**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**

|  |  |  |
| --- | --- | --- |
| Студент гр. 6383 |  | Липендин А.А. |
| Преподаватель |  | Губкин А. Ф. |

Санкт-Петербург

2018

**1. 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. **Теоретические 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

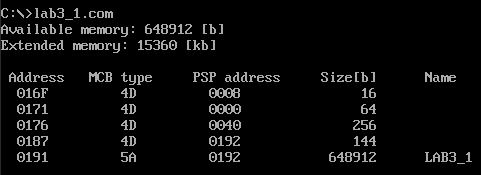
out 70h,AL

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

1. **Результаты работы программы**

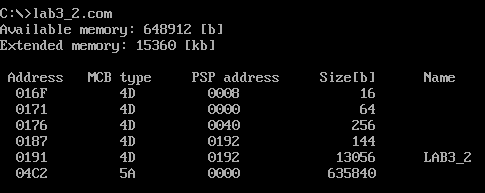
Результаты работы представлены на рисунках 1, 2, 3, 4.

1. lab3\_1.com – размер блока, отведенного под программу, равен 648912 байта.



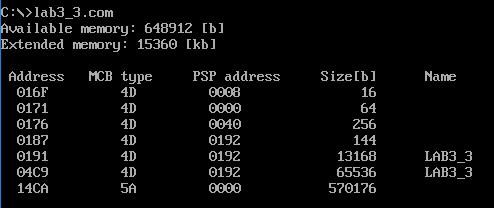
*Рис. 1 Результат работы программы lab3\_1.com*

1. lab3\_2.com – размер блока, отведенного под программу, равен 13056 байтов, при этом создался еще один блок - свободный участок памяти размером 635840 байтов.



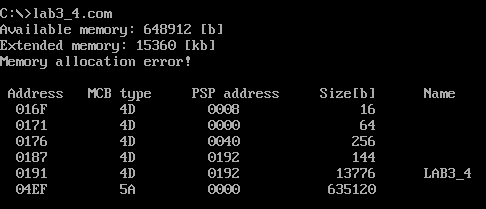
*Рис. 2 Результат работы программы lab3\_2.com*

1. lab3\_3.com – размер блока, отведенного под программу, равен 13168 байтов, также помимо блока со свободной памятью, размер которого уменьшился на 64 Кб и стал 570176 байтов, образовался новый блок, запрошенный программой, весом 64 Кб (65536 байтов).



*Рис. 3 Результат работы программы lab3\_3.com*

1. lab3\_4.com – размер блока, отведенного под программу, равен 13776 байтов, затем программа запрашивает выделение 64 Кб памяти до освобождения неиспользуемой памяти, в связи с чем появляется текст «Memory allocation error!», сообщающий об ошибке.



*Рис. 4 Результат работы программы lab3\_4.com*.

1. **Ответы на контрольные вопросы:**
2. *Что означает «доступный объём памяти»?*

Доступный объем памяти – количество памяти, не занятое другими программами, которое загрузчик выделяет программе при загрузке в основную память, после чего программа сама распоряжается этой памятью с помощью тех же функций, что использовал загрузчик. Поэтому программа может освободить не используемую память, или запросить расширение текущего объема памяти. Если программа запрашивает слишком большой объем памяти (больше, чем размер ее доступной памяти) система откажет ей в этом.

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

Напротив блока программы написано имя программы

В первом случае (рисунок 1) – последняя строчка (занимает всю доступную память)

Во втором случае (рисунок 2) – предпоследняя строчка. Неиспользованная память выделена в отдельный блок.

В третьем случае (рисунок 3) – второй и третий блоки снизу. Второй снизу блок относится к программе, так как она запрашивает выделение дополнительных 64 Кб памяти для себя.

В четвертом случае (рисунок 4) – аналогия со вторым случаем, так как выделение дополнительных 64 Кб памяти под программу закончилось неудачей.

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

LAB3\_1.COM 648912 байт.

LAB3\_2.COM 648912 – 635840 – 16 = 13056 байт.

LAB3\_3.COM 648912 – 570176 – 65536 - 32 = 13168 байт.

LAB3\_4.COM 648912 – 635120 – 16 = 13776 байт.

1. **Вывод**

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.