

**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 памяти.**

Студент гр. 7381

\_\_\_\_\_

Лукашев Р.С.

Преподаватель

\_\_\_\_\_

Ефремов М.А.

Санкт-Петербург

2019

## **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.

## **Постановка задачи.**

Шаг 1. Необходимо написать и отладить программный модуль типа .COM, 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.

Шаг 2. Далее необходимо изменить программу таким образом, чтобы она освобождала память, которую она не занимает. Для этого необходимо использовать функцию 4Ah прерывания 21h (пример в разделе «Использование функции 4AH»).

Шаг 3. Необходимо изменить программу еще раз таким образом, чтобы после освобождения памяти, программа запрашивала 64Кб памяти функцией 48H прерывания 21H.

Шаг 4. Далее нужно изменить первоначальный вариант программы, запросив 64Кб памяти функцией 48H прерывания 21H до освобождения памяти.

### Ход работы.

1) Результат работы первой версии программы представлен на рисунке 1.

```
C:\>LAB3_1.COM
Accesible memory size -      648912 B
Extended memory size -      15360 KB
MCB table:
Address!   Type!   Owner!     Size!  Name
  016F!     4D!     0008!       16!
  0171!     4D!     0000!       64! DPMILOAD
  0176!     4D!     0040!      256!
  0187!     4D!     0192!      144!
  0191!     5A!     0192!  648912! LAB3_1
```

Рисунок 1 – результат работы программы

Все доступные 648912 байт отдаются программе.

2) Результат работы второй версии программы представлен на рисунке 2.

```
C:\>LAB3_2.COM
Accesible memory size -      648912 B
Extended memory size -      15360 KB
MCB table:
Address!   Type!   Owner!     Size!  Name
  016F!     4D!     0008!       16!
  0171!     4D!     0000!       64! DPMILOAD
  0176!     4D!     0040!      256!
  0187!     4D!     0192!      144!
  0191!     4D!     0192!   13136! LAB3_2
  04C7!     5A!     0000!  635760!
```

Рисунок 2 – результат работы программы

Исходный код программы был изменен. Теперь, после освобождения неиспользуемой памяти программа занимает 13136 байт, а также появился свободный блок размером 635760 байт.

3) Результат работы третьей версии программы представлен на рисунке 3.

```
C:\>LAB3_3.COM
Accesible memory size - 648912 B
Extended memory size - 15360 KB
MCB table:
Address:  Type:  Owner:  Size:  Name
016F:    4D:    0008:    16:
0171:    4D:    0000:    64:
0176:    4D:    0040:   256:
0187:    4D:    0192:   144:
0191:    4D:    0192:  14544: LAB3_3
051F:    4D:    0192:  65536: LAB3_3
1520:    5A:    0000: 568800:
```

Рисунок 3 – результат работы программы

Исходный код программы был изменен. Теперь, после освобождения, в блоке свободной памяти происходит выделение дополнительно 64 КБ (65536 Б) памяти, следующих за основным блоком программы.

4) Результат работы четвертой версии программы представлен на рисунке 4.

```
C:\>LAB3_4.COM
Accesible memory size - 648912 B
Error occured during memory request. Error code: 0008
Extended memory size - 15360 KB
MCB table:
Address:  Type:  Owner:  Size:  Name
016F:    4D:    0008:    16:
0171:    4D:    0000:    64: DPMILOAD
0176:    4D:    0040:   256:
0187:    4D:    0192:   144:
0191:    5A:    0192: 648912: LAB3_4
```

Рисунок 4 – результат работы программы

Исходный код первоначальной программы был изменен. Теперь происходит запрос на выделение дополнительных 64КБ памяти до освобождения неиспользуемой памяти, что приводит к ошибке. Код ошибки 8 означает, что было недостаточно памяти. Очевидно, откуда возникла ошибка, - вся доступная память уже и так отведена программе.

### Ответы на контрольные вопросы.

1) Что означает “доступный объем памяти”?

Доступный объём памяти – это объем памяти, которую операционная система выделила для программы. В нашем случае это объём базовой или

стандартной памяти (conventional memory). Эта память представляет собой "нижние" 640 Кбайт ОЗУ. Для использования базовой памяти не нужны никакие дополнительные драйверы, поскольку операционная система MS DOS изначально создана для работы в адресах 0 - 640 Кбайт

2) Где МСВ блок вашей программы в списке?

Блок с адресом 0192h

3) Какой размер памяти занимает программа в каждом случае?

1. 648912 Б – весь объем доступной памяти;

2. 13136 Б – размер используемой программой памяти;

3.  $14544 + 65536 = 80080$  Б – размер используемой программой памяти + выделенный блок в 64 КБ;

4. 648912 Б – весь объем доступной памяти.

### **Заключение.**

В ходе работы было проведено исследование структуры данных и работа функций управления памятью ядра операционной системы, а также рассмотрены нестраничная память и способ управления динамическими разделами.