**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

|  |  |  |
| --- | --- | --- |
| Студент гр. 6383 |  | Любчук Д.В. |
| Преподаватель |  | Губкин А.Ф. |

Санкт-Петербург

2017

***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 по соответствующему адресу вектора прерывания. Обработчик прерывания получает управление и выполняет соответствующие действия.

В данной лабораторной работе предлагается построить обработчик прерываний сигнала 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.

***Порядок выполнения работы:***

1. Написать и отладить программный модуль типа .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 4Ch прерывания int 21h.
   3. Если пользовательское прерывание установлено, то выводится соответствующее сообщение и осуществляется выход по функции 4Ch прерывания int 21h.
   4. Выгрузка прерывания по соответствующему значению параметра в командной строке /un.
2. Запустить программу и убедится, что резидентный обработчик прерывания установлен (работа должны отображаться на экране). Также необходимо запустить программу предыдущей лабораторной работы, отображающую карту памяти в виде списка MCB.
3. Запустить программу ещё раз и убедится, что программа распознаёт установленный обработчик прерывания.
4. Запустить программу с ключом выгрузки и убедится, что резидентный обработчик прерывания выгружен, то есть сообщения на экран не выводятся, а память, занятая резидентом, освобождена.

***Сведения о функциях и структурах данных управляющей программы:***

* Функция *GetCurs* определяет текущие позицию и размер курсора
* Функция *SetCurs* устанавливает курсор на заданную позицию
* Функция *wrd\_to\_dec* осуществляет перевод слова, помещенного в ax, в последовательность символов в десятичной системе счисления, помещая результат в si.
* Функция *my\_int* осуществляет обработку прерывания.
* Функция *old\_int\_save* сохраняет сегмент и смещение системного прерывания.
* Функция *set\_new\_int* устанавливает вместо системного пользовательское прерывание
* Функция *load\_my\_int* оставляет прерывание резидентным в памяти
* Функция *delete\_my\_int* восстанавливает системное прерывание и освобождает память, занимаемую резидентом
* Функция *print* осуществляет вывод строки на экран

***Пример работы программы:***

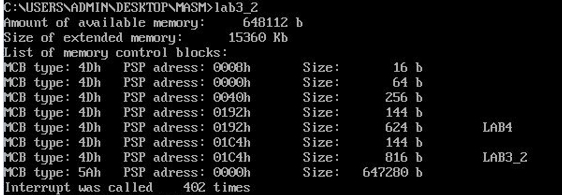
1.Вызов программы с параметром /un оповещает о том, что прерывание не установлено:

D:\Users\Admin\Desktop\Снимок1.JPG

2.Программа выполняет установку прерывания и завершает работу. Результат работы прерывания отображается на экране:



3.После запуска программы, отображающей список MCB:



Даже после завершения программы, область, установленная резидентной из памяти, не высвобождается и функция, реализующая прерывание успешно выполняется.

***Ответы на контрольные вопросы:***

1. Как реализован механизм прерывания от часов?

Во время каждого тика аппаратных часов (каждые 55 миллисекунд или приблизительно 18.2 раз в секунду) происходит аппаратное прерывание от таймера (INT 8h), которое вызывает программное прерывание INT 1Сh.

1. Какого типа прерывания использовались в работе?

INT 10H - стандартного видеосервиса ROM-BIOS.

INT 21H - сервис DOS

INT 1CH: пользовательское прерывание по таймеру