

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. 0381

Магнитов С.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

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.

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 МСВ 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 функцию 4Ah прерывания 21h (пример в разделе «Использование функции 4АН»). 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 предыдущих шагах. Сохраните результаты, полученные программой, и включите их в отчет в виде скриншота.

Шаг 4. Измените 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.

Теоретические сведения.

Таблица 1 – Структура MCB

<i>Смещение</i>	<i>Длина поля</i>	<i>Содержимое поля</i>
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

Выполнение работы.

В файле lab3.asm написан код исходного .COM модуля. В начале прописаны строки для вывода сообщений. Для их вывода была написана процедура PRINT, а также процедуры PRINT_SYM и PRINT_ENT для вывода символа и переноса строки. Помимо этого была написана процедура NEW_INTEGER, которое выводит число в десятичной системе счисления.

Была написана процедура PRINT_AVAIL_MEM для вывода количества доступной памяти в байтах с помощью строки SIZE_AVAILABLE_MEMORY.

Была написана процедура PRINT_MEM_SIZE для вывода размера расширенной памяти с помощью строки SIZE_EXTENDED_MEMORY.

Была написана процедура PRINT_MCB для вывода цепочки блоков управления памятью с помощью строки MEMORY_CONTROL_BLOCKS: выводятся порядковый номер, тип, адрес блока, сегментный адрес PSP, размер участка в байтах и системные данные. Это осуществляется благодаря функции 52h прерывания 21h, которая получает адрес самого первого блока и начинает выводить информацию до тех пор, пока тип блока не покажет, что он последний.

При выполнении программы показывается, что размер доступной памяти равен 64 байтам, потому что программа занимает весь доступный объем памяти. Размер расширенной памяти равняется 240 Кб. Результаты запуска модуля lab3.com представлены на рисунке 1.

```
F:\>lab3.com
Amount of available memory: 64 bytes
Extended memory size: 585728 bytes
Memory control blocks:
Type: 4Dh MCB adress: 016Fh PSP adress: 0008h MCB size: 16 SCSD:
Type: 4Dh MCB adress: 0171h PSP adress: 0000h MCB size: 64 SCSD:
Type: 4Dh MCB adress: 0176h PSP adress: 0040h MCB size: 256 SCSD:
Type: 4Dh MCB adress: 0187h PSP adress: 0192h MCB size: 144 SCSD:
Type: 5Ah MCB adress: 0191h PSP adress: 0192h MCB size: 648912 SCSD: LAB3
```

Рисунок 1 – Запуск модуля lab3.com

В файле lab32.asm был написан модифицированный код исходного .COM модуля, в котором освобождается неиспользованная память. Поэтому объем свободой памяти увеличивается, а также в выводе цепочки блоков управления появляется новый пункт со свободным блоком. Результаты запуска модуля lab32.com представлены на рисунке 2.

```
F:\>lab32.com
Amount of available memory: 583360 bytes
Extended memory size: 585728 bytes
Memory control blocks:
Type: 4Dh MCB address: 016Fh PSP address: 0000h MCB size: 16 SCSD:
Type: 4Dh MCB address: 0171h PSP address: 0000h MCB size: 64 SCSD:
Type: 4Dh MCB address: 0176h PSP address: 0040h MCB size: 256 SCSD:
Type: 4Dh MCB address: 0187h PSP address: 0192h MCB size: 144 SCSD:
Type: 4Dh MCB address: 0191h PSP address: 0192h MCB size: 65536 SCSD: LAB32
Type: 5Ah MCB address: 1192h PSP address: 0000h MCB size: 583360 SCSD:
```

Рисунок 2 – Запуск модуля lab32.com

В файле lab33.asm был изменен код таким образом, что теперь программа запрашивает 64 Кб памяти после её освобождения. Благодаря этому появился дополнительный блок, у которого объем свободной памяти остается прежним и равен сумме размеров двух последних блоков. Результаты запуска модуля lab33.com представлены на рисунке 3.

```
F:\>lab33.com
Amount of available memory: 517808 bytes
Extended memory size: 585728 bytes
Memory control blocks:
Type: 4Dh MCB address: 016Fh PSP address: 0000h MCB size: 16 SCSD:
Type: 4Dh MCB address: 0171h PSP address: 0000h MCB size: 64 SCSD:
Type: 4Dh MCB address: 0176h PSP address: 0040h MCB size: 256 SCSD:
Type: 4Dh MCB address: 0187h PSP address: 0192h MCB size: 144 SCSD:
Type: 4Dh MCB address: 0191h PSP address: 0192h MCB size: 65536 SCSD: LAB33
Type: 4Dh MCB address: 1192h PSP address: 0192h MCB size: 65536 SCSD: LAB33
Type: 5Ah MCB address: 2193h PSP address: 0000h MCB size: 517808 SCSD:
```

Рисунок 3 – Запуск модуля lab33.com

В файле lab34.asm функция 48h прерывания int 21h была вызвана до функции освобождения памяти. Она устанавливает флаг CF и не позволяет выделять память без предварительного её освобождения. В результате проверки выводится сообщение о невозможности данного действия. Результаты запуска модуля lab34.com представлены на рисунке 4.

```
F:\>lab34.com
All memory is distributed!
Amount of available memory: 64 bytes
Extended memory size: 585728 bytes
Memory control blocks:
Type: 4Dh MCB adress: 016Fh PSP adress: 0008h MCB size: 16 SCSD:
Type: 4Dh MCB adress: 0171h PSP adress: 0000h MCB size: 64 SCSD:
Type: 4Dh MCB adress: 0176h PSP adress: 0040h MCB size: 256 SCSD:
Type: 4Dh MCB adress: 0187h PSP adress: 0192h MCB size: 144 SCSD:
Type: 5Ah MCB adress: 0191h PSP adress: 0192h MCB size: 648912 SCSD: LAB34
```

Рисунок 4 – Запуск модуля lab34.com

Вывод.

Во время выполнения лабораторной работы была исследована работа функций управления памятью ядра операционной системы.

Ответы на контрольные вопросы.

1. Что означает «доступный объем памяти»?

Доступный объем памяти – свободный участок памяти наибольшего размера, который может запросить программа.

2. Где MCB блок Вашей программы в списке?

В первом случае – 5 блок (последний). Во втором случае – 5 блок (предпоследний). В третьем случае – 5 и 6 блок. В четвертом случае – 5 блок (последний).

3. Какой размер памяти занимает программа в каждом случае?

В первом случае – 648912 байт (всё возможное пространство). Во втором случае – 65536 байт. В третьем случае – 131072 байт. В четвертом случае – 648912.