

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

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

Студент гр

Диденко Д.В

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

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 не страничная 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, предусматривают и преобразуют этот список.

В лабораторной работе исследуются структуры данных и работа функций управления памятью ядра операционной системы

Постановка задачи.

Требуется написать и отладить программный модуль типа .COM, который выбирает и распечатывает следующую информацию:

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

Выполнение работы.

Были написаны строки для вывода информации:

- `ACCESSED_MEMORY db 13,10,'Size of accessed memory: $';`
- `EXTENDED_MEMORY db 13,10,'Size of extended memory: $';`
- `STR_BYTE db ' byte $';`
- `STR_MCB db 13,10,'MCB:0 $';`
- `ADRESS db 'Adress: $';`
- `ADRESS_PSP db 'PSP adress: $';`
- `STR_SIZE db 'Size: $';`
- `MCB_SD_SC db ' SD/SC: $' ;`

- STR_ERROR db 13,10,'Memory Error!\$';
- STR_SUCCECC db 13,10,'Success!\$'.

Были составлены функции для считывания данных из MCB, выделения памяти и преобразования чисел (см. табл.1)

Таблица 1 – функции в программе

Процедура	Описание
TETR_TO_HEX	Перевод десятичной цифры в код символа
BYTE_TO_HEX	Перевод байта в 16-ной с/с в символьный код
WRD_TO_HEX	Перевод слова в 16-ной с/с в символьный код
BYTE_TO_DEC	Перевод байта в 16-ной с/с в символьный код в 10-ной с/с
WRITE_STRING	Вывод строки на экран
WRITE_SIZE	Запись десятичного числа в строку
PRINT_MCB	Вывод информации о цепочке блоков управления памятью
FREE_UNUSED_MEMORY	Освобождение неиспользуемой памяти
GET_EXTRA_MEMORY	Выделение дополнительной памяти

В результате выполнения были получены следующие значения(рис.1-4):

Рисунок 1 – результат работы программы по условиям 1 пункта

```
C:\>LAB3_1.COM
Size of accessed memory: 648912 byte
Size of extended memory: 245760 byte
MCB:01 Address: 016F PSP adress: 0008 Size: 16 SD/SC:
MCB:02 Address: 0171 PSP adress: 0000 Size: 64 SD/SC: DPMILOAD
MCB:03 Address: 0176 PSP adress: 0040 Size: 256 SD/SC:
MCB:04 Address: 0187 PSP adress: 0192 Size: 144 SD/SC:
MCB:05 Address: 0191 PSP adress: 0192 Size: 648912 SD/SC: LAB3_1
```

Рисунок 2 – результат работы программы по условиям 2 пункта

```
C:\>lab3_2.com  
  
Size of accessed memory: 648912 byte  
Size of extended memory: 245760 byte  
MCB:01 Address: 016F PSP address: 0008 Size: 16 SD/SC:  
MCB:02 Address: 0171 PSP address: 0000 Size: 64 SD/SC: DPMILOAD  
MCB:03 Address: 0176 PSP address: 0040 Size: 256 SD/SC:  
MCB:04 Address: 0187 PSP address: 0192 Size: 144 SD/SC:  
MCB:05 Address: 0191 PSP address: 0192 Size: 832 SD/SC: LAB3_2  
MCB:06 Address: 01C6 PSP address: 0000 Size: 648064 SD/SC: T \| / 0 2 |  
C:\>
```

Рисунок 3 – результат работы программы по условиям 3 пункта

```
C:\>lab3_3.com

Size of accessed memory: 648912 byte
Success!
Size of extended memory: 245760 byte
MCB:01 Address: 016F PSP address: 0008 Size: 16 SD/SC:
MCB:02 Address: 0171 PSP address: 0000 Size: 64 SD/SC: DPMILOAD
MCB:03 Address: 0176 PSP address: 0040 Size: 256 SD/SC:
MCB:04 Address: 0187 PSP address: 0192 Size: 144 SD/SC:
MCB:05 Address: 0191 PSP address: 0192 Size: 864 SD/SC: LAB3_3
MCB:06 Address: 01C8 PSP address: 0192 Size: 65536 SD/SC: LAB3_3
MCB:07 Address: 11C9 PSP address: 0000 Size: 582480 SD/SC: µTP
C:\>
```

Рисунок 4 – результат работы программы по условиям 4 пункта

```
C:\>lab3_4.com

Size of accessed memory: 648912 byte
Memory Error!
Size of extended memory: 245760 byte
MCB:01 Address: 016F PSP address: 0008 Size: 16 SD/SC:
MCB:02 Address: 0171 PSP address: 0000 Size: 64 SD/SC: DPMILOAD
MCB:03 Address: 0176 PSP address: 0040 Size: 256 SD/SC:
MCB:04 Address: 0187 PSP address: 0192 Size: 144 SD/SC:
MCB:05 Address: 0191 PSP address: 0192 Size: 864 SD/SC: LAB3_4
MCB:06 Address: 01C8 PSP address: 0000 Size: 648032 SD/SC: LAB3_3
C:\>
```

Выводы.

В ходе лабораторной работы были исследованы структуры данных и работа функций управления памятью ядра операционной системы, а затем написана утилита, которая выводит информацию об исследованной основной памяти.

ПРИЛОЖЕНИЕ А

ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что означает “Доступный объём памяти”?

Доступный объём памяти – часть оперативной памяти выделяемой системой программе для её корректной работы.

2. Где МСВ блок Вашей программы в списке?

Для ответа на вопрос обратимся к рисункам:

В первом случае МСВ блок программы находится в конце списка.

Во втором случае МСВ блок программы находится на предпоследнем месте т.к. последнее место занимает блок с высвобожденной неиспользуемой памятью.

В третьем случае МСВ блок программы находится третьим снизу т.к. сначала мы высвобождаем неиспользуемую память, а после выделяем от неё небольшую часть под программу.

В четвёртом случае МСВ блок программы находится предпоследним т.к. мы не смогли выделить дополнительную память не освободив незанятую.

3. Какой размер памяти занимает программа в каждом случае?

В первом случае всю свободную память, т.е. 648912.

Во втором только необходимый объём памяти, т.е. 832

В третьем случае необходимый объём памяти + дополнительно выделенный 64Кб объём памяти, т.е. 864+65536.

В четвёртом случае только необходимый объём, так как память была освобождена, но освобождена после запроса о выделении дополнительной памяти, т.е. 864.