

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

ОТЧЕТ
по лабораторной работе № 4
по дисциплине «Операционные системы»
Тема: «Обработка стандартных исключений»

Студентка гр. 8381

Гречко В.Д.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2020

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 о 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 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.

Необходимые сведения для составления программы.

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

Для установки написанного прерывания используется функция 25H прерывания 21H, которая устанавливает вектор прерывания на указанный адрес.

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.

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

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

Постановка задачи:

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 /un. Выгрузка прерывания состоит в восстановлении стандартного вектора прерываний и освобождения 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 определённом, 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. Длина кода сигнатуры должна быть достаточной, чтобы сделать случайное совпадение маловероятным.

Программа должна содержать код устанавливаемого прерывания в виде удалённой процедуры. Этот код будет работать после установки при возникновении прерывания. Он должен выполнять следующие функции:

1. Сохранить значение регистров в стеке при входе и восстановить их при выходе.
2. 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.

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 отчёт.

Шаг 3. Запустите отлаженную программу ещё раз и убедитесь, что программа определяет установленный обработчик прерываний. Полученные результаты поместите в отчёт.

Шаг 4. Запустите отлаженную программу с ключом загрузки 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 отчёт.

Ход работы

Шаг 1. Написан и отлажен программный модуль типа **.EXE**, который выбирает и распечатывает следующую информацию:

- 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 о 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.

Результат работы программы представлен на рис. 1.



```
F:\>lab4.exe
Interruption was loaded
667 interruptions
```

Рисунок 1 — Вывод программы lab4.exe после первого запуска

Шаг 2. Запущена отлаженная программа, резидентный обработчик прерывания 1Ch установлен. Работа прерывания отображается на экране, проверено размещение прерывания в памяти. Для этого запущена программа ЛР 3, которая отображает карту памяти в виде списка блоков МСВ. Результат представлен на рис. 2.

```

F:\>lab3_2.com
Available memory: 640 kbytes
Extended memory: 15360 kbytes
ICBs:
ICB number 1
Block is occupied by MS DOS, size = 16 bytes; occupied by: no info
ICB number 2
Block is free, size = 64 bytes; occupied by: no info
ICB number 3
Block is owned by PSP = 0040, size = 256 bytes; occupied by: no info
ICB number 4
Block is owned by PSP = 0193, size = 160 bytes; occupied by: no info
ICB number 5
Block is owned by PSP = 0193, size = 720 bytes; occupied by: LAB4
ICB number 6
Block is owned by PSP = 01CC, size = 160 bytes; occupied by: no info
ICB number 7
Block is owned by PSP = 01CC, size = 1072 bytes; occupied by: LAB3_2
ICB number 8
1579 interruptions      46896 bytes; occupied by: ^♦í\>ï_^

```

Рисунок 2 — Вывод программы lab3_2.com после выполнения lab4.exe

Шаг 3. Повторно запущен программа. Как видно из рис. 3, процедура прерывания осталась резидентной в памяти. Также на рис. 3 показано, что при повторном запуске программа выводит сообщение о том, что резидентный обработчик уже загружен.

```

F:\>lab4.exe
Interruption is loaded
2281 interruptions

```

Рисунок 3 — Вывод программы lab4.exe при повторном запуске

Шаг 4. Программа была запущена с ключом выгрузки. Из рис. 4 видно, что после выгрузки резидентного обработчика из памяти вся занятая им память была освобождена. Как видно из рис. 5, при выгрузке резидентного обработчика было выведено сообщение, а также при запросе повторной выгрузки было показано, что резидентный обработчик не загружен.

```

F:\>lab4.exe /un
Interruption was unloaded
F:\>lab3_2.com
Available memory: 640 kbytes
Extended memory: 15360 kbytes
MCBs:
MCB number 1
Block is occupied by MS DOS, size = 16 bytes; occupied by: no info
MCB number 2
Block is free, size = 64 bytes; occupied by: no info
MCB number 3
Block is owned by PSP = 0040, size = 256 bytes; occupied by: no info
MCB number 4
Block is owned by PSP = 0193, size = 160 bytes; occupied by: no info
MCB number 5
Block is owned by PSP = 0193, size = 1072 bytes; occupied by: LAB3_2
MCB number 6
Block is free, size = 647808 bytes; occupied by: no info

```

Рисунок 4 - Вывод программы lab3_2.com после выполнения lab4.exe с ключом выгрузки

```

F:\>lab4.exe
Interruption was loaded
86 interruptions
F:\>lab4.exe /un
Interruption was unloaded
F:\>lab4.exe /un
Interruption is not loaded

```

Рисунок 5 — Вывод программы lab4.exe при повторном запуске с ключом выгрузки

Шаг 5. Даны ответы на контрольные вопросы:

1) Как реализован механизм прерывания от часов?

Ответ: Любой компьютер (даже простейший IBM PC) содержит устройство, называемое системным таймером. Это устройство подключено к линии запроса на прерывание IRQ0 и вырабатывает прерывание INT 8h приблизительно 18,2 раза в секунду (точное значение - $1193180/65536$ раз в секунду). Стандартный обработчик этого прерывания увеличивает счётчик и вызывает другое прерывание — 1CH. По умолчанию оно указывает на команду IRET. Во время выполнения этих двух прерываний не вызываются другие.

2) Какого типа прерывания использовались в работе?

Ответ: INT 21h – сервис DOS. INT 10h – программное прерывание BIOS. Написанный обработчик применялся к асинхронному аппаратному прерыванию, 1 CH, прерыванию от таймера.

Вывод.

В результате выполнения данной лабораторной работы была изучена работа прерываний от системного таймера, а также механизм загрузки и выгрузки резидентных обработчиков.