

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. 7381

Ильясов А.В.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2019

Цель работы

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.

Необходимые сведения для составления программы

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 21P
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

Порядок выполнения работы

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 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 списка блоков МСВ.

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.

Ход работы

1) Сначала была запущена программа lab3_1.com из предыдущей лабораторной работы, чтобы посмотреть на начальное состояние памяти.

```
C:\>LAB3_1.COM
Number of available memory: 648912 bytes
Extended memory size: 15360 Kbytes
Memory control circuitry:
ADDRESS | OWNER | SIZE | NAME
016F    | 0008   | 16   |
0171    | 0000   | 64   |
0176    | 0040   | 256  |
0187    | 0192   | 144  |
0191    | 0192   | 6432 | LAB3_1
0324    | 0000   | 642464
```

Рисунок 1 – состояние памяти до загрузки прерывания

2) Далее было загружено прерывание и просмотрено изменение состояния памяти.

```
C:\>LAB4.EXE
Interruption is loaded
Interrupts call count: 0334

C:\>LAB3_1.COM
Number of available memory: 647952 bytes
Extended memory size: 15360 Kbytes
Memory control circuitry:
ADDRESS : OWNER : SIZE : NAME
016F      0008      16
0171      0000      64
0176      0040     256
0187      0192     144
0191      0192     784    LAB4
01C3      01CE     144
01CD      01CE    7392    LAB3_1
039C      0000   640544
```

Рисунок 2 – отображение счетчика обработчика прерывания и состояние памяти

3) В конце прерывание было выгружено и снова была проверена память.

```
C:\>LAB4.EXE/un
Interruption is restored

C:\>LAB3_1.COM
Number of available memory: 648912 bytes
Extended memory size: 15360 Kbytes
Memory control circuitry:
ADDRESS : OWNER : SIZE : NAME
016F      0008      16
0171      0000      64
0176      0040     256
0187      0192     144
0191      0192    6432    LAB3_1
0324      0000   642464
```

Рисунок 3 – состояние памяти после выгрузки прерывания

Выводы

В процессе выполнения данной лабораторной работы был построен обработчик прерываний сигналов таймера.

Ответы на контрольные вопросы.

1) Как реализован механизм прерывания от часов?

Ответ: При каждом «тике» таймера происходит следующее:

1. сохраняется состояние регистров
2. определяется источник прерывания (по номеру источника прерывания определяется смещение в таблице векторов прерываний)
3. первые два байта помещаются в IP, второе два байта в CS передаётся управление по адресу CS:IP (т.е. «запускается» обработчик прерывания) обработка прерывания возврат управления прерванной программе

2) Какого типа прерывания использовались в работе?

Ответ: в работе использовались программные (int 21h, int 10h) и аппаратные прерывания (int 1Ch).