выводы.**

В ходе лабораторной работы была исследована обработка стандартных прерываний, а также построен обработчик прерываний сигналов таймера, которые генерируются аппаратурой через определённые интервалы времени. Программа загружает и выгружает резидент, а также производится проверка флагов и загрузки прерывание в память. С помощью rout выполняет накопление и вывод числа накопленных прерываний на экран.

## **ПРИЛОЖЕНИЕ А**

### **ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ**

1. Как реализован механизм прерывания от часов?

Принимается сигнал прерывания (приходит примерно каждые 54 мс), 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. Какие прерывания использовались в работе?

- Int 10h – видео сервис BIOS
- Int 21h – сервисы DOS
- Пользовательское прерывание с вектором 1ch int 21h