

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 основной памятью

Студент гр. 8382

Преподаватель

Гордиенко А.М.

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.

Результат выполнения программы указан на рисунке 1.

```
C:\>lr3-1.com
Available memory - 648912 B.
Extended memory - 15360 B.
MCB Type - 4D Sector - MS DOS Size - 16 Bytes. Last 8 bytes -
MCB Type - 4D Sector - 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
```

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. Результат выполнения указан на рисунке 2.

```
C:\>lr3-2.com
Available memory - 648912 B.
Extended memory - 15360 B.
MCB Type - 4D Sector - MS DOS Size -      16 Bytes. Last 8 bytes -
MCB Type - 4D Sector - 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 -    13024 Bytes. Last 8 bytes - LR3-2
MCB Type - 5A Sector - FREE Size -    635872 Bytes. Last 8 bytes - in an o
```

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. Результат выполнения программы указан на рисунке 3.

```
C:\>lr3-3.com
Available memory - 648912 B.
Extended memory - 15360 B.
MCB Type - 4D Sector - MS DOS Size -      16 Bytes. Last 8 bytes -
MCB Type - 4D Sector - 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 -    13136 Bytes. Last 8 bytes - LR3-3
MCB Type - 4D Sector - 0192 Size -    65536 Bytes. Last 8 bytes - LR3-3
MCB Type - 5A Sector - FREE Size -    570208 Bytes. Last 8 bytes -
```

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. Результат выполнения указан на рисунке 4.

```
C:\>lr3-4.com
Available memory - 648912 B.
Wrong allocated memory
Extended memory - 15360 B.
MCB Type - 4D Sector - MS DOS Size -      16 Bytes. Last 8 bytes -
MCB Type - 4D Sector - 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 -    13728 Bytes. Last 8 bytes - LR3-4
MCB Type - 5A Sector - FREE Size -    635168 Bytes. Last 8 bytes - r40.L3C
```

Рисунок 4. Результат выполнения программы lr3-4.com

Контрольные вопросы.

1. Что означает “доступный объем памяти”?

Некоторое количество оперативной памяти, которая выделяется программе и к которой программа может обратиться.

2. Где MSB блок Вашей программы в списке?

По теории мы знаем, что блоки с адресом причастности 0192h являются блоками MSB.

В первой программе занимает 4 и 5 блоки.

Во второй - также 4 и 5.

В третьей, это 4,5 и 6 блоки.

В четвертой - 4 и 5 блоки.

3. Какой размер памяти занимает программа в каждом случае?

В первой программе это 649056 байт.

Во второй - 13168 байт.

В третьей - 78672 байта.

В четвертой программе была ошибка с определением памяти, и если программой все посчитано верно, то ей выделено 13872 байта.

Выводы.

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