**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. 7383 |  | Лосев М.Л. |
| Преподаватель |  | Ефремов М.А. |

Санкт-Петербург

2018

## Постановка задачи.

**Цель работы:** в архитектуре компьютера существуют стандартные прерывания, за которыми закреплены определенные вектора прерываний. Вектор прерывания 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 предлагается построить обработчик прерываний сигналов таймера. Эти сигнал 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.

## Сведения об используемых функциях и структурах данных.

BYTE\_TO\_HEX – переводит значение регистра AL в его запись в шестнадцатеричной с/с, помещает ее в AX

WRD\_TO\_HEX – переводит значение регистра AX в его запись в шестнадцатеричной с/с, помещает ее в память так, что DI указывает на младшую цифру.

BYTE\_TO\_DEC – переводит значение регистра AL в его запись в десятичной ичной с/с, помещает результат в память так, что SI указывает на младшую цифру.

**OUTPUT\_PROC** – вызывает прерывание DOS вывода строки.

**outputBP** –

**getCurs** –

**setCurs** –

**ROUT** –

**KEEP\_HANDLER** –

**RECOVER\_HANDLER** –

**SET\_HANDLER** –

**KEEP** –

**check\_handler** –

**check\_tail** –

**unload\_handler**–

## Последовательность действий, выполняемых утилитой.

Утилита выводит количество доступной памяти, размер расширенной памяти и таблицу всех блоков управления памятью.

Первая модификация перед выводом таблицы освобождает память, которую не использует.

Вторая модификация освобождает неиспользуемую память и выделяет себе из освобожденной памяти 64 кб.

Третья модификация запрашивает выделение 64 кб памяти (память не может быть выделена, так как нет невыделенной памяти) и только после этого освобождает память.

**Результаты.**

Скриншоты с результатами представлены на рис. 1 – рис. 4.

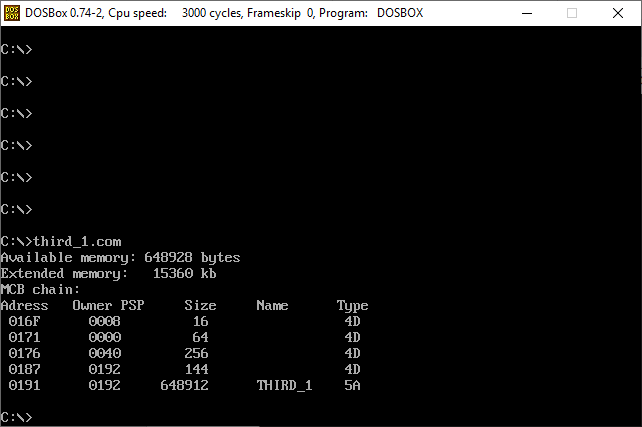


Рисунок 1. До изменений программы

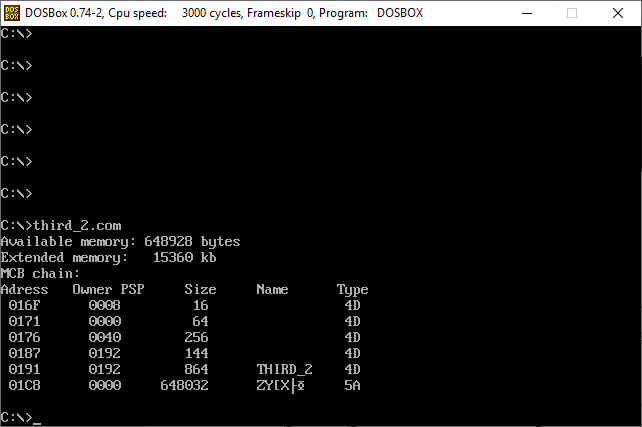


Рисунок 2. Первая модификация

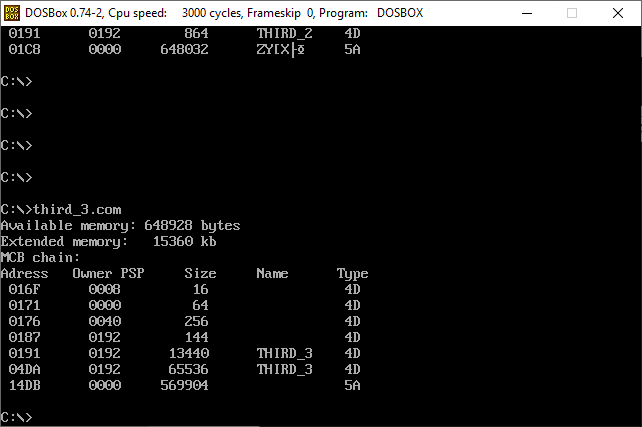


Рисунок 3. Вторая модификация

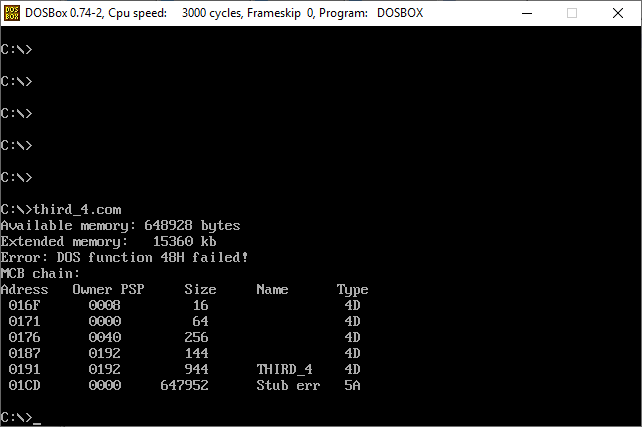


Рисунок 4. Третья модификация

**Контрольные вопросы.**

1. *Что означает «доступный объем памяти»?*

Это наибольший объем памяти, которая может быть предоставлена программе.

1. *Где MCB-блок вашей программы в списке?*

У программы есть во всех случаях не менее двух блоков: первый во всех четырех случаях имеет сегментный адрес 0187 и управляет областью переменных среды, а второй во всех случаях имеет адрес 0191 и управляет областью, выделенной для программы. В третьем случае есть еще третий блок, который управляет выделенной областью размера 64 кб.

1. *Какой размер памяти занимает программа в каждом случае?*

В первом случае – всю доступную память, 648912 байт. Во втором – ровно столько, сколько ей необходимо, это 864 байта. В третьем – 864 минимально необходимых ей и еще 64 кб, которые были выделены по запросу. В четвертом – всю доступную память, 944 байта (сначала запросили память, но она не выделилась, а потом освободили неиспользуемую память.).

**Заключение.**

Была исследована организация управления динамическими разделами нестраничной памяти в DOS: внутренние структуры данных ОС (List of lists и MCB) и работа функций управления памятью ядра ОС (функции 4Аh и 48h). List of lists содержит указатель на первый MCB. Каждый MCB занимает 16 байт и содержит адрес PSP программы, которая владеет блоком памяти, размер блока памяти, имя программы, которая владеет блоком, и тип: 4D, если блок не последний или 5A, если блок последний. Функция 4Аh освобождает память, а функция 48h выделяет память. Если программа запрашивает у ОС с помощью функции 48h выделение большего объема памяти, чем ОС может выделить, то паять не выделяется и устанавливается флаг CF.