

**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. 9383

\_\_\_\_\_

Ноздрин В.Я.

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.

### **Задание.**

**Шаг 1.** Для выполнения лабораторной работы необходимо написать и отладить программный модуль типа .COM, который выбирает и распечатывает следующую информацию:

- 1) Количество доступной памяти.
- 2) Размер расширенной памяти.
- 3) Выводит цепочку блоков управления памятью.

Адреса при выводе представляются шестнадцатеричными числами. Объем памяти функциями управления памятью выводится в параграфах. Необходимо преобразовать его в байты и выводить в виде десятичных чисел. Последние восемь байт MSB выводятся как символы, не следует преобразовывать их в шестнадцатеричные числа.

Запустите программу и внимательно оцените результаты. Сохраните результаты, полученные программой, и включите их в отчет в виде скриншота.

**Шаг 2.** Измените программу таким образом, чтобы она освобождала память, которую она не занимает. Для этого используйте функция 4Ah прерывания 21h. Повторите эксперимент, запустив модифицированную программу. Сравните выходные данные с результатами, полученными на предыдущем шаге. Сохраните результаты, полученные программой и включите их в отчет в виде скриншота.

**Шаг 3.** Измените программу еще раз таким образом, чтобы после освобождения памяти, программа запрашивала 64Кб памяти функцией 48h прерывания 21h. Повторите эксперимент, запустив модифицированную программу. Сравните выходные данные с результатами, полученными на предыдущем шаге. Сохраните результаты, полученные программой и включите их в отчет в виде скриншота.

**Шаг 4.** Первоначальный вариант программы, запросив 64Кб памяти функцией 48h прерывания 21h до освобождения памяти. Обязательно обрабатывайте завершение функций ядра, проверяя флаг CF. Сохраните результаты, полученные программой и включите их в отчет в виде скриншота.

**Шаг 5.** Оцените результаты, полученные на предыдущих шагах.

#### **Основные теоретические положения.**

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.

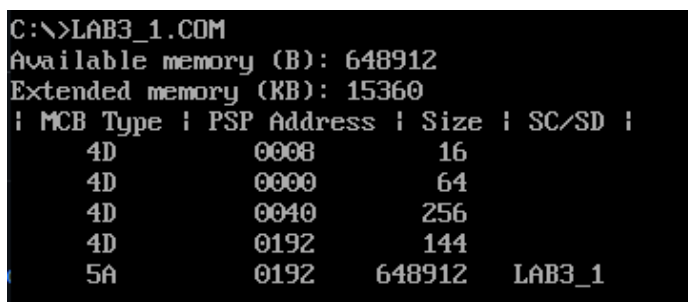
Адрес первого MCB находится во внутренней структуре MS DOS, называемой «List of Lists». Доступ к указателю на эту структуру можно получить используя функцию f52h «Get List of Lists» int 21h. В результате выполнения этой функции ES:BX будет указывать на список списков. Слово по адресу ES:[BX-2] и есть 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 размера расширенной
памяти
mov BL, AL
mov AL, 31h      ; запись адреса ячейки CMOS
out 70h, AL
in  AL, 71h      ; чтение старшего байта размера расширенной
памяти
```

### **Выполнение работы.**

**Шаг 1.** Написан текст исходного .COM модуля lab3\_1.asm, который определит количество доступной памяти и выводит его на экран, затем вычисляет количество расширенной памяти и выводит его на экран, а затем выводит на экран информацию о блоках MCB.



```
C:\>LAB3_1.COM
Available memory (B): 648912
Extended memory (KB): 15360
| MCB Type | PSP Address | Size | SC/SD |
4D        0008      16
4D        0000      64
4D        0040     256
4D        0192     144
5A        0192   648912   LAB3_1
```

Рисунок 1 — Вывод модуля LAB3\_1.COM

**Шаг 2.** lab3\_1.com изменен таким образом, чтобы программа освобождала память, которую она не занимает. Для этого использовалась функция 4Ah прерывания INT 21h.

```
C:\>LAB3_2.COM
Available memory (B): 648912
Extended memory (KB): 15360
| MCB Type | PSP Address | Size | SC/SD |
4D         0008         16
4D         0000         64
4D         0040        256
4D         0192        144
4D         0192       13184   LAB3_2
5A         0000      635712   w|i||e^_
```

Рисунок 2 — Вывод модуля LAB3\_2.COM

**Шаг 3.** lab3\_2.com изменен таким образом, чтобы программа освобождала память, которую она не занимает, а затем запрашивает 64Кб памяти. Для этого использовалась функция 48h прерывания INT 21h.

```
C:\>LAB3_3.COM
Available memory (B): 648912
Extended memory (KB): 15360
| MCB Type | PSP Address | Size | SC/SD |
4D         0008         16
4D         0000         64
4D         0040        256
4D         0192        144
4D         0192       13296   LAB3_3
4D         0192       65536   LAB3_3
5A         0000      570048
```

Рисунок 3 — Вывод модуля LAB3\_3.COM

**Шаг 4.** Исходный модуль lab3\_1.com изменен так, чтобы запрос 64Кб памяти происходил до освобождения памяти. В процессе выполнения программы на экран выводится сообщение об ошибке и очистке памяти, которую программа не занимает.

```
C:\>LAB3_4.COM
Available memory (B): 648912
Extended memory (KB): 15360
Memory allocation error
| MCB Type | PSP Address | Size | SC/SD |
4D         0008         16
4D         0000         64
4D         0040        256
4D         0192        144
4D         0192        164   LAB3_4
5A         0000      648832
```

Рисунок 4 — Вывод модуля LAB3\_4.COM

### **Используемые функции**

TETR\_TO\_HEX – функция, переводящая десятичную цифру в код символа.

BYTE\_TO\_HEX – функция, переводящая байт в шестнадцатеричной системе счисления в код символа.

WRD\_TO\_HEX – функция, переводящая шестнадцатеричное число в символьный код.

BYTE\_TO\_DEC – функция, переводящая байт в шестнадцатеричной системе счисления в символьный код десятичной системы счисления.

PRINT\_STRING – функция, выводящая строку на экран

GET\_AVAILABLE\_MEMORY – функция определяющая количество доступной памяти и выводящая результат на экран.

GET\_EXTENDED\_MEMORY – функция определяющая количество расширенной памяти и выводящая результат на экран.

GET\_MCB\_DATA – функция получает данные о каждом MCB блоке.

GET\_MCB\_ADDRESS – функция определяет адрес MCB блока.

GET\_MCB\_TYPE – функция определяет тип MCB блока.

GET\_MCB\_SIZE – функция определяет размер участка в параграфах.

GET\_PCP\_ADDRESS – определяет адрес PSP.

### **Контрольные вопросы.**

#### **Сегментный адрес недоступной памяти**

1) Что означает «доступный объем памяти»?

Доступный объем памяти это максимально возможный объем памяти, выделенный программе операционной системой.

2) Где MCB блок Вашей программы в списке?

MCB lab3\_1.com расположен последним в списке

MSB lab3\_2.com расположен предпоследним в списке. Последним идет блок, обозначенный как пустой участок.

MSB lab3\_3.com расположен третьим с конца в списке. После него идут выделенный по запросу и свободный блоки.

MSB lab3\_4.com расположен предпоследним в списке. Последним идет блок, обозначенный как пустой участок.

3) Какой размер памяти занимает программа в каждом случае?

Lab3\_1.com: 648912Bt

Lab3\_1.com: 648912Bt-635712Bt-16Bt=13184Bt

Lab3\_1.com: 648912Bt-570048Bt-65536Bt-2\*16Bt=13296Bt

Lab3\_1.com: 648912Bt-648832Bt-16Bt=64Bt

### **Выводы.**

В процессе выполнения лабораторной работы были изучены структуры данных и работа функций управления памятью ядра операционной системы.