**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

# **ПОСТАНОВКА ЗАДАЧИ**

## **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 IS\_LOADED, исходя из условия их не совпадения, либо вывод сообщения в IS\_ALR\_LOADED в обратном случае. В случае запуска с ключом /un и совпадении сигнатур выполняет присвоение DEL\_FLAG = 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).

# **ХОД РАБОТЫ**

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

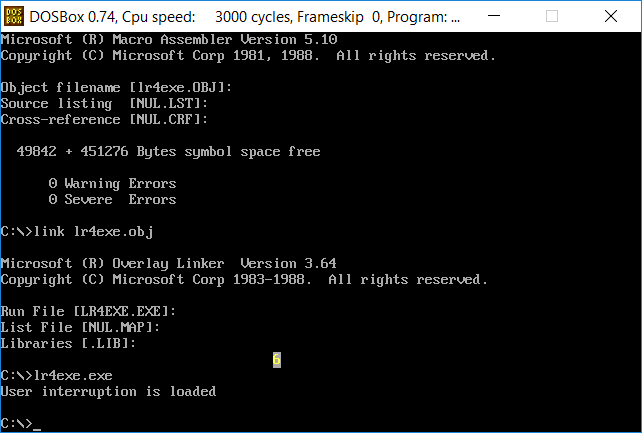


Рисунок 1 – Результаты выполнения программы

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

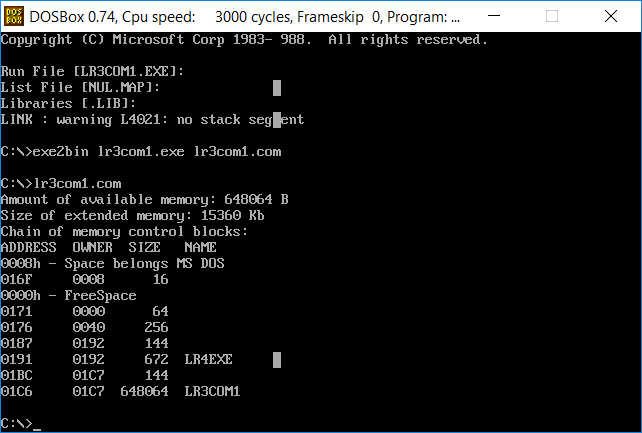


Рисунок 2 – Результаты выполнения программы

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

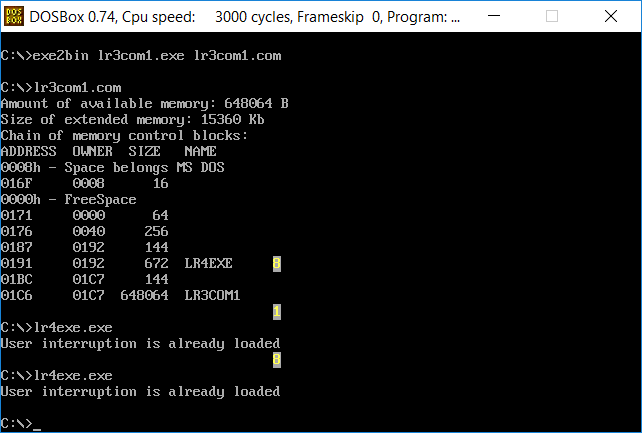


Рисунок 3 – Результаты выполнения программы

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

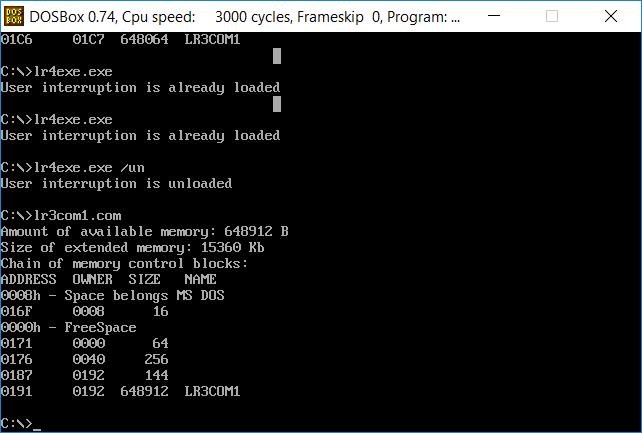


Рисунок 4 – Результаты выполнения программы

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?*

Использовались два типа – аппаратные и программные.

# **ЗАКЛЮЧЕНИЕ**

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