

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

Студент гр. 9381

Щеглов Д.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

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.

Требуется написать и отладить .COM модуль, который 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.

Процедуры в программе:

TETR_TO_HEX – перевод половины байта в 16 с/с.

BYTE_TO_HEX – перевод байта в два символа в 16 с/с.

WRD_TO_HEX – перевод двухбайтного числа в 16 с/с.

BYTE_TO_DEC – перевод байта в число в 10 с/с и запись в строку по адресу в регистре SI.

OUTPUTTING_STRING_TO_CONSOLE – вывод строки на консоль.

WRITE_NUMBER_MEM – перевод и вывод количества памяти, заданной в параграфах, в байты.

OUTPUTTING_MCB – вывод цепочки блоков MCB.

FREE_MEMORY – освобождение занятой памяти.

GET_MEM – запрос на аллокацию дополнительной памяти.

Выполнение работы.

Сначала был написан код для .COM модуля. Результат работы модуля представлен на рисунке 1.

```
C:\>lab3_1.com
Number of accessed memory: 648912 bytes
Size of extended memory: 245760 bytes
MCB 1: Segment address of PSP: 0008 Size of area: 16 SD/SC:
MCB 2: Segment address of PSP: 0000 Size of area: 64 SD/SC:
MCB 3: Segment address of PSP: 0040 Size of area: 256 SD/SC:
MCB 4: Segment address of PSP: 0192 Size of area: 144 SD/SC:
MCB 5: Segment address of PSP: 0192 Size of area: 648912 SD/SC: LAB3_1
C:\>
```

Рисунок 1 – Результат работы .COM модуля

Модуль вывел количество доступной и расширенной памяти, а также информацию о цепочке MCB.

Далее код модуля был изменен таким образом, чтобы программа освобождала память, которую она не занимает. Результат работы модуля представлен на рисунке 2.

```
C:\>lab3_2.com
Number of accessed memory: 648912 bytes
Size of extended memory: 245760 bytes
MCB 1: Segment address of PSP: 0008 Size of area: 16 SD/SC:
MCB 2: Segment address of PSP: 0000 Size of area: 64 SD/SC:
MCB 3: Segment address of PSP: 0040 Size of area: 256 SD/SC:
MCB 4: Segment address of PSP: 0192 Size of area: 144 SD/SC:
MCB 5: Segment address of PSP: 0192 Size of area: 768 SD/SC: LAB3_2
MCB 6: Segment address of PSP: 0000 Size of area: 648128 SD/SC: X▲▼s♠°
C:\>
```

Рисунок 2 – Результат работы .COM модуля

Далее код модуля был изменен таким образом, чтобы программа после освобождения памяти, которую она не занимает, запрашивала 64Кб памяти. Также проверялся флаг CF при запросе памяти и выводилась строка, сообщающая о результате запроса. В данном случае запрос о выделении памяти завершился успешно. Результат работы модуля представлен на рисунке 3.

```
C:\>lab3_3.com
Number of accessed memory: 648912 bytes
Allocating memory ended successful!
Size of extended memory: 245760 bytes
MCB 1: Segment address of PSP: 0008 Size of area: 16 SD/SC:
MCB 2: Segment address of PSP: 0000 Size of area: 64 SD/SC:
MCB 3: Segment address of PSP: 0040 Size of area: 256 SD/SC:
MCB 4: Segment address of PSP: 0192 Size of area: 144 SD/SC:
MCB 5: Segment address of PSP: 0192 Size of area: 880 SD/SC: LAB3_3
MCB 6: Segment address of PSP: 0192 Size of area: 65536 SD/SC: LAB3_3
MCB 7: Segment address of PSP: 0000 Size of area: 582464 SD/SC: .64
C:\>
```

Рисунок 3 – Результат работы .COM модуля

Далее код модуля был изменен таким образом, чтобы программа запрашивала 64Кб памяти, а после освобождала память. Также проверялся флаг CF при запросе памяти и выводилась строка, сообщающая о результате запроса. В данном случае запрос о выделении памяти завершился неудачно. Результат работы модуля представлен на рисунке 4.

```
C:\>lab3_4.com
Number of accessed memory: 648912 bytes
Allocating memory ended with failure!
Size of extended memory: 245760 bytes
MCB 1: Segment address of PSP: 0008 Size of area: 16 SD/SC:
MCB 2: Segment address of PSP: 0000 Size of area: 64 SD/SC:
MCB 3: Segment address of PSP: 0040 Size of area: 256 SD/SC:
MCB 4: Segment address of PSP: 0192 Size of area: 144 SD/SC:
MCB 5: Segment address of PSP: 0192 Size of area: 880 SD/SC: LAB3_4
MCB 6: Segment address of PSP: 0000 Size of area: 648016 SD/SC: LAB3_3
C:\>
```

Рисунок 4 – Результат работы .COM модуля

Ответы на вопросы.

1. Что означает “доступный объем памяти”?

Доступный объем памяти – часть оперативной памяти, выделяемая программе системой для ее работы.

2. Где МСВ блок Вашей программы в списке?

В случае первой программы, МСВ блок находится в конце списка, т.к. программа не освобождает память, которую она занимает.

В случае второй программы, ее блок находится на предпоследнем месте, т.к. она освободила память, которую не занимает.

В случае третьей программы, ее блок находится на двух местах(3е и 2е с конца), т.к. программа сначала освободила память, а потом выделила.

В четвертом случае блок программы находится на предпоследнем месте, т.к. она не смогла выделить память (64Кб).

3. Какой размер памяти занимает программа в каждом случае?

Первая программа занимает весь доступный объем памяти – 648912 байт.

Вторая программа занимает 768 байт.

Третья программа занимает 880 байт и выделенные 64Кб.

Четвертая программа занимает только 880 байт, т.к. выделить 64Кб ей не удалось.

Выводы.

Была исследована организация управления памятью.