**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Факультет компьютерных технологий и информатики**

**Кафедра математического обеспечения и применения ЭВМ**

отчет

**по лабораторной работе №4**

**по дисциплине «Операционные системы»**

**Тема: Обработка стандартных прерываний.**

|  |  |  |
| --- | --- | --- |
| Студент гр. 6381 |  | Федянин Н.И. |
| Преподаватель |  | Губкин А.Ф. |

Санкт-Петербург

2018

## **Цель работы**

Целью работы является построение обработчика прерываний сигналов таймера.

## **Структура данных управляющей программы**

Таблица 1 – Назначение переменных

|  |  |
| --- | --- |
| **Название** | **Назначение** |
| Loaded | Хранение сведений о том, что пользовательское прерывание загружено |
| Unloaded | Хранение сведений о том, что пользовательское прерывание выгружено |
| AlreadyLoaded | Хранение сведений о том, что пользовательское прерывание уже было загружено |
| Signature | Строка для проверки пользовательского прерывания |
| KeepCS | Хранение значения из CS |
| KeepIP | Хранение значения из IP |
| KeepPSP | Хранение сегментного адреса PSP |
| DeleteFlag | Изначально хранит ноль, когда же программа будет запущена с ключом /un, DeleteFlag будет хранить единицу. |
| Counter | Накопление прерываний |
| KeepSP | Хранение значения из SP |
| KeepSS | Хранение значения из SS |

## **Функции управляющей программы**

Таблица 2 – Назначение функций

|  |  |
| --- | --- |
| **Название** | **Назначение** |
| PRINT\_A\_STR | Вывод строки на экран |
| OutputAL | Вывод символа из AL |
| GetCursor | Получение позиции курсора |
| SetCursor | Установка позиции курсора |
| Rout | Процедура обработчика прерывания |
| SetINT | Функция установки пользовательской функции прерывания |
| CheckSignature | Проверка сигнатур. Установка пользовательского обработчика прерывания и вывод сообщения в Loaded, исходя из условия их не совпадения, либо вывод сообщения в AlreadyLoaded в обратном случае. В случае запуска с ключом /un и совпадении сигнатур выполняет присвоение DeleteFlag = 1 |

## **Последовательность действий, выполняемых утилитой**

* Происходит проверка, установлено ли пользовательское прерывание с вектором 1Ch.
* Если это не так, устанавливается резидентная функция для обработки прерывания и настраивается вектор прерываний.
* Если прерывание установлено, то выводится соответствующее сообщение и осуществляется выход.
* Выгрузка прерывания по соответствующему значению параметра в командной строке /un (восстановление стандартного вектора прерываний и освобождение памяти, занимаемой резидентом).

**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 | 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

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

# **Ход работы**

1. Написан и отлажен **.EXE** модуль. Результаты выполнения программы:



Рисунок 1 – Результаты выполнения программы

1. Проверка установки резидентного обработчика прерывания 1Ch с помощью программы s1.com из лабораторной работы №3. Результаты:

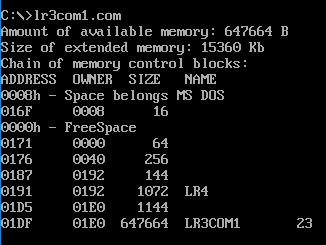


Рисунок 2 – Результаты выполнения программы

1. Попытка запустить программу lr4.exe еще раз. Результаты:



Рисунок 3 – Результаты выполнения программы

1. Запуск программы lr4.exe с ключом выгрузки, проверка выгрузки с помощью s1.com. Результаты выполнения:



Рисунок 4 – Результаты выполнения программы

1. Ответы на контрольные вопросы:
   1. *Как реализован механизм прерывания от часов?*

Приблизительно 18,2 раза в секунду системный таймер последним своим действием выполняет вызов прерывания INT 1Ch. После инициализации системы вектор INT 1Ch указывает на команду IRET, т.е. ничего не выполняется. Программа может установить собственный обработчик этого прерывания для того, чтобы выполнять какие-либо периодические действия.

* 1. *Какого типа прерывание использовались в работе?*

В работе использовались аппаратные прерывания (int 1Сh), прерывания функций DOS (int 21h) и прерывания функций BIOS(int 10h).

# **ЗАКЛЮЧЕНИЕ**

В ходе данной лабораторной работы был построен обработчик прерываний сигналов таймера.