

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра математического обеспечения и применения ЭВМ

ОТЧЕТ
по лабораторной работе №3
по дисциплине «Операционные системы»
Тема: Исследование организации управления памятью

Студент гр. 8382

Щеглов А.С.

Преподаватель

Ефремов. М.А.

Санкт-Петербург

2020

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.

Выполнение работы.

В ходе работы был написан и отлажен программный модуль типа .com, который выбирает и распечатывает следующую информацию:

- Количество доступной памяти
- Размер расширенной памяти
- Выводит цепочку блоков управления памятью

Количество доступной памяти получено с использованием функции 4Ah с заведомо большей памятью в регистре BX, чем может быть предоставлено ОС.

Размер расширенной памяти был получен при помощи обращения к ячейкам CMOS по адресам 30h, 31h.

Для получения адреса первого MCB программа обращается к внутренней структуре MS DOS, называемой список списков. Доступ к указателю на эту структуру возможно получить используя функцию 52h. В результате выполнения этой функции EX:BS будет указывать на список списков. Слово по адресу ES:[BX-2] будет адресом самого первого MCB.

```

C:\>lr3_1.com
Available memory - 648912 B.
Extended memory size - 15360 B.
MCB Type - 4D Sector - MS DOS Size -      16 Bytes. Last 8 bytes -
MCB Type - 4D Sector - IS FREE  Size -      64 Bytes. Last 8 bytes -
MCB Type - 4D Sector - 0040 Size -      256 Bytes. Last 8 bytes -
MCB Type - 4D Sector - 0192 Size -      144 Bytes. Last 8 bytes -
MCB Type - 5A Sector - 0192 Size - 648912 Bytes. Last 8 bytes - LR3_1
C:\>

```

Рисунок 1. Результат выполнения программы lr3_1.com

Далее программы надо было модифицировать так, что она высвобождала память, которую не использует. Сделать это можно было с помощью функции 4Ah.

```

C:\>lr3_2.com
Available memory - 648912 B.
Extended memory size - 15360 B.
MCB Type - 4D Sector - MS DOS Size -      16 Bytes. Last 8 bytes -
MCB Type - 4D Sector - IS FREE  Size -      64 Bytes. Last 8 bytes -
MCB Type - 4D Sector - 0040 Size -      256 Bytes. Last 8 bytes -
MCB Type - 4D Sector - 0192 Size -      144 Bytes. Last 8 bytes -
MCB Type - 4D Sector - 0192 Size - 13552 Bytes. Last 8 bytes - LR3_2
MCB Type - 5A Sector - IS FREE  Size - 635344 Bytes. Last 8 bytes -
C:\>

```

Рисунок 2. Результат выполнения программы lr3_2.com

Затем программу надо было еще раз модифицировать так, чтобы после высвобождения неиспользуемой памяти ей выделялось добавочные 64Кб памяти, с помощью функции 48h.

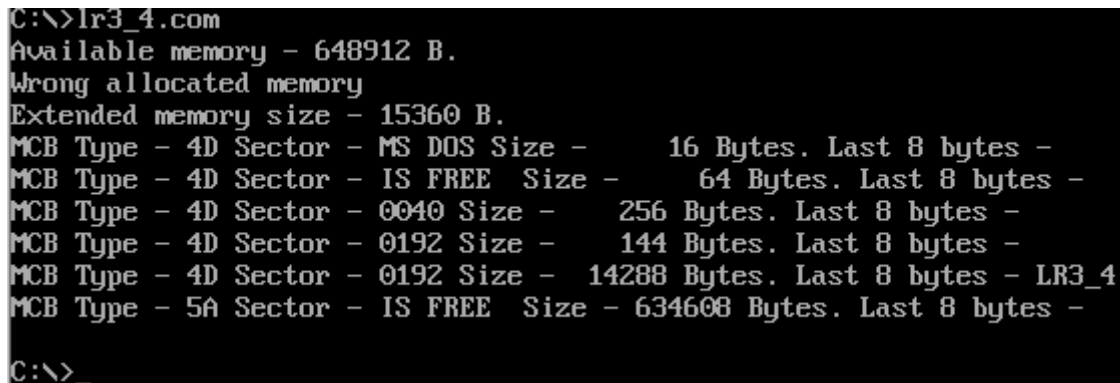
```

C:\>lr3_3.com
Available memory - 648912 B.
Extended memory size - 15360 B.
MCB Type - 4D Sector - MS DOS Size -      16 Bytes. Last 8 bytes -
MCB Type - 4D Sector - IS FREE  Size -      64 Bytes. Last 8 bytes -
MCB Type - 4D Sector - 0040 Size -      256 Bytes. Last 8 bytes -
MCB Type - 4D Sector - 0192 Size -      144 Bytes. Last 8 bytes -
MCB Type - 4D Sector - 0192 Size - 13664 Bytes. Last 8 bytes - LR3_3
MCB Type - 4D Sector - 0192 Size - 65536 Bytes. Last 8 bytes - LR3_3
MCB Type - 5A Sector - IS FREE  Size - 569680 Bytes. Last 8 bytes -
C:\>

```

Рисунок 3. Результат выполнения программы lr3_3.com

И наконец в последней модификации нужно было изменить порядок высвобождения и добавления памяти с отслеживанием регистрового флага CF. В результате выполнения такой программы была обнаружена ошибка аллокации памяти.



```
C:\>lr3_4.com
Available memory - 648912 B.
Wrong allocated memory
Extended memory size - 15360 B.
MCB Type - 4D Sector - MS DOS Size -      16 Bytes. Last 8 bytes -
MCB Type - 4D Sector - IS FREE  Size -      64 Bytes. Last 8 bytes -
MCB Type - 4D Sector - 0040 Size -     256 Bytes. Last 8 bytes -
MCB Type - 4D Sector - 0192 Size -     144 Bytes. Last 8 bytes -
MCB Type - 4D Sector - 0192 Size -    14288 Bytes. Last 8 bytes - LR3_4
MCB Type - 5A Sector - IS FREE  Size -   634608 Bytes. Last 8 bytes -
C:\>
```

Рисунок 4. Результат выполнения программы lr3_4.com

Ответы на контрольные вопросы.

1) Что означает «доступный объём памяти»?

Доступный объём памяти – 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 MCB блок Вашей программы в списке?

На рисунках 1, 2, 4 - это блоки под номерами 4 и 5.

На рисунке 3 - это блоки под номерами 4, 5, 6.

3) Какой размер памяти занимает программа в каждом случае?

1. $648912 + 144 = 649056$ байт.
2. $144 + 13552 = 13696$ байт.
3. $144 + 13664 + 65536 = 79344$ байт.
4. $144 + 14288 = 14432$ байт.

Выводы.

В ходе лабораторной работы были исследованы структуры данных и работы функций управления памятью ядра операционной системы.