

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

Студентка гр. 7383

Тян Е.

Преподаватель

Ефремов М. А.

Санкт-Петербург

2019

Постановка задачи.

Исследовать организацию управления памятью, ориентируясь на тип основной памяти, реализованный в компьютере и способ организации, принятый в ОС. Рассмотреть нестраничную 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.

Исследовать структуры данных и работу функций управления памятью.

В данной работе были использованы процедуры:

- Write_msg – выводит сообщение на экран
- TERT_TO_HEX – переводит из двоичной в шестнадцатеричную систему счисления
- BYTE_TO_HEX – переводит байтовое число в шестнадцатеричную систему счисления
- WRD_TO_HEX – переводит шестнадцатибитовое число в шестнадцатеричную систему счисления;
- WRD_TO_DEC – переводит шестнадцатибитовое число в десятичную систему счисления;
- DEFINE_AVAIL_MEM – определяет и выводит количество доступной памяти;
- DEFINE_WID_MEM – определяет и выводит размер расширенной памяти;
- DEFINE_TAIL – определяет окончание блока памяти;
- DEFINE_BLOCK_CHAIN – выводит цепочку блоков управления памятью.

В данной программе использовались следующие структуры данных:

- AVAIL_MEM – строка, хранящая размер доступной памяти;
- WID_MEM – строка, хранящая размер расширенной памяти;
- BLOCK_CHAIN – строка, хранящая строку «MCBs chain»;
- MCB_OWN – строка, хранящая
- MCB_ADD – строка, хранящая сегментный адрес блока памяти;
- MCB_SIZE – строка, хранящая размер участка;
- MCB_TAIL – строка, хранящая восьмибайтное окончание блока памяти;

- ERR_OR – строка, информирующая об ошибке, которая возникла при выделении памяти.

Программа определяет количество доступной памяти, размер расширенной памяти и выводит цепочку блоков управления памятью.

Ход работы.

1. Был написан текст исходного .COM модуля, для четырех программ:

first.asm: программа определяет количество доступной памяти, размер расширенной памяти и выводит цепочку блоков управления памятью.

deuxieme.asm: программа определяет количество доступной памяти, освобождает память, которую она не занимает, определяет размер расширенной памяти и выводит цепочку блоков управления памятью.

third.asm: программа определяет количество доступной памяти, освобождает память, которую она не занимает, запрашивает 64Кб

```
Number of available memory: 648912B
Quantity of widened memory: 15360KB
MCBs chain:
Address:016F      Owner: 0008      Size:      16      Tail:
Address:0171      Owner: 0000      Size:      64      Tail:
Address:0176      Owner: 0040      Size:     256      Tail:
Address:0187      Owner: 0192      Size:     144      Tail:
Address:0191      Owner: 0192      Size: 648912      Tail:FIRST
```

Рисунок 1 — Результат работы программы first.asm

```
Number of available memory: 648912B
Quantity of widened memory: 15360KB
MCBs chain:
Address:016F      Owner: 0008      Size:      16      Tail:
Address:0171      Owner: 0000      Size:      64      Tail:
Address:0176      Owner: 0040      Size:     256      Tail:
Address:0187      Owner: 0192      Size:     144      Tail:
Address:0191      Owner: 0192      Size:     832      Tail:DEUXIEME
Address:01C6      Owner: 0000      Size: 648064      Tail: [?] t
```

Рисунок 2 — Результат работы программы deuxieme.asm

```
Number of available memory: 648912B
Quantity of widened memory: 15360KB
MCBs chain:
Address:016F      Owner: 0008      Size:      16      Tail:
Address:0171      Owner: 0000      Size:      64      Tail:
Address:0176      Owner: 0040      Size:     256      Tail:
Address:0187      Owner: 0192      Size:     144      Tail:
Address:0191      Owner: 0192      Size:     832      Tail:THIRD
Address:01C6      Owner: 0192      Size: 264544      Tail:THIRD
Address:425D      Owner: 0000      Size: 383504      Tail:
```

Рисунок 3 — Результат работы программы third.asm

памяти функцией 4Ah прерывания 21h, определяет размер расширенной памяти и выводит цепочку блоков управления памятью.

```
Number of available memory: 648912B  
Error!
```

Рисунок 4 — Результат работы программы fourth.asm

fourth.asm: программа определяет количество доступной памяти, запрашивает 64Кб памяти функцией 4Ah прерывания 21h. Проверяется флаг CF, если флаг информирует об ошибке, возникшей при выделении памяти, то выводи сообщение об ошибке, если все проходит благополучно, то определяет размер расширенной памяти и выводит цепочку блоков управления памятью.

2. Результаты работы программы приведены на рис. 1, рис. 2, рис. 3, рис. 4.

Ответы на контрольные вопросы.

1) Что означает «доступный объем памяти»?

Ответ: т. к. MS-DOS – однопрограммная ОС, поэтому под доступным объемом памяти понимают тот объем памяти, который предоставлен программе после ее запуска. Всю остальную память занимает ядро операционной системы.

2) Где MCB блок Вашей программы в списке?

Ответ: каждой программе принадлежат следующие MCB:

- first.asm: 4 и 5 строчки в выводе MCB блоков программой;
- deuxieme.asm: 4 и 5 строчки в выводе MCB блоков программой;
- third.asm: 4, 5 и 6 строчки в выводе MCB блоков программой.

4-ый блок MCB для каждой программы владеет блоком памяти, содержащим область среды программы.

5-ый блок MCB для каждой программы владеет блоком памяти, выделенной программе.

6-ой блок MCB для каждой программы владеет блоком памяти, выделенным программе в 64Кб.

3) Какой размер памяти занимает программа в каждом случае?

Ответ: для каждой программы:

- first.asm: 648 912 байт(весь доступный объем памяти, память не высвобождалась и не выделялась);
- deuxieme.asm: 832 байт(освободили ненужную память);
- third.asm: $832 + 264\,544 = 265\,376$ байт(освободили ненужную память + выделили 64 КБ).

Выводы.

В данной работе была исследована организация управления памятью, ориентируясь на тип основной памяти, реализованный в компьютере и способ организации, принятый в ОС. Была рассмотрена нестраничная память и способ управления динамическими разделами.