

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. 0381

Соколов 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.

Шаг 1. 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.

Шаг 2. Шаг 2. Измените программу таким образом, чтобы она освобождала память, которую она не занимает. Для этого используйте функцию 4Ah прерывания 21h (пример в разделе «Использование функции 4Ah»). Повторите эксперимент, запустив модифицированную программу. Сравните выходные данные с результатами, полученными на предыдущем шаге. 3. Сохраните результаты, полученные программой, и включите их в отчет в виде скриншота.

Шаг 3. Измените программу еще раз таким образом, чтобы после освобождения памяти, программа запрашивала 64Кб памяти функцией 48h

прерывания 21H. Повторите эксперимент, запустив модифицированную программу. Сравните выходные данные с результатами, полученными на предыдущих шагах. Сохраните результаты, полученные программой, и включите их в отчет в виде скриншота.

Шаг 4. Измените первоначальный вариант программы, запросив 64Кб памяти функцией 48H прерывания 21H до освобождения памяти. Обязательно обрабатывайте завершение функций ядра, проверяя флаг CF. Сохраните результаты, полученные программой, и включите их в отчет в виде скриншота.

Шаг 5. Оцените результаты, полученные на предыдущих шагах. Ответьте на контрольные вопросы и оформите отчет.

Выполнение работы.

Были написаны строки для вывода информации и следующие процедуры:

- FREE_MEM_INFO - выводит размер доступной памяти в байтах.
- EXTENDED_MEM_INFO - выводит размер расширенной памяти в байтах.
- MCB_INFO - выводит тип, адрес, сегмент владельца и размер участка каждого MCB-блока.
- WRD_TO_DEC - переводит размер памяти в параграфах в байты и добавляет их десятичное представление в строку.

В результате выполнения были получены следующие значения:

```
S:\>lab_31.com  
Size of free memory:      64 byte  
Size of extended memory: 15728640 byte  
MCB: 4Dh  Address: 016Fh  Owner: MS DOS  Area size: 16 byte  
SC/SD:  
MCB: 4Dh  Address: 0171h  Owner: Free    Area size: 64 byte  
SC/SD:  
MCB: 4Dh  Address: 0176h  Owner: 0040h   Area size: 256 byte  
SC/SD:  
MCB: 4Dh  Address: 0187h  Owner: 0192h   Area size: 144 byte  
SC/SD:  
MCB: 5Ah  Address: 0191h  Owner: 0192h   Area size: 648912 byte  
SC/SD: LAB_31
```

Рисунок 1. Результат работы программы по условиям 1 пункта

На 2 шаге программа была модифицирована для освобождения неиспользуемой памяти.

```
S:\>lab_32.com  
Size of free memory:      583360 byte  
Size of extended memory: 15728640 byte  
MCB: 4Dh  Address: 016Fh  Owner: MS DOS  Area size: 16 byte  
SC/SD:  
MCB: 4Dh  Address: 0171h  Owner: Free    Area size: 64 byte  
SC/SD:  
MCB: 4Dh  Address: 0176h  Owner: 0040h   Area size: 256 byte  
SC/SD:  
MCB: 4Dh  Address: 0187h  Owner: 0192h   Area size: 144 byte  
SC/SD:  
MCB: 4Dh  Address: 0191h  Owner: 0192h   Area size: 65536 byte  
SC/SD: LAB_32  
MCB: 5Ah  Address: 1192h  Owner: Free    Area size: 583360 byte  
SC/SD: ength
```

Рисунок 2. Результат работы программы по условиям 2 пункта

Затем, программа изменена таким образом, что после освобождения памяти запрашивается еще 64 кб.

```

S:\>lab_33.com
Size of free memory:      517808 byte
Size of extended memory:  15728640 byte
MCB: 4Dh  Address: 016Fh  Owner: MS DOS  Area size: 16 byte
SC/SD:
MCB: 4Dh  Address: 0171h  Owner: Free    Area size: 64 byte
SC/SD:
MCB: 4Dh  Address: 0176h  Owner: 0040h   Area size: 256 byte
SC/SD:
MCB: 4Dh  Address: 0187h  Owner: 0192h   Area size: 144 byte
SC/SD:
MCB: 4Dh  Address: 0191h  Owner: 0192h   Area size: 65536 byte
SC/SD: LAB_33
MCB: 4Dh  Address: 1192h  Owner: 0192h   Area size: 65536 byte
SC/SD: LAB_33
MCB: 5Ah  Address: 2193h  Owner: Free    Area size: 517808 byte
SC/SD:

```

Рисунок 3. Результат работы программы по условиям 3 пункта

В последней версии программа пробует получить дополнительную память, перед этим её не освободив. Функция 48h устанавливает флаг CF, который говорит о том, что невозможно выделить память. Об этом выводится специальное сообщение, и дальше выводимая информация совпадает с первым пунктом.

```

S:\>lab_34.com
unable to allocate memory
Size of free memory:      64 byte
Size of extended memory:  15728640 byte
MCB: 4Dh  Address: 016Fh  Owner: MS DOS  Area size: 16 byte
SC/SD:
MCB: 4Dh  Address: 0171h  Owner: Free    Area size: 64 byte
SC/SD:
MCB: 4Dh  Address: 0176h  Owner: 0040h   Area size: 256 byte
SC/SD:
MCB: 4Dh  Address: 0187h  Owner: 0192h   Area size: 144 byte
SC/SD:
MCB: 5Ah  Address: 0191h  Owner: 0192h   Area size: 648912 byte
SC/SD: LAB_34

```

Рисунок 4. Результат работы программы по условиям 4 пункта

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.