

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. 0381

Михайлов В.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.

Постановка задачи.

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.

Шаг 5. Оцените результаты, полученные на предыдущих шагах. Ответьте на контрольные вопросы и оформите отчет.

Исходные данные.

За основу был взят предоставленный шаблон, содержащий процедуры: TETR_TO_HEX, BYTE_TO_HEX_WRD_TO_HEX, BYTE_TO_DEC.

Таблица 1 – Структура MCB

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 исключительной 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 нём системный код “SD” – если участок принадлежит MS DOS, то в нём системные данные

Выполнение работы.

1. В начале объявлены строки, необходимые для вывода информации согласно заданию. Далее реализованы следующие процедуры:

- FREE_MEM_INFO. Выводит размер доступной памяти в байтах.
- EXTENDED_MEM_INFO. Выводит размер расширенной памяти в байтах.
- MCB_INFO. Выводит тип, адрес, сегмент владельца и размер участка каждого MCB-блока.
- WRD_TO_DEC. Переводит размер памяти в параграфах в байты и добавляет их десятичное представление в строку.

```
F:\>lr3_1.com
Size of free memory:      64 byte
Size of extended memory: 15728640 byte
MCB: 4Dh Address: 016Fh Owner: MS DOS Area size: 16 byte
SC/SD:
MCB: 4Dh Address: 0171h Owner: Free Area size: 64 byte
SC/SD:
MCB: 4Dh Address: 0176h Owner: 0040h Area size: 256 byte
SC/SD:
MCB: 4Dh Address: 0187h Owner: 0192h Area size: 144 byte
SC/SD:
MCB: 5Ah Address: 0191h Owner: 0192h Area size: 648912 byte
SC/SD: LR3_1
```

Рис. 1 – Вывод. Шаг 1.

2. Программа модифицируется таким образом, чтобы освобождать неиспользуемую память, поэтому объём свободный увеличивается:

```
F:\>lr3_2.com
Size of free memory:      583360 byte
Size of extended memory: 15728640 byte
MCB: 4Dh Address: 016Fh Owner: MS DOS Area size: 16 byte
SC/SD:
MCB: 4Dh Address: 0171h Owner: Free Area size: 64 byte
SC/SD:
MCB: 4Dh Address: 0176h Owner: 0040h Area size: 256 byte
SC/SD:
MCB: 4Dh Address: 0187h Owner: 0192h Area size: 144 byte
SC/SD:
MCB: 4Dh Address: 0191h Owner: 0192h Area size: 65536 byte
SC/SD: LR3_2
MCB: 5Ah Address: 1192h Owner: Free Area size: 583360 byte
SC/SD: ength
```

Рис. 2 – Вывод. Шаг 2.

3. Теперь программа после высвобождения памяти будет запрашивать ещё 64 Кб. Видно, что в списке добавился ещё один блок:

```
F:\>lr3_3.com

Size of free memory:      517808 byte
Size of extended memory:  15728640 byte
MCB: 4Dh  Address: 016Fh  Owner: MS DOS  Area size: 16 byte
SC/SD:
MCB: 4Dh  Address: 0171h  Owner: Free    Area size: 64 byte
SC/SD:
MCB: 4Dh  Address: 0176h  Owner: 0040h   Area size: 256 byte
SC/SD:
MCB: 4Dh  Address: 0187h  Owner: 0192h   Area size: 144 byte
SC/SD:
MCB: 4Dh  Address: 0191h  Owner: 0192h   Area size: 65536 byte
SC/SD: LR3_3
MCB: 4Dh  Address: 1192h  Owner: 0192h   Area size: 65536 byte
SC/SD: LR3_3
MCB: 5Ah  Address: 2193h  Owner: Free    Area size: 517808 byte
SC/SD:
```

Рис 3. – Вывод. Шаг 3.

4. В последней версии программа пробует получить дополнительную память, перед этим её не освободив. Функция 48h устанавливает флаг CF, который говорит о том, что невозможно выделить память. Об этом выводится специальное сообщение, и дальше выводимая информация совпадает с первым пунктом:

```
F:\>lr3_4.com
unable to allocate memory
Size of free memory:      64 byte
Size of extended memory:  15728640 byte
MCB: 4Dh  Address: 016Fh  Owner: MS DOS  Area size: 16 byte
SC/SD:
MCB: 4Dh  Address: 0171h  Owner: Free    Area size: 64 byte
SC/SD:
MCB: 4Dh  Address: 0176h  Owner: 0040h   Area size: 256 byte
SC/SD:
MCB: 4Dh  Address: 0187h  Owner: 0192h   Area size: 144 byte
SC/SD:
MCB: 5Ah  Address: 0191h  Owner: 0192h   Area size: 648912 byte
SC/SD: LR3_4
```

Рис 4. – Вывод. Шаг 4.

Вывод.

Была исследована структура данных и работа функций управления памятью ядра операционной системы.

Контрольные вопросы.

1) Что означает «доступный объём памяти»?

Доступный объём памяти — это свободный участок памяти максимального размера, который может запросить программа.

2) Где МСВ блок Вашей программы в списке?

На первом шаге это последний 5-й блок.

На втором — предпоследний 5-й блок.

На третьем — 5-й и 6-й блоки.

На четвёртом — последний 5-й блок.

3) Какой размер памяти занимает программа в каждом случае?

В первом случае программа занимает всё доступное пространство — 648912 байт.

Во втором — 65536 байт.

В третьем — 131072 байт (два блока по 65536 байт).

В четвёртом, как в первом.