

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

ОТЧЕТ

по лабораторной работе №3

по дисциплине «Операционные системы»

Тема: «Исследование организации управления основной памятью»

Студент гр. 8381

Сосновский Д.Н.

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.

Основные теоретические положения.

Учет занятой и свободной памяти ведётся 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.

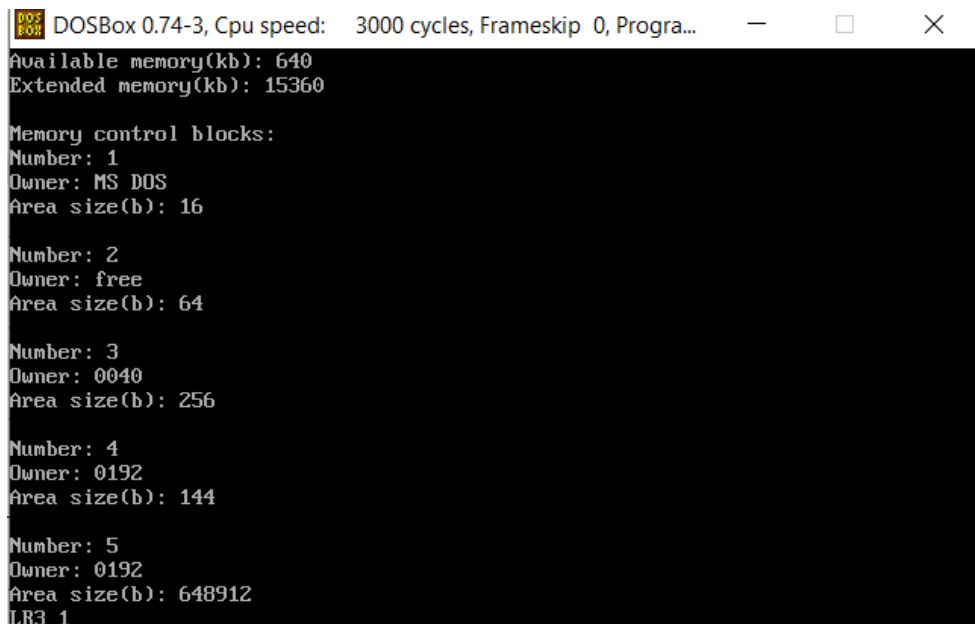
Размер расширенной памяти находится в ячейках 30h, 31h CMOS. CMOS это энергонезависимая память, в которой хранится информация о конфигурации ПЭВМ. Объем памяти составляет 64 байта.

Ход работы.

Мною был разработан исходный код .COM модуля, который выводит на экран следующую информацию:

- 1) Количество доступной памяти
- 2) Размер расширенной памяти
- 3) Цепочку блоков управления памятью

Результаты выполнения работы программы представлены на рисунке 1.



```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
Available memory(kb): 640
Extended memory(kb): 15360

Memory control blocks:
Number: 1
Owner: MS DOS
Area size(b): 16

Number: 2
Owner: free
Area size(b): 64

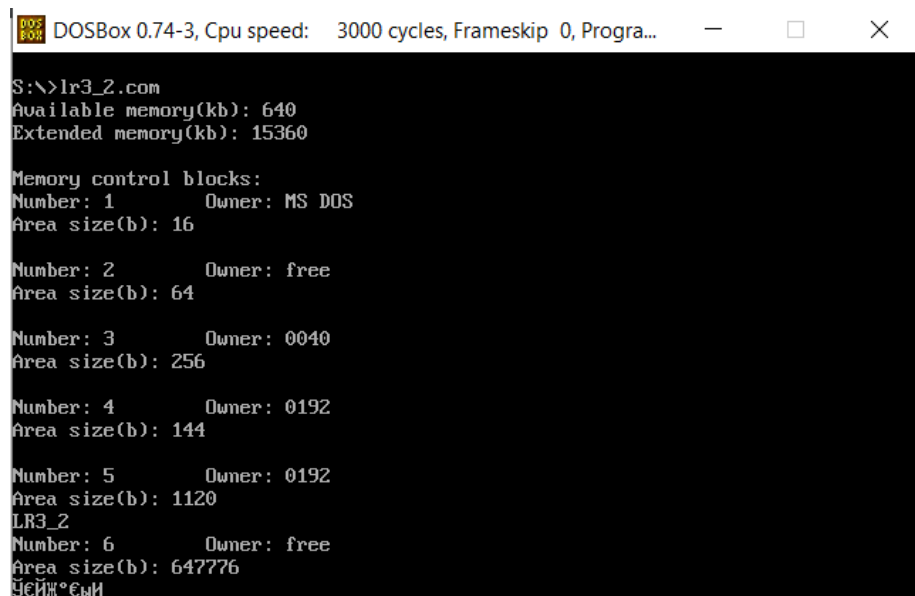
Number: 3
Owner: 0040
Area size(b): 256

Number: 4
Owner: 0192
Area size(b): 144

Number: 5
Owner: 0192
Area size(b): 648912
LR3_1
```

Рисунок 1 – результат выполнения программы (LR3_1.com)

Далее эта программа была изменена таким образом, чтобы она освобождала память, которую она не занимает. Результат выполнения этой версии программы представлен на рисунке 2.



```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
S:\>lr3_2.com
Available memory(kb): 640
Extended memory(kb): 15360

Memory control blocks:
Number: 1      Owner: MS DOS
Area size(b): 16

Number: 2      Owner: free
Area size(b): 64

Number: 3      Owner: 0040
Area size(b): 256

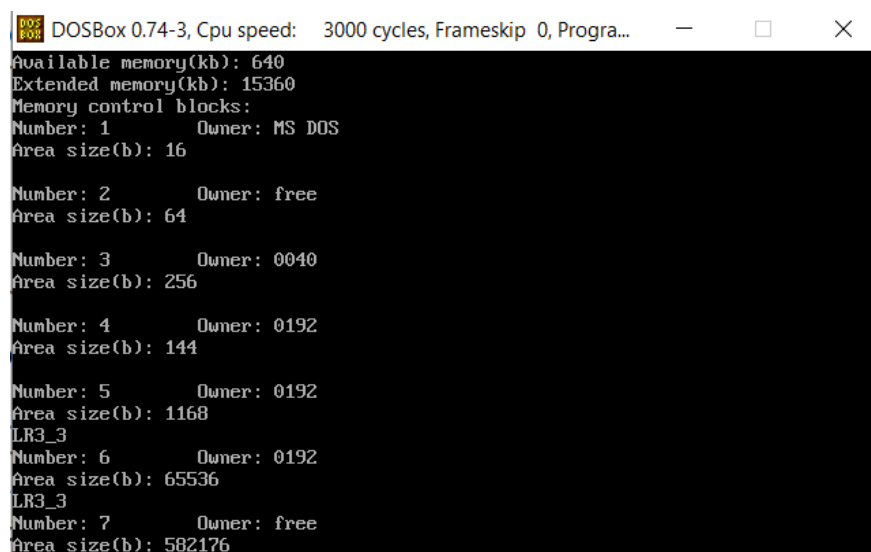
Number: 4      Owner: 0192
Area size(b): 144

Number: 5      Owner: 0192
Area size(b): 1120
LR3_2
Number: 6      Owner: free
Area size(b): 647776
ЧЕИЖ°€ИИ
```

Рисунок 2 - результат выполнения программы (LR3_2.com)

Исходя из результата выполнения, освобождённая память является 6 номером МСВ, который свободен.

Далее программа была изменена ещё раз таким образом, чтобы после освобождения памяти она ещё запрашивала 64Кб памяти. Результат выполнения этой программы представлен на рисунке 3.



```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
Available memory(kb): 640
Extended memory(kb): 15360
Memory control blocks:
Number: 1      Owner: MS DOS
Area size(b): 16

Number: 2      Owner: free
Area size(b): 64

Number: 3      Owner: 0040
Area size(b): 256

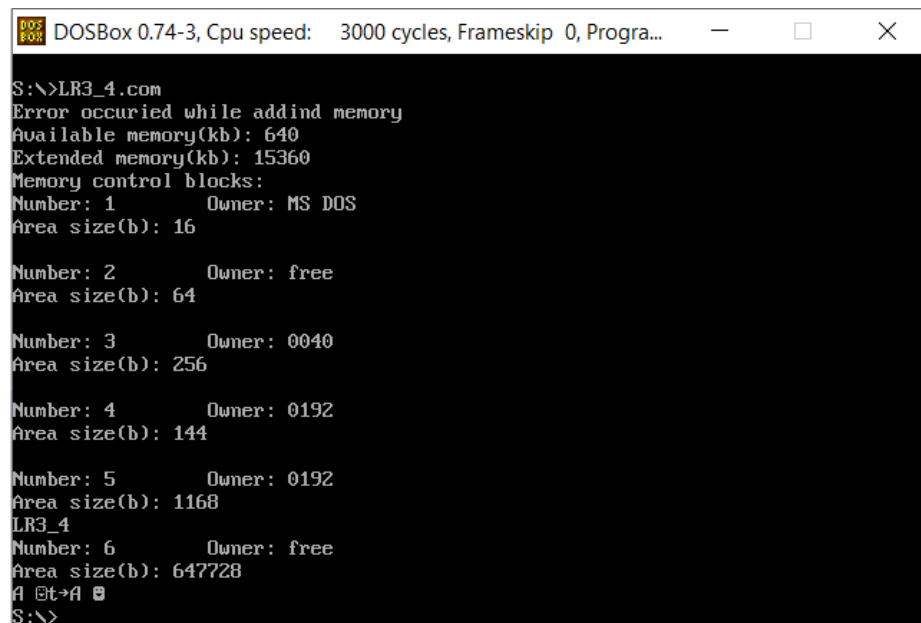
Number: 4      Owner: 0192
Area size(b): 144

Number: 5      Owner: 0192
Area size(b): 1168
LR3_3
Number: 6      Owner: 0192
Area size(b): 65536
LR3_3
Number: 7      Owner: free
Area size(b): 582176
```

Рисунок 3 - результат выполнения программы (LR3_3.com)

Далее была составлена ещё одна программа, которая сначала запрашивает 64Кб памяти, а потом освобождает память.

Результат выполнения этой программы представлен на рисунке 4.



```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
S:\>LR3_4.com
Error occurred while addind memory
Available memory(kb): 640
Extended memory(kb): 15360
Memory control blocks:
Number: 1      Owner: MS DOS
Area size(b): 16

Number: 2      Owner: free
Area size(b): 64

Number: 3      Owner: 0040
Area size(b): 256

Number: 4      Owner: 0192
Area size(b): 144

Number: 5      Owner: 0192
Area size(b): 1168
LR3_4
Number: 6      Owner: free
Area size(b): 647728
A @t>A
S:\>
```

Рисунок 4 - результат выполнения программы (LR3_4.com)

Исходя из результата выполнения программы, видно, что дополнительная память не была выделена.

Ответы на контрольные вопросы

1) Что означает «доступный объем» памяти?

Доступный объем памяти – это максимальный объем памяти, который может использовать программа.

2) Где МСВ блок Вашей программы в списке?

Во всех программах это 4-ый и 5-ый блоки МСВ, но в программе LR3_3 в дополнение к ним ещё относится 6-ой блок МСВ (в результате выделения дополнительных 64Кб)

3) Какой размер памяти занимает программа в каждом случае?

Для LR3_1 = $(648912 + 144)$ байт = 649056 байт

Для LR3_2 = $(1120 + 144)$ байт = 1264 байт после освобождения памяти

Для $LR3_3 = (1168 + 144) = 1312$ байт после освобождения памяти, и ещё 65536 байт после выделения памяти.

Для $LR3_4 = (1168 + 144) = 1312$ байт

Вывод.

В ходе выполнения данной лабораторной работы были изучены блоки МСВ, а также методы выделения и освобождения памяти для программы.