**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**

|  |  |  |
| --- | --- | --- |
| Студентка гр. 7381 |  | Кушкоева А.О. |
| Преподаватель |  | Ефремов М.А. |

Санкт-Петербург

2019

**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.

**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 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.

**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 с писка блоков MCB.

**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.

**Шаг 5.** Оформить отчёт и ответить на контрольные вопросы.

**Необходимые сведения для выполнения лабораторной работы:**

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 | 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 |

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

**Ход работы**

Результат работы показан на рисунке ниже.

1)Первый запуск lab4.exe, прерывание не установлено, попытаемся восстановить прерывание, используя при этом ключ /un. В результате должна вылезти ошибка, так как нечего ещё восстанавливать.

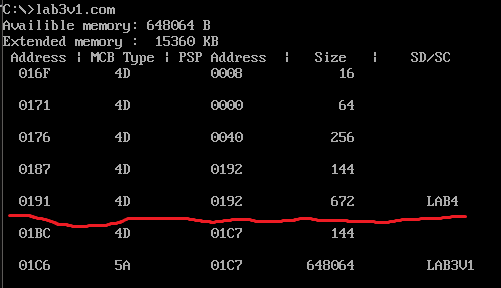


2)Запуск lab4.exe без ключа восстановления. На рисунке видно, что в верхней части окна расположен счётчик, в котором показано, сколько раз было вызвано прерывание.



Программа завершилась, а счётчик до сих пор работает, собственно, как и должно быть.

3)Запуск lab3v1.com для проверки того, осталось ли прерывание в памяти. Область памяти с кодом нашего прерывание выделена красным.



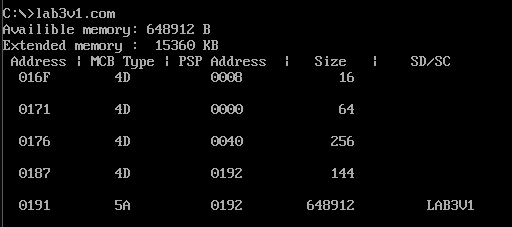
4)Повторный запуск lab4.exe. Ожидается сообщение о том, что прерывание наше уже находится в памяти.



5)Запуск lab4.exe с ключом /un для восстановления стандартного обработчика прерывания.



6)Запуск lab3v1.com для проверки того, что память была освобождена.



**Контрольные вопросы**

а) Как реализован механизм прерывания от часов?

Сначала сохраняется содержимое регистров, потомопределяется источник прерывания, по номеру которого определяется смещение в таблице векторов прерывания, сохраняется в CS:IP, передаётся управление по адресу CS:IP и происходит выполнение обработчика, и в конце происходит возврат управления прерванной программе. Аппаратное прерывание от таймера происходит каждые 55 мс.

б) Какого типа прерывания использовались в работе?

1)аппаратные прерывания

2)прерывания функций DOS(21h)

3)прерывания функций BIOS(10h)

**Вывод:**

Построен обработчик прерывания от сигналов таймера. Изучены дополнительные функции работы с памятью: установка программы-резидента и его выгрузка из памяти.