

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»

Студентка гр. 6381

Дайнович А.Ю.

Преподаватель

Губкин А. Ф.

Санкт-Петербург

2018

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 и выполняет соответствующие действия.

В лабораторной работе номер 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 прерывание - это отдельная процедура, имеющая следующую структуру:

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	:		
MOV	:	OFFSET	;смещение для процедуры в DX
MOV	:	SEG ROUT	;сегмент процедуры
MOV	:	AX	;помещаем в DS
MOV	:	25H	;функция установки вектора
MOV	:	1CH	;номер вектора
INT	:		;меняем прерывание

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
INT 21H
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

Вывод на экран информации обработчиком прерываний осуществляется с помощью функций прерывания 10h.

Алгоритм работы программы.

1. Проверяется, установлено ли пользовательское прерывание с вектором 1Ch. Проверка происходит путём сравнения значения сигнатуры, расположенной на определённом известном смещении в теле резидента, с реальным кодом, находящимся в резиденте.
2. Если пользовательское прерывание не установлено (код не совпадает с сигнатурой), то устанавливается резидентная функция для обработки прерывания, настраивается вектор прерываний и осуществляется выход по функции 4Ch прерывания int 21h.
3. Если пользовательское прерывание установлено (код совпадает с сигнатурой), то выводится соответствующее сообщение и осуществляется выход по функции 4Ch прерывания int 21h.
4. Если пользовательское прерывание установлено и программа запущена с ключом /up, то прерывание выгружается из памяти: происходит восстановление стандартного вектора прерываний и освобождение памяти, занимаемой резидентом. Затем осуществляется выход по функции 4Ch прерывания int 21h.

Сведения о функциях и структурах данных управляющей программы.

Название процедуры	Входные данные	Выходные данные	Описание
PRINT	DX	-	Выводит на экран строку
setCurs	BH, DX	-	Устанавливает курсор по данным в регистре DX: в DH - номер строки, в DL – номер колонки
getCurs	BH	DX	Определяет позицию и размер курсора. После выполнения: в DH – текущая строка курсора, в DL – текущая колонка курсора
ROUT	SIGNATUR, DELETE, KEEP_CS, KEEP_IP, KEEP_PSP, COUNTER	COUNTER	Пользовательский обработчик прерывания: накапливает общее суммарное число прерываний и выводит на экран это значение; при значении переменной DELETE равной 1 - восстанавливает стандартное прерывание и выгружается из памяти
CHECK_INT	-	DELETE	Проверяет, установлено ли пользовательское прерывание, а так же помещает в переменную DELETE функции прерывания значение 1, если в командной строке был параметр /up и сигнатуры функций прерываний совпадали
SET_INT	-	KEEP_IP, KEEP_CS	Устанавливает пользовательское прерывание в поле векторов прерываний

Таблица 1. Описание функций.

Название переменной	Тип	Описание
ALR_LOADED	db	Строка-сообщение для вывода информации о том, что пользовательское прерывание уже установлено
UNLOADED	db	Строка-сообщение для вывода информации о том, что пользовательское прерывание выгружено
IS_LOADED	db	Строка-сообщение для вывода информации о том, что пользовательское прерывание установлено
SIGNATURE	db	Строка-сигнатура, которая используется для определения: было ли установлено пользовательское прерывание или нет
KEEP_CS	dw	Переменная для сохранения значения в регистре CS
KEEP_IP	dw	Переменная для сохранения значения в регистре IP
KEEP_PSP	dw	Переменная для сохранения сегментного адреса PSP
DELETION	db	Переменная, позволяющая определить: нужно ли выгружать пользовательское прерывание (при значении 1) или нет (при значении 0)
COUNTER	db	Строка-сообщение для вывода информации об общем суммарном числе прерываний

Таблица 2. Описание переменных.

Ход работы.

1. Оценим состояние памяти до запуска программы lab4.exe с помощью ранее разработанной программы lab3_1.com:

```

C:\>lab3_1
Amount of available memory: 648912 B
Size of extended memory: 15360 Kb
Chain of memory control blocks:
ADDRESS  OWNER  SIZE  NAME
0008h - Space belongs MS DOS
016F      0008    16
0000h - FreeSpace
0171      0000    64
0176      0040   256
0187      0192   144
0191      0192 648912  LAB3_1

```

Рис. 1. Результат работы программы lab3_1.com до загрузки резидентного обработчика прерывания 1Ch в память.

2. Запуск программы lab4.exe:

```
C:\>lab4
User interruption is loaded! Total number of interrupts: 0722
```

Рис. 2. Результат запуска программы lab4.exe.

Как видно на Рис. 2, резидентный обработчик прерывания 1Ch установлен, и счетчик работает корректно, то есть прерывания от таймера срабатывают.

3. После загрузки резидента в память, проверим его размещение в памяти с помощью программы lab3_1.com, которая отображает карту памяти в виде списка блоков MCB:

```
C:\>lab3_1
Amount of available memory: 647776 B
Size of extended memory: 15360 Kb
Chain of memory control blocks:
ADDRESS  OWNER  SIZE  NAME
0008h - Space belongs MS DOS
016F      0008    16
0000h - FreeSpace
0171      0000    64
0176      0040   256
0187      0192   144
0191      0192   960  LAB4
01CE      01D9   144
01D8      01D9 647776  LAB3_1
Total number of interrupts: 3417
```

Рис. 3. Результат работы программы lab3_1.com после загрузки резидентного обработчика прерывания 1Ch в память.

Исходя из результата убеждаемся, что обработчик прерывания остается в памяти резидентно.

4. Запустим программу lab4.exe ещё раз:

```
C:\>lab4
User interruption is already loaded! Total number of interrupts: 6345
```

Рис. 4. Результат повторного запуска программы lab4.exe.

Можно убедиться, в том, что программа определяет, что обработчик уже находится в памяти резидентно.

5. Запустим программу lab4.exe с ключом выгрузки /un:

```
C:\>lab4/un  
User interruption is unloaded!
```

Рис. 5. Результат работы программы lab4.exe, запущенной с ключом /un.

Наблюдаем, что резидентный обработчик прерывания был выгружен так, как об этом нас информирует сообщение, выведенное на экран программой, а также то, что счетчик больше не присутствует на экране.

6. Убедимся в том, что занятая резидентным обработчиком прерывания 1Ch память освобождена, запустив программу lab3_1.com:

```
C:\>lab3_1  
Amount of available memory: 648912 B  
Size of extended memory: 15360 Kb  
Chain of memory control blocks:  
ADDRESS  OWNER  SIZE  NAME  
0008h - Space belongs MS DOS  
016F      0008    16  
0000h - FreeSpace  
0171      0000    64  
0176      0040   256  
0187      0192   144  
0191      0192 648912 LAB3_1
```

Рис. 6. Результат работы программы lab3_1.com выгрузки пользовательского прерывания.

Исходя из результата убеждаемся, что обработчик прерывания был выгружен.

Ответы на контрольные вопросы.

1. Как реализован механизм прерывания от часов?

Ответ: Прерывание 1Ch вызывается автоматически при каждом тике системного таймер (каждые 55мс). После вызова, сохраняется содержимое регистров, затем определяется источник прерывания, по номеру которого определяется смещение в таблице векторов прерываний. Полученный адрес

сохраняется в регистры CS:IP, после чего управление передается по этому адресу, т.е. выполняется запуск обработчика прерываний и происходит его выполнение. После выполнения, происходит возврат управления прерванной программе.

2. Какого типа прерывания использовались в работе?

Ответ: В данной лабораторной работе использовались: аппаратные прерывания (int 1Ch), программные прерывания функций DOS (int 21h) и программные прерывания функций BIOS (int 10h).

Заключение.

В результате выполнения лабораторной работы был создан обработчик прерываний сигналов таймера, изучены дополнительные функции работы с памятью, такие как установка программы-резидента и его выгрузка из памяти.