

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

Захаров Ф.С.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы

Исследовать структуры данных и работу функций управления памятью ядра операционной системы.

Порядок выполнения работы

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 функцию 4Ah прерывания 21h (пример в разделе «Использование функции 4AH»). 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.

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:

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 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

```

Выполнение работы.

1. Был написан и отлажен .com модуль, который выбирает и печатает следующую информацию:
 - Количество доступной памяти.
 - Размер расширенной памяти.
 - Цепочка блоков управления памятью.

```
F:\>LB3_1.COM
Available memory:
648912
Extended memory:
245760
MCB type is: 016F PSP adress is: 0008 Size is: 16 SC/SD:
MCB type is: 0171 PSP adress is: 0000 Size is: 64 SC/SD:
MCB type is: 0176 PSP adress is: 0040 Size is: 256 SC/SD:
MCB type is: 0187 PSP adress is: 0192 Size is: 144 SC/SD:
MCB type is: 0191 PSP adress is: 0192 Size is: 648912 SC/SD: LB3_1
```

Рисунок 1- Вариант программы для шага 1

2. Модифицированная программа освобождает память, которую не использует и теперь выводит такие данные:

```
F:\>LB3_2.COM
Available memory:
648912
Extended memory:
245760
MCB type is: 016F PSP adress is: 0008 Size is: 16 SC/SD:
MCB type is: 0171 PSP adress is: 0000 Size is: 64 SC/SD:
MCB type is: 0176 PSP adress is: 0040 Size is: 256 SC/SD:
MCB type is: 0187 PSP adress is: 0192 Size is: 144 SC/SD:
MCB type is: 0191 PSP adress is: 0192 Size is: 6432 SC/SD: LB3_2
MCB type is: 0324 PSP adress is: 0000 Size is: 642464 SC/SD: .ïþ
.Ï▲
```

Рисунок 2 - Вариант программы для шага 2

3. После того, как неиспользуемая память освобождается, программа теперь запрашивает 64Кб памяти.

```

F:\>LB3_3.COM
Available memory:
648912
Extended memory:
245760
MCB type is: 016F PSP adress is: 0008 Size is: 16 SC/SD:
MCB type is: 0171 PSP adress is: 0000 Size is: 64 SC/SD:
MCB type is: 0176 PSP adress is: 0040 Size is: 256 SC/SD:
MCB type is: 0187 PSP adress is: 0192 Size is: 144 SC/SD:
MCB type is: 0191 PSP adress is: 0192 Size is: 6432 SC/SD: LB3_3
MCB type is: 0324 PSP adress is: 0192 Size is: 65536 SC/SD: LB3_3
MCB type is: 1325 PSP adress is: 0000 Size is: 576912 SC/SD:

```

Рисунок 3 - Вариант программы для шага 3

4. 64Кб теперь запрашиваются до освобождения памяти.

```

F:\>LB3_4.COM
Available memory:
648912
Extended memory:
245760
CAN NOT REQUEST MEMORY
MCB type is: 016F PSP adress is: 0008 Size is: 16 SC/SD:
MCB type is: 0171 PSP adress is: 0000 Size is: 64 SC/SD:
MCB type is: 0176 PSP adress is: 0040 Size is: 256 SC/SD:
MCB type is: 0187 PSP adress is: 0192 Size is: 144 SC/SD:
MCB type is: 0191 PSP adress is: 0192 Size is: 6432 SC/SD: LB3_4
MCB type is: 0324 PSP adress is: 0000 Size is: 642464 SC/SD: .i6p
.Ä▲

```

Рисунок 4 - Вариант программы для шага 4

Вывод: были исследованы структуры данных и работа функций управления памятью ядра операционной системы. Была реализована программа, которая выводит информацию об использовании памяти в системе с помощью системных функций и обращений к МСВ.

Контрольные вопросы.

1. Что означает «доступный объем памяти»?

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

2. Где МСВ блок вашей программы в списке?

1 шаг: 5 строчка

2 шаг: 5 строчка

3 шаг: 5, 6 строчка

4 шаг: 5 строчка

3. Какой размер памяти занимает программа в каждом случае?

1 шаг: 648912 байт (всю возможную)

2 шаг: 6432 байт (сколько требуется)

3 шаг: 71968 (сколько требуется + 64Кб)

4 шаг: 6432 байт (сколько требуется)