**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

отчет

**по лабораторной работе №3**

**по дисциплине «Операционные системы»**

**Тема: Исследование организации управления основной памятью**

|  |  |  |
| --- | --- | --- |
| Студентка гр. 6381 |  | Вероха В. Н. |
| Преподаватель |  | Губкин А. Ф. |

Санкт-Петербург

2018

**Постановка задачи**

1. Цель работы:

Для исследования организации управления памятью необходимо ориентироваться на тип основной памяти, реализованный в компьютере и способ организации, принятый в ОС. В лабораторной работе рассматривается не страничная память и способ управления динамическими разделами. Для реализации управления памятью в этом случае строится список занятых и свободных участков памяти. Функции ядра, обеспечивающие управление основной памятью, просматривают и преобразуют этот список.

В лабораторной работе исследуются структуры данных и работа функций управления памятью ядра операционной системы.

1. Описание функций:

|  |  |
| --- | --- |
| Название | Назначение |
| TETR\_TO\_HEX | перевод половины байта в 16-ую систему счисления |
| BYTE\_TO\_HEX | перевод байта регистра AL в 16-ую систему счисления в AX |
| WRD\_TO\_HEX | перевод двух байт регистра AX в 16-ую систему счисления в DI |
| BYTE\_TO\_DEC | перевод байта регистра AL в 10-ую систему счисления, помещая результат в SI |
| GET\_A\_MEMORY | Получение размера доступной памяти |
| GET\_E\_MEMORY | Получение размера расширенной памяти |
| GET\_MCB\_DATA | Получение данных одного MSB блока |
| GET\_ALL\_MSB\_DATA | Получение данных со всех MSB блоков |
| PRINT | Вывод строки на экран |

1. Описание данных:

|  |  |  |
| --- | --- | --- |
| Название | Тип | Назначение |
| AVAILABLE\_MEMORY | db | Доступная память |
| EXTENDED\_MEMORY | db | Расширенная память |
| TABLE\_TITLE | db | Таблица MCB |
| DATA\_IN\_TABLE | db | Строка для вывода |

1. Общие сведения:

Учёт занятой и свободной памяти ведется при помощи списка блоков управления памятью MCB (Memory Control Block). MCB занимает 16 байт (параграф) и располагается всегда с адреса кратного 16 (адрес сегмента ОП) и находится в адресном пространстве непосредственно перед тем участком памяти, которым он управляет.

MCB имеет следующую структуру:

|  |  |  |
| --- | --- | --- |
| Смещение | Длина поля (байт) | Содержимое поля |
| 00h | 1 | тип MCB:  5Ah, если последний в списке,  4Dh, если не последний |
| 01h | 2 | Сегментный адрес PSP владельца участка памяти, либо  0000h - свободный участок,  0006h - участок принадлежит драйверу OS XMS UMB  0007h - участок является исключенной верхней памятью драйверов  0008h - участок принадлежит MS DOS  FFFAh - участок занят управляющим блоком 386MAX UMB  FFFDh - участок заблокирован 386MAX  FFFEh - участок принадлежит 386MAX UMB |
| 03h | 2 | Размер участка в параграфах |
| 05h | 3 | Зарезервирован |
| 08h | 8 | "SC" - если участок принадлежит MS DOS, то в нем системный код  "SD" - если участок принадлежит MS DOS, то в нем системные данные |

По сегментному адресу и размеру участка памяти, контролируемого этим MCB можно определить местоположение следующего MCB в списке.

Адрес первого MCB хранится во внутренней структуре MS DOS, называемой "List of Lists" (список списков). Доступ к указателю на эту структуру можно получить, используя функцию f52h "Get List of Lists" int 21h. В результате выполнения этой функции ES:BX будет указывать на список списков. Слово по адресу ES:[BX-2] и есть адрес самого первого MCB.

Размер расширенной памяти находится в ячейках 30h, 31h CMOS. CMOS это энергонезависимая память, в которой хранится информация о конфигурации ПЭВМ. Объем памяти составляет 64 байта. Размер расширенной памяти в Кбайтах можно определить обращаясь к ячейкам CMOS следующим образом:

mov AL,30h ; запись адреса ячейки CMOS

out 70h,AL

in AL,71h ; чтение младшего байта

mov BL,AL ; размера расширенной памяти

mov AL,31h ; запись адреса ячейки CMOS

out 70h,AL

in AL,71h ; чтение старшего байта размера расширенной памяти

**Ход работы:**

Результаты работы представлены на рисунках 1, 2, 3, 4.

1. LAB3\_1.COM

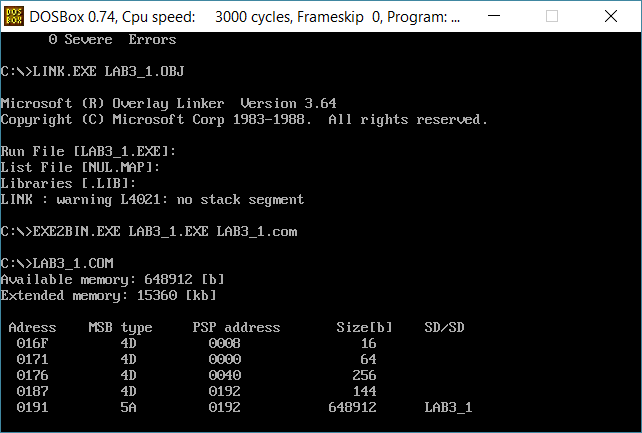


Рис. 1

Все доступные 648912 байт отдаются программе.

1. LAB3\_2.COM

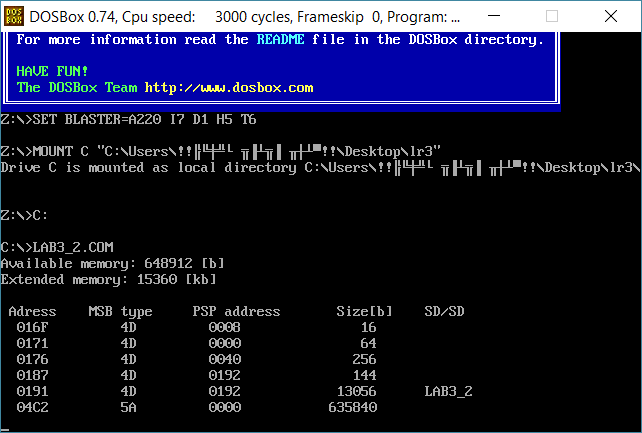


Рис. 2

При изменении исходного кода программы освобождается занимаемая ее память. Создается новый блок, который обозначен как свободный участок, размером 635840 байт.

1. LAB3\_3.COM

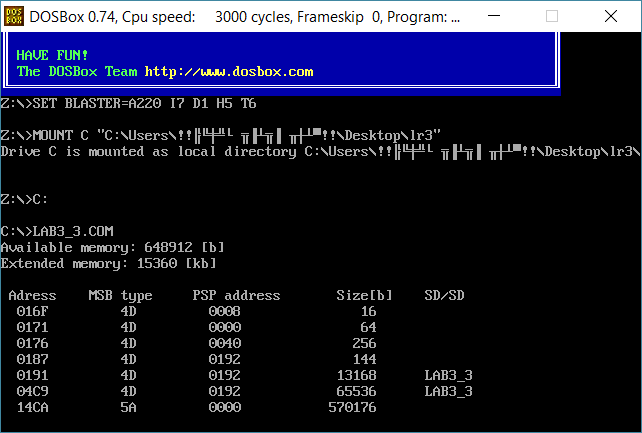


Рис. 3

Изменяем код программы, после чего происходит освобождение памяти (то есть происходит то же, что и в первом случае). Затем программа запрашивает 64 Кбайт (65536 байт) памяти. На свободном участке создается новый блок, который следует за основным блоком программы и занимает 65536 байт.

1. LAB3\_4.COM

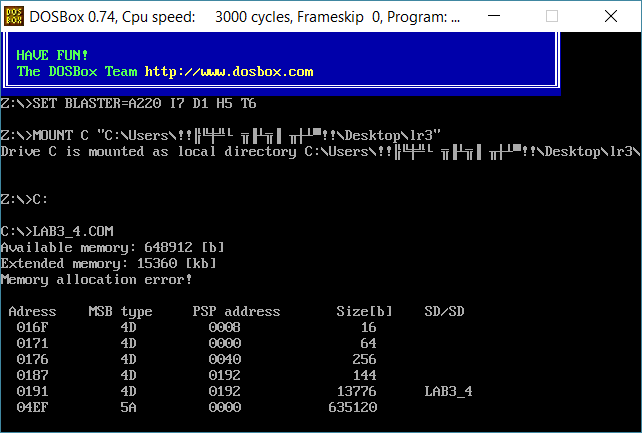


Рис. 4

Изменение кода. Далее запрос 64 Кбайт до освобождения памяти. Но вылетает ошибка, так как запрос памяти происходит в момент, когда вся доступная память занята программой. Затем происходит освобождение памяти, аналогично второму случаю.

**Ответы на контрольные вопросы:**

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

Доступный объём памяти – это максимальный доступный для использования программами объем памяти.

1. *Где MCB блок Вашей программы в списке?*

В 1- ой, 2 - ой и четвёртой программах MCB блок имеет адрес 0191h.

В 3 - ей программе два MCB блока: 1 - ый имеет адрес 0191h, а 2 - ой – 04CFh.

Так же в каждой из программ присутствует еще один блок MCB (для управления памятью для области среды программы), имеющий адрес 0187h и размер 144 б.

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

LAB3\_1.COM 648912 байт.

LAB3\_2.COM 13056 байт.

LAB3\_3.COM 13168 байт.

LAB3\_4.COM 13776 байт.

**Заключение:**

В ходе работы было проведено исследование структуры данных и работы функций управления памятью ядра операционной системы, а также рассмотрены не страничная память и способы управления динамическими разделами.