**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 |  | Яковлев Е.А. |
| Преподаватель |  | Губкин А.Ф. |

Санкт-Петербург

2018

**Цель лабораторной работы**

В архитектуре компьютера существуют стандартные прерывания, за которыми закреплены определенные вектора прерываний. Вектор прерываний хранит адрес подпрограммы обработчика прерываний. При возникновении прерывания, аппаратура компьютера передает управление по соответствующему адресу вектора прерывания. Обработчик прерываний получает управление и выполняет соответствующие действия.

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 определенным значением вектора. Таким образом, управление будет передано функции, чья точка входа записана в соответствующий вектор прерывания.

**Постановка задачи**

Шаг 1. Для выполнения лабораторной работы необходимо написать и отладить программный модуль типа .EXE, который выполняет следующие функции:

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

Шаг 2. Далее необходимо запустить отлаженную программу и убедиться, что резидентный обработчик прерывания 1Ch установлен. Работа прерывания должна отображаться на экране, а также необходимо проверить размещение прерывания в памяти. Для этого нужно запустить программу ЛР 3, которая отображает карту памяти в виде с писка блоков MCB.

Шаг 3. Затем необходимо запустить отлаженную программу еще раз и убедиться, что программа определяет установленный обработчик прерываний.

Шаг 4. Далее нужно запустить отлаженную программу с ключом выгрузки и убедиться, что резидентный обработчик прерывания выгружен, то есть сообщения на экран не выводятся, а память, занятая резидентом освобождена. Для этого также следует запустить программу ЛР 3.

Оформить отчёт и ответить на контрольные вопросы.

Процедуры, которые используются в программе.

|  |  |
| --- | --- |
| ROUT | Функция обработчика прерывания. |
| IS\_LOADED | Функция проверки закружен ли резидент в память. |
| IS\_UNLOADED | Функция проверки ключа выгрузки резидента из памяти. |
| RES\_LOAD | Функция загрузки резидента в память. |
| RES\_UNLOAD | Функция выгрузки резидента из памяти. |
| PRINT | Функция печати на экран. |

Переменные, которые используются в программе.

|  |  |  |
| --- | --- | --- |
| resNotSet | db | Применяется для вывода информации о том, что резидент не загружен в память. |
| resUnload | db | Применяется для вывода информации о том, что резидент выгружен из памяти. |
| resAlrSet | db | Применяется для вывода информации о том, что резидент уже загружен в память. |
| resLoad | db | Применяется для вывода информации о том, что резидент загружен в память. |
| COUNTER | db | Счетчик обработчика прерывания |
| INTER\_ADR | db | Применяется для проверки того, загружен ли уже резидент в память или нет. |
| KEEP\_IP | dw | Применяется для запоминания смещения вектора прерывания. |
| KEEP\_CS | dw | Применяется для запоминания сегмента вектора прерывания. |

**Ход выполнения работы**

Результаты выполнения программы представлены на рисунках 1.1-1.6:

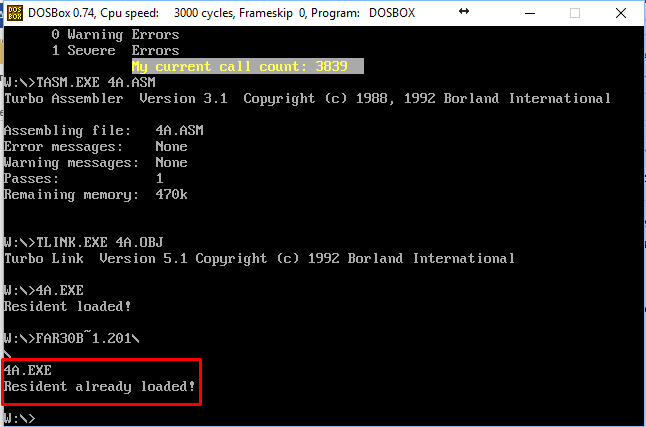


Рис.1.1

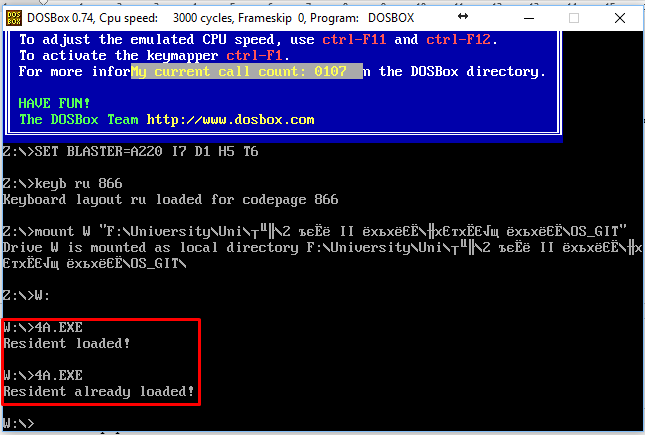


Рис.1.2

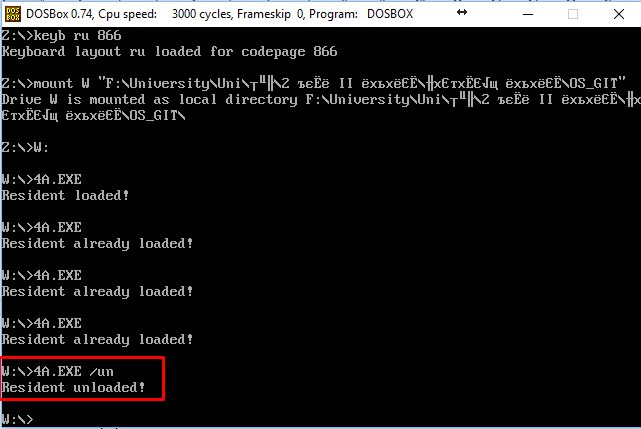


Рис.1.3

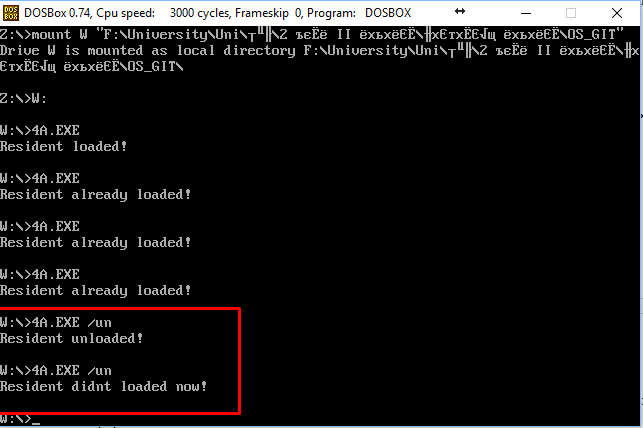


Рис.1.4

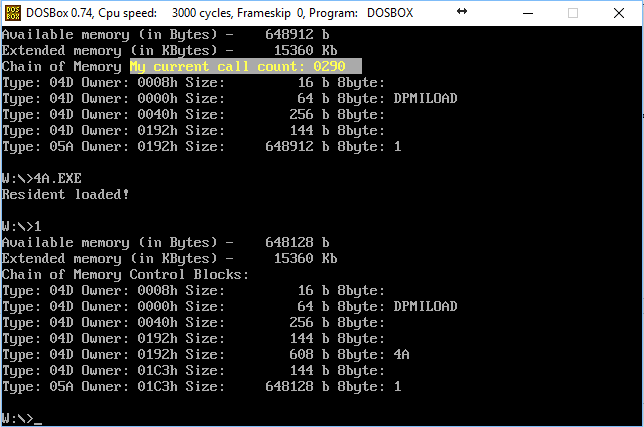


Рис.1.5

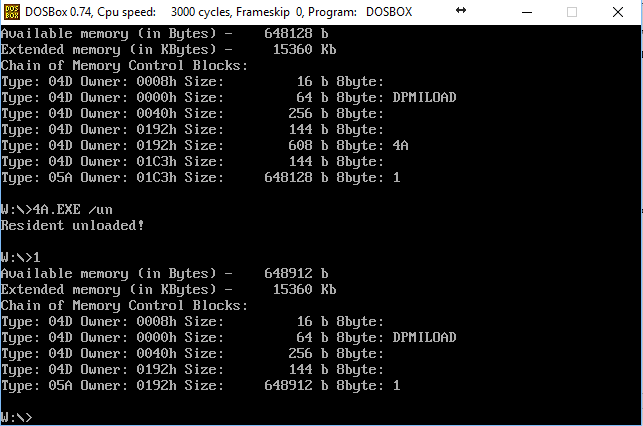


Рис.1.6

**Контрольные вопросы**

1. *Как реализован механизм прерывания от часов?*

При каждом такте таймера (часов) происходит(каждые 55 миллисекунд или приблизительно 18.2 раз в секунду): сначала сохраняется состояние регистров, затем определяется источник прерывания (по номеру источника прерывания определяется смещение в таблице векторов прерываний), далее эти данные сохраняются, т. е. - первые два байта помещаются в IP, второе два байта в CS, далее запускается обработчик прерывания по адресу CS:IP и обрабатывается прерывание, в конце происходит возврат управления прерванной программе

1. *Какого типа прерывания использовались в работе?*

Были использованы пользовательские прерывания, такие как int 10h(стандартного видеосервиса ROM-BIOS) и int 21h(сервис DOS) и аппаратные прерывания (1Ch(пользовательское прерывание по таймеру)).

**Вывод**

В результате выполнения данной лабораторной работы были исследованы организация и управление прерываниями. Была написана программа, в которой построен обработчик прерываний сигналов таймера.