**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МОЭВМ**

Лабораторная работа №4

**по дисциплине «Операционные системы»**

Тема: «**Обработка стандартных прерываний»**

|  |  |  |
| --- | --- | --- |
| Студентка гр. 6383 |  | Гомонова А.А. |
| Преподаватель |  | Губкин А.Ф. |

Санкт-Петербург

2018

**1. Постановка задачи**

* 1. **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.

* 1. **Сведения о функциях и структурах данных управляющей программы**

*Функции*

|  |  |
| --- | --- |
| **Название** | **Назначение** |
| Write\_message | Вывод сообщения на экран |
| outputAL | Вывод в текущее положение курсора символа из AL |
| SaveCurs | Запоминает в DX текущую позицию курсора |
| SetCurs | Устанавливает курсор в указанную в DX позицию |
| My\_2F | Собственный обработчик прерывания для 2F. Проверяет, была ли программа установлена резидентной в памяти |
| My\_1C | Собственный обработчик прерывания для 1C. Выводит в 33-ую позицию курсора количество прерываний, которые были вызваны (счётчик обновляется каждые 10 раз). Прерывания генерируются системным таймером с частотой примерно 18.2 раза в секунду. |
| Un\_check | Проверяет, не указал ли пользователь флаг «/un» при вызове программы |
| Keep\_interr | Запоминает старые обработчики прерывания, используя функцию 35h прерывания int 21h |
| Load\_interr | Устанавливает новые обработчики прерывания, используя функцию 25h прерывания int 21h |
| Unload\_interr | Восстанавливает сохранённые заранее обработчики прерываний и выгружает резидентную программу |
| Make\_resident | Оставляет программу резидентной в памяти |
| Main | Основная функция |

*Переменные:*

|  |  |
| --- | --- |
| **Название** | **Назначение** |
| Count | Запоминает количество вызванных прерываний |
| flag | Флаг, равный 1, если программа не является резидентной, и 0 – в противном случае |
| Message1 | Сообщение о том, что программа только что была загружена в память резидентной |
| Message2 | Хранит тип текущего блока памяти |
| Message3 | Хранит сегментный адрес текущего блока памяти |
| keep\_1c | Переменная для хранения сегмента и смещения старого прерывания 1c |
| keep\_2f | Переменная для хранения сегмента и смещения старого прерывания 2f |
| keep\_PSP | Переменная для хранения старого значения ES до того, как программа была оставлена резидентной в памяти |

1. **Ход работы**

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.

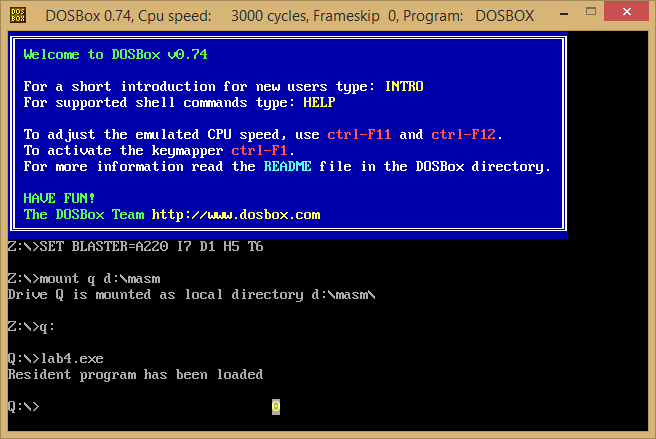
4) Выгрузка прерывания по соответствующему значению параметра в командной строке /un. Выгрузка прерывания состоит в восстановлении стандартного вектора прерываний и освобождении памяти, занимаемой резидентом. Затем осуществляется выход по функции 4Сh прерывания int 21h.

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.

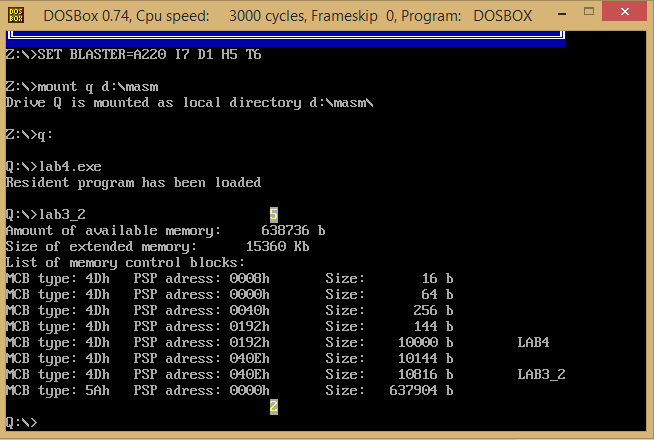
2) При выполнении тела процедуры накапливать общее суммарное число прерываний и выводить на экран. Для вывода на экран следует использовать прерывание int 10h, которое позволяет непосредственно выводить информацию на экран.

1. **Результаты работы программы**



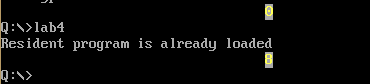
*Рисунок 1*

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*

Запустили отлаженную программу еще раз и убедились, что программа определяет установленный обработчик прерываний.



*Рисунок 3*

Запустили отлаженную программу с ключом выгрузки /un и убедились, 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.

4_1.png

*Рисунок 4*

**

*Рисунок 5*

1. **Ответы на контрольные вопросы**
2. *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).

1. **Вывод**

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