**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 |  | Шарипова Р.А. |
| Преподаватель |  | Губкин А.Ф. |

Санкт-Петербург

2018

**Цель работы:**

В лабораторной работе №4 предлагается построить обработчик прерываний сигналов таймера. Эти сигналы генерируются аппаратурой через определенные 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.

**Функции и структуры данных управляющей программы:**

|  |  |
| --- | --- |
| **Название** | **Назначение** |
| Write\_message | Вывод сообщения на экран |
| My\_1C | Собственный обработчик прерывания для 1C. Вывод в позицию курсора количество вызванных прерываний. |
| Un\_check | Проверяет, не указал ли пользователь флаг «/un» при вызове программы |
| Unload\_interr | Выгружает обработчики прерываний, сохраненные заранее |
| Make\_resident | Оставляет программу резидентной в памяти |
| Is\_loaded | Проверка установки прерывания |

|  |  |
| --- | --- |
| **Название** | **Назначение** |
| Count | Запоминает количество вызванных прерываний |
| Message1 | Сообщение о том, что программа только что была загружена в память резидентной |
| Message2 | Сообщение о том, что программа выгружена из памяти |
| Message3 | Сообщение о том, что программа уже загружена в память |
| Message4 | Сообщение о том, что программа не загружена в память |
| myPSP | переменная для сохранения PSP |
| KEEP\_SP | Переменная для хранения регистра SP |
| KEEP\_SS | Переменная для хранения регистра SS |
| NEW\_STACK | Выделение памяти под новый стек |
| keep\_cs | Хранит сегмент старого прерывания 1c |
| keep\_ip | Смещения старого прерывания 1c |

**Ход работы:**

1. Был написан и отлажен программный модуль .EXE, который выполняет следующие функции:

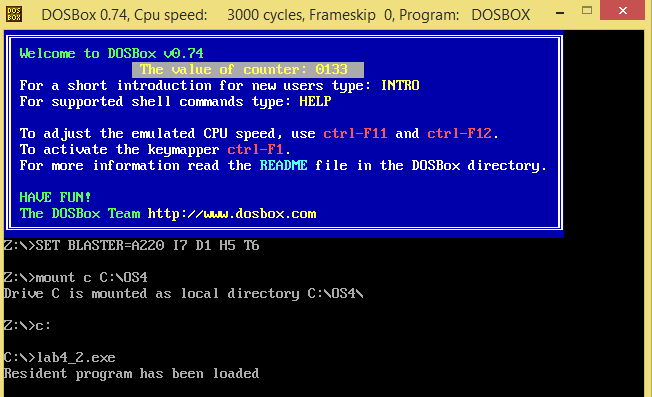
1. Проверяет, установлено ли пользовательское прерывание с вектором 1Сh.
2. Устанавливает резидентную функцию для обработки прерывания и настраивает вектор прерываний, если прерывание не установлено, и осуществляется выход по функции 4Сh прерывания int 21h.
3. Если прерывание установлено, то выводится соответствующее сообщение и осуществляется выход по функции 4Сh прерывания int 21h.
4. Выгрузка прерывания по 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 4Сh прерывания int 21h.

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

1. Сохранить значения 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.

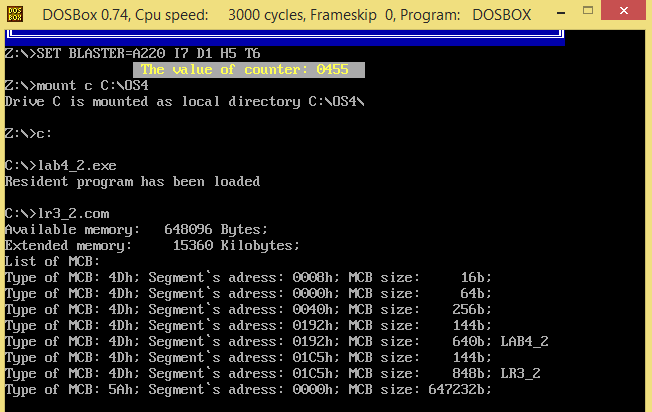
2. Запустила отлаженную программу и убедилась, что резидентный обработчик прерывания 1Ch установлен.

Результат:



*Рисунок 1*

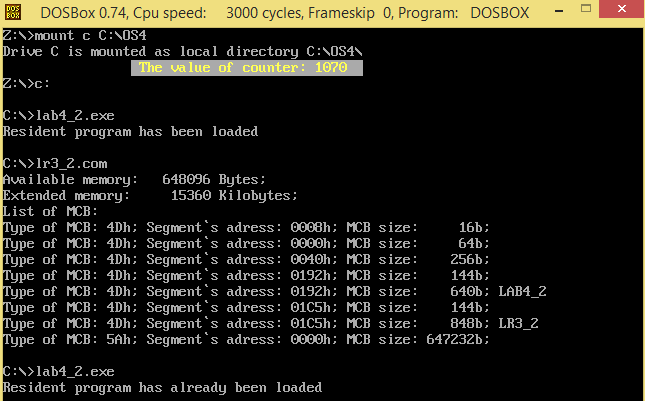
Также проверено размещение прерывания в памяти. Для этого была запущена программа предыдущей лабораторной работы, которая отображает карту памяти в виде списка блоков MCB:



*Рисунок 2*

3. Запустила отлаженную программу еще раз и убедилась, что программа определяет установленный обработчик прерываний.

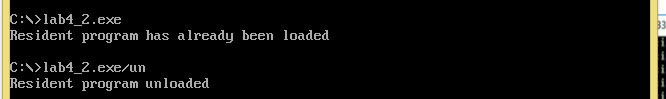
Результат:



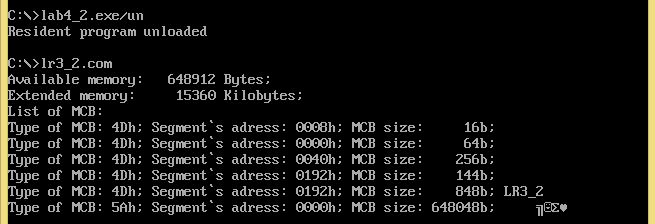
*Рисунок 3*

4. Запустила отлаженную программу с ключом выгрузки /un и убедилась, что резидентный обработчик прерываний выгружен, то есть сообщение на экране не выводятся, а память, занятая резидентом освобождена. Для этого также запускаем программу из предыдущей лабораторной работы.

Результат:



*Рисунок 4*



*Рисунок 5*

**Ответы на контрольные вопросы:**

1. *Как реализован механизм прерывания от часов?*

Каждые 55 мс:

* 1. сохраняется состояние регистров
  2. определяется источник прерывания (по номеру источника прерывания определяется смещение в таблице векторов прерываний)
  3. первые два байта помещаются в IP, второе два байта в CS
  4. передаётся управление по адресу CS:IP (т.е. «запускается» обработчик прерывания)
  5. обработка прерывания
  6. возврат управления прерванной программе

1. *Какого типа прерывания использовались в работе?*

В работе использовались аппаратные прерывания (int 1Ch) и пользовательские прерывания (int 21h, int 10h).

**Вывод:**

В ходе данной лабораторной работы был построен собственный обработчик прерывания для аппаратного прерывания 1Ch, происходящего по сигналу системного таймера.