

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МОЭВМ**

**ОТЧЕТ**  
по лабораторной работе №3  
по дисциплине «Операционные системы»  
на тему: Исследование организации управления основной памятью

Студент гр. 8381

\_\_\_\_\_

Сахаров В.М.

Преподаватель

\_\_\_\_\_

Ефремов М.А.

Санкт-Петербург

2020

### **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 нестраничная 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.

### **Описание функций:**

PRINT_AVL_MEMORY	Получение размера доступной памяти
PRINT_EXT_MEMORY	Получение размера расширенной памяти
PRINT_MCB	Получение данных об MSB блоках
WRITE	Вывод строки из адреса DX на экран
DEC_WORD_PRINT	Вывод десятичного числа из AX на экран
HEX_BYTE_PRINT	Вывод шестнадцатеричного числа из AL (1 байт) на экран
HEX_WORD_PRINT	Вывод шестнадцатеричного числа из AX (2 байта) на экран
FREE_MEMORY	Вспомогательная функция для освобождения памяти
ALLOC_MEMORY	Вспомогательная функция для выделения памяти

### Необходимые сведения и описание структур данных:

Учет 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.

Таблица 1 — Структура MCB

Смещение	Длина (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

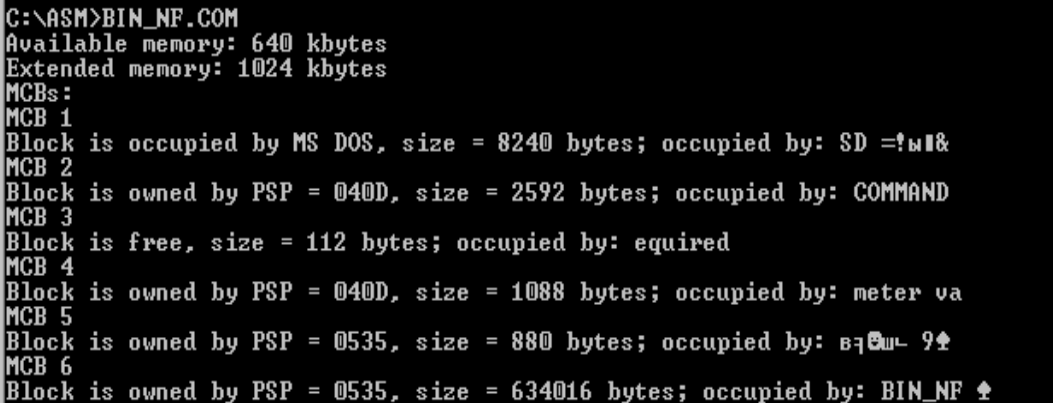
выполнения этой функции ES:BX будет указывать на список списков. Слово по адресу ES:[BX-2] и есть адрес самого первого MCB

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:

Программа выводит на экран количество доступной и расширенной памяти, а также цепочку блоков управления MCB. Результат работы представлен на рисунке 1.



```
C:\ASM>BIN_NF.COM
Available memory: 640 kbytes
Extended memory: 1024 kbytes
MCBs:
MCB 1
Block is occupied by MS DOS, size = 8240 bytes; occupied by: SD =!m!&
MCB 2
Block is owned by PSP = 040D, size = 2592 bytes; occupied by: COMMAND
MCB 3
Block is free, size = 112 bytes; occupied by: equired
MCB 4
Block is owned by PSP = 040D, size = 1088 bytes; occupied by: meter va
MCB 5
Block is owned by PSP = 0535, size = 880 bytes; occupied by: вт@ш- 9#
MCB 6
Block is owned by PSP = 0535, size = 634016 bytes; occupied by: BIN_NF #
```

Рисунок 1 — результат работы программы bin\_nf.com

Из рисунка видно, что программа занимает максимум памяти, потому что при запросе размера доступной памяти мы выделяем, по сути, столько памяти, сколько возможно.

```

C:\ASM>BIN_F.COM
Available memory: 640 kbytes
Extended memory: 1024 kbytes
MCBs:
MCB 1
Block is occupied by MS DOS, size = 8240 bytes; occupied by: SD =!MI&
MCB 2
Block is owned by PSP = 040D, size = 2592 bytes; occupied by: COMMAND
MCB 3
Block is free, size = 112 bytes; occupied by: equired
MCB 4
Block is owned by PSP = 040D, size = 1088 bytes; occupied by: meter va
MCB 5
Block is owned by PSP = 0535, size = 880 bytes; occupied by: B7QW- 9#
MCB 6
Block is owned by PSP = 0535, size = 1136 bytes; occupied by: BIN_F #
MCB 7
Block is free, size = 632864 bytes; occupied by: M?&|c%Q

```

Рисунок 2 — результат работы программы bin\_f.com

В данном случае мы освобождаем память. В итоге остается столько памяти, сколько занимает программа. После освобождения памяти, как видно на рисунке, есть блок свободной памяти, из которого, если вдруг нам потребуется ещё, будет выделяться память.

```

C:\ASM>BIN_NF64.COM
Available memory: 640 kbytes
Extended memory: 1024 kbytes
MCBs:
MCB 1
Block is occupied by MS DOS, size = 8240 bytes; occupied by: SD =!MI&
MCB 2
Block is owned by PSP = 040D, size = 2592 bytes; occupied by: COMMAND
MCB 3
Block is free, size = 112 bytes; occupied by: equired
MCB 4
Block is owned by PSP = 040D, size = 1088 bytes; occupied by: meter va
MCB 5
Block is owned by PSP = 0535, size = 880 bytes; occupied by: B7QW- 9#
MCB 6
Block is owned by PSP = 0535, size = 1152 bytes; occupied by: BIN_NF64
MCB 7
Block is owned by PSP = 0535, size = 65536 bytes; occupied by: {e|| n
MCB 8
Block is free, size = 567296 bytes; occupied by: no info

```

Рисунок 3 — результат работы программы bin\_nf64.com

```

F:\>bin_nf64
Available memory: 640 kbytes
Extended memory: 15360 kbytes
MCBs:
MCB 1
Block is occupied by MS DOS, size = 16 bytes; occupied by: no info
MCB 2
Block is free, size = 64 bytes; occupied by: no info
MCB 3
Block is owned by PSP = 0040, size = 256 bytes; occupied by: no info
MCB 4
Block is owned by PSP = 0192, size = 144 bytes; occupied by: no info
MCB 5
Block is owned by PSP = 0192, size = 1152 bytes; occupied by: BIN_NF64
MCB 6
Block is owned by PSP = 0192, size = 65536 bytes; occupied by: BIN_NF64
MCB 7
Block is free, size = 582192 bytes; occupied by: no info

```

Рисунок 4 — результат работы программы bin\_nf64.com в DOS-BOX, с работающими названиями блоков

В данном случае мы сначала выделяем всю доступную память, потом освобождаем то, что не нужно. Затем запрашиваем блок памяти 64 кб, в итоге система выделяет нам ещё 64 кб памяти.

```

C:\ASM>BIN_F64.COM
Some error occurs during memory allocating.
Available memory: 640 kbytes
Extended memory: 1024 kbytes
MCBs:
MCB 1
Block is occupied by MS DOS, size = 8240 bytes; occupied by: SD =!wI&
MCB 2
Block is owned by PSP = 040D, size = 2592 bytes; occupied by: COMMAND
MCB 3
Block is free, size = 112 bytes; occupied by: equired
MCB 4
Block is owned by PSP = 040D, size = 1088 bytes; occupied by: meter va
MCB 5
Block is owned by PSP = 0535, size = 880 bytes; occupied by: в70ш- 9#
MCB 6
Block is owned by PSP = 0535, size = 1152 bytes; occupied by: BIN_F64
MCB 7
Block is free, size = 632848 bytes; occupied by: {e#||  п

```

Рисунок 5 — результат работы программы bin\_f64.com

В данном случае мы выделяем всё доступную память, а затем ещё запрашиваем 64 кб. В результате возникает ошибка. Она возникает из-за того, что мы в первый раз уже выделили всё доступную память, т.е. больше выделить уже нельзя, но мы всё равно пытаемся, и в итоге получаем ошибку.

### Контрольные вопросы:

1. Что означает «доступный объём памяти»?

Ответ: Это максимальный объем памяти, который может быть доступен программе.

2. Где MCB блок Вашей программы в списке?

Ответ: принадлежность блока памяти можно определить, по адресу его владельца (расположенному со смещением в один байт в MCB). В данном случае это 0535. Также название модуля-владельца может содержаться в последних восьми байтах MCB. Как видно из вывода программы, ей обычно принадлежит два блока, первый из которых имеет фиксированный размер в 880 байтов, а второй зависит от размера исходного кода. Также программе будет принадлежать запрошенная и выделенная память.

3. Какой размер памяти занимает программа в каждом случае?

Ответ:

bin\_nf.com 634016 байт.

bin\_f.com 1136 байт.

bin\_nf64.com 1152 байт (без блока в 64кб).

bin\_f64.com 1152 байт.