

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра математического обеспечения и применения ЭВМ

ОТЧЕТ
по практической работе №3
по дисциплине «Операционные системы»
Тема: Исследование организации управления основной памятью

Студентка гр. 7381

Кушкочева А.О.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2019

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.

Необходимые сведения для составления программы.

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.

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.

В результате выполнения лабораторной работы была написана программа, описание функций которой представлено в таблице ниже.

GET_AVAILABLE_MEMORY	Получение размера доступной памяти
GET_EXTENDED_MEMORY	Получение размера расширенной памяти
GET_MCB_DATA	Получение данных одного MSB блока
GET_ALL_MSB_DATA	Получение данных со всех MSB блоков
PRINT_STRING	Вывод строки на экран

Результат работы показан на рисунке ниже.

Pre.S. Размер участка (Size (Para)) указан в байтах.

1. LAB3V1.COM

```
C:\>lab3v1.com
Available memory: 648912 B
Extended memory : 15360 KB
Address | MCB Type | PSP Address | Size | SD/SC
016F    4D    0008        16
0171    4D    0000        64    DPMILOAD
0176    4D    0040       256
0187    4D    0192       144
0191    5A    0192   648912    LAB3V1
```

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

2. LAB3V2.COM

```
C:\>lab3v2.com
Available memory: 648912 B
Extended memory : 15360 KB
```

Address	MCB Type	PSP	Address	Size	SD/SC
016F	4D		0008	16	
0171	4D		0000	64	DPMILOAD
0176	4D		0040	256	
0187	4D		0192	144	
0191	4D		0192	13424	LAB3V2
04D9	5A		0000	635472	

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.

3. LAB3V3.COM

```
C:\>lab3v3.com
Available memory: 648912 B
Extended memory : 15360 KB
```

Address	MCB Type	PSP	Address	Size	SD/SC
016F	4D		0008	16	
0171	4D		0000	64	DPMILOAD
0176	4D		0040	256	
0187	4D		0192	144	
0191	4D		0192	13536	LAB3V3
04E0	4D		0192	65536	LAB3V3
14E1	5A		0000	569808	

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.

4. LAB3V4.COM

```
C:\>lab3v4.com
Available memory: 648912 B
ERROR
Extended memory : 15360 KB
Address | MCB Type | PSP Address | Size | SD/SC
016F    4D    0008        16
0171    4D    0000        64    DPMILOAD
0176    4D    0040       256
0187    4D    0192       144
0191    4D    0192     13840    LAB3V4
04F3    5A    0000    635056
```

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:

а) Что означает «доступный объём памяти»?

Это максимальный объем памяти, который может быть доступен программе.

В этом мы убеждаемся в четвёртом пункте данной л.р., когда после выделения всей доступной памяти, мы пытаемся выделить ещё, но, увы, – нельзя.

б) Где MCB блок Вашей программы в списке?

Блок нашей программы выделен красным на рисунке.

```
C:\>lab3v1.com
Availible memory: 648912 B
Extended memory : 15360 KB
```

Address	MCB Type	PSP	Address	Size	SD/SC
016F	4D	0008		16	
0171	4D	0000		64	DPMILOAD
0176	4D	0040		256	
0187	4D	0192		144	
0191	5A	0192		648912	LAB3V1

в) Какой размер памяти занимает программа в каждом случае?

LAB3V1.com) 648912 байт.

LAB3V2.com) 13328 байт.

LAB3V3.com) 13440 байт (без блока в 64кб).

LAB3V4.com) 14048 байт.