

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. 0381

Ефимов Н.Д.

Преподаватель

Губкин А.Ф.

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 No 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 /up. 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.

Выполнение работы.

Для отображения сообщений были инициализированы некоторые строки:

```
STR_IS_LOAD DB 'Interrupt is not load', 0AH, 0DH, '$'
```

```
STR_ALR_LOAD DB 'Interrupt is already loaded', 0AH, 0DH, '$'
```

```
STR_SUC_LOAD DB 'Interrupt has been loaded', 0AH, 0DH, '$'
```

```
STR_IS_UNLOAD DB 'Interrupt is unloaded', 0AH, 0DH, '$'
```

Так же потребовались некоторые переменные для хранения флагов:

flag db 0 – флаг удаления;

flag_load db 0 – флаг загрузки.

Ну и переменные для прерывания:

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 'INTERRUPTS: 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;
INTERRUPT_STACK dw 32 dup (?) – стек прерывания;
END_IT_STACK dw ? – конец стека прерывания;

Для выполнения лабораторной работы были написаны следующие процедуры:

-MY_INTERRUPT – 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_INTERRUPT – Сохранение первоначального прерывания и загрузка пользовательского прерывания в память

-UNLOAD_INTERRUPT – Выгрузка пользовательского прерывания из памяти, а также освобождение памяти и восстановление первоначальных прерываний

-MAIN - Главная функция

Результаты выполнения шагов изложены на скриншотах ниже:

1 тест – прерывание было загружено:

```
DOSBox 0.74-3-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988. All rights reserved.
    INTERRUPTS: 00340
Object filename [lab4.OBJ]:
Source listing [NUL.LST]:
Cross-reference [NUL.CRF]:

    49892 + 453273 Bytes symbol space free

    0 Warning Errors
    0 Severe Errors

F:\>link lab4

Microsoft (R) Overlay Linker Version 3.64
Copyright (C) Microsoft Corp 1983-1988. All rights reserved.

Run File [LAB4.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:

F:\>lab4
Interrupt has been loaded

F:\>
```

2 тест – попытка загрузить новое прерывание:

```
F:\>lab4
Interrupt is already loaded

F:\>
```

3 тест – выгрузить прерывание:

```
Interrupt is already loaded

F:\>lab4 /un
Interrupt is unloaded
```

При этом тесте значения таймера застыли.

4 тест – повторная выгрузка прерывания:

```
F:\>lab4 /un
Interrupt is not load
```

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 `cout` 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) Какого типа прерывания использовались в работе?

Первое прерывание – int 10h – видео сервис BIOS, второе – int 21h – сервисы DOS, ну и пользовательское прерывание с вектором 1ch int21h