

**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. 9381

Судаков Е.В.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

**Цель работы:** Исследование структур данных и работы функций управления памятью ядра операционной системы.

### Необходимые сведения для составления программы

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.

Структура MCB показана на рисунке 1.

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

Рисунок 1 – Структура MCB

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 52h "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, как показано на рисунке 2.

```
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
```

Рисунок 2 – Определение размера расширенной памяти

#### Ход 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.

- Далее необходимо изменить программу таким образом, чтобы она освобождала память, которую она не занимает. Для этого используйте функцию 4Ah прерывания 21h (пример в разделе «Использование функции 4AH»). Хвост командной строки в символьном виде.

- Затем необходимо изменить программу еще раз таким образом, чтобы после освобождения памяти, программа запрашивала 64Кб памяти функцией 48H прерывания 21H.

- Далее нужно изменить первоначальный вариант программы, запросив 64Kб памяти функцией 48H прерывания 21H до освобождения памяти. Оформить отчёт и ответить на контрольные вопросы.

#### Описание функций

Название	Назначение
TETR_TO_HEX	Перевод числа из 2-ой в 16-ую с/с (1/2 байта)
BYTE_TO_HEX	Перевод числа из 2-ой в 16-ую с/с (1 байт)
WRD_TO_HEX	Перевод числа из 2-ой в 16-ую с/с (2 байта)
BYTE_TO_DEC	Перевод числа из 2-ой в 10-ую с/с (1 байт)
WRD_TO_DEC	Перевод числа из 2-ой в 10-ую с/с (2 байта)
PRINT_SYMB	Вывод символа на экран
AVAILABLE_MEM ORY_SIZE	Вывод количества доступной памяти
EXTENDED_MEM ORY_SIZE	Вывод размера расширенной памяти
PRINT_LIST_MSB	Выводит цепочку блоков управления памятью.

Результат работы обычной версии программы представлен на рисунке 3.

```
F:\>lab3_1.com
Available memory: 648912b
Extended memory: 15360kb
List of MCB:
MCB type: 4Dh   PSP adress: 0008h   Size:      16 b
MCB type: 4Dh   PSP adress: 0000h   Size:      64 b
MCB type: 4Dh   PSP adress: 0040h   Size:     256 b
MCB type: 4Dh   PSP adress: 0192h   Size:     144 b
MCB type: 5Ah   PSP adress: 0192h   Size:   648912 b   LAB3_1
```

Рисунок 3 – Результат работы обычной версии программы

Результат работы программы после освобождения неиспользуемой памяти представлен на рисунке 4. Можно заметить, что размер блока, отведённого программе, уменьшился и образовался свободный участок

памяти.

```
F:\>lab3_2.com
Available memory: 648912b
Extended memory: 15360kb
List of MCB:
MCB type: 4Dh   PSP address: 0008h   Size:      16 b
MCB type: 4Dh   PSP address: 0000h   Size:      64 b
MCB type: 4Dh   PSP address: 0040h   Size:     256 b
MCB type: 4Dh   PSP address: 0192h   Size:     144 b
MCB type: 4Dh   PSP address: 0192h   Size:      800 b      LAB3_2
MCB type: 5Ah   PSP address: 0000h   Size: 648096 b      C> Zt9î
```

Рисунок 4 – Результат работы программы после освобождения  
неиспользуемой памяти

Результат работы программы после освобождения неиспользуемой памяти и запроса 64Кб показан на рисунке 5. Можно заметить, что размер участка свободной памяти уменьшился на 64Кб и образовался участок памяти в 64Кб, запрошенной программой.

```
F:\>lab3_3.com
Available memory: 648912b
Extended memory: 15360kb
List of MCB:
MCB type: 4Dh   PSP address: 0008h   Size:      16 b
MCB type: 4Dh   PSP address: 0000h   Size:      64 b
MCB type: 4Dh   PSP address: 0040h   Size:     256 b
MCB type: 4Dh   PSP address: 0192h   Size:     144 b
MCB type: 4Dh   PSP address: 0192h   Size:      848 b      LAB3_3
MCB type: 4Dh   PSP address: 0192h   Size: 65536 b      LAB3_3
MCB type: 5Ah   PSP address: 0000h   Size: 582496 b      crosoft
```

Рисунок 5 – Результат работы программы после освобождения  
неиспользуемой памяти и запроса 64Кб

Попытка запроса выделения 64Кб памяти до освобождения неиспользуемой памяти завершается неудачей. Результат работы программы

показан на рисунке 6.

```
F:\>lab3_4.com
Available memory: 648912b
Error! Memory can not be allocated!
Extended memory: 15360kb
List of MCB:
MCB type: 4Dh   PSP adress: 0008h   Size:      16 b
MCB type: 4Dh   PSP adress: 0000h   Size:      64 b
MCB type: 4Dh   PSP adress: 0040h   Size:     256 b
MCB type: 4Dh   PSP adress: 0192h   Size:     144 b
MCB type: 4Dh   PSP adress: 0192h   Size:     848 b   LAB3_4
MCB type: 5Ah   PSP adress: 0000h   Size:   648048 b   LAB3_3
```

Рисунок 6 – Результат работы программы при попытке выделить 64 Кб памяти до освобождения неиспользуемой памяти

### Ответы на контрольные вопросы:

#### 1. Что означает «доступный объем памяти»?

Доступный объем памяти – количество памяти, 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.

#### 2. Где MCB блок Вашей программы в списке?

В первой версии программы он занимает последние два пункта в конце списка: первый – окружение среды, второй принадлежит программе, которая была последней загружена в память и обладает всем объемом свободной ранее памяти.

Во второй версии, MCB блок так же четвертый и пятый в списке, но уже не последний. Последним является блок, освобожденной программой памяти.

В третьей версии, блок так же четвертый и пятый в списке, но после него располагаются блок, памяти в 64Кб, выделенный по запросу программы и после – блок свободной памяти.

В четвертой версии ситуация аналогична ситуации во второй версии, так как запрос выделения памяти завершился неудачей

3. Какой размер памяти занимает программа в каждом случае?

В первом случае программа занимает  $648912 \text{ байт} + 144 \text{ байт} = \mathbf{649056 \text{ байт}}$ .

Во втором –  $800 \text{ байт} + 144 \text{ байт} = \mathbf{944 \text{ байт}}$ .

В третьем –  $144 \text{ байт} + 848 \text{ байт} + 64 \text{ Кб}$ , выделенные ей по требованию  $= \mathbf{66528 \text{ байт}}$ .

В четвертом –  $848 \text{ байт} + 144 \text{ байт} = \mathbf{992 \text{ байта}}$ .

### **Вывод**

В результате выполнения данной лабораторной работы были исследованы структуры данных и работа функций управления памятью ядра операционной системы.