

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МОЭВМ

ОТЧЕТ
по лабораторной работе №3
по дисциплине «Операционные системы»

Тема: Исследование организации управления основной памятью

Студент гр. 0381

Печеркин А. С.

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 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.

Задание.

Шаг 1. Для выполнения лабораторной работы необходимо написать и отладить программный модуль типа .COM, 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 MCB выводятся как символы, не следует преобразовывать их в шестнадцатеричные числа.

Шаг 2. Измените программу таким образом, чтобы она освобождала память, которую она не занимает. Для этого используйте функцию 4Ah прерывания 21h. Повторите эксперимент, запустив модифицированную

программу. Сравните выходные данные с результатами, полученными на предыдущем шаге. 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 48Н прерывания 21Н. 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 48Н прерывания 21Н до освобождения памяти. Обязательно обрабатывайте завершение функций ядра, проверяя флаг CF. Сохраните результаты, полученные программой, и включите их в отчет в виде скриншота.

Выполнение работы.

Шаг 1. При запуске программы вызывается процедура PRINT_AVAILABLE_MEM, которая выводит доступную память. Для этого используется функция 48h прерывания 21h. Если с помощью этой функции запросить заведомо больший объем памяти, чем свободно в системе, она вернет размер свободной памяти. Полученное значение измеряется в параграфах. Оно переводится в байты и выводится на экран в десятичной системе счисления.

Далее, в процедуре PRINT_EXTENDED_MEM, выводится объем расширенной памяти, для этого программа обращается к ячейкам 30h и 31h памяти CMOS. Полученное значение является размером расширенной памяти в килобайтах. Для вывода на экран значение переводится в байты. Вывод осуществляется в десятичной системе счисления.

Затем, с помощью функции 52h int 21h, программа получает адрес самого первого MCB. Потом вызывается функция NEXTMCB, которая копирует содержимое блока, на который указывает es, в таблицу расположенную в сегменте данных программы. Данные из этой таблицы выводятся на экран. Для перехода к следующему блоку используется сегментный адрес текущего блока и размер управляемой им области. Считывание цепочки блоков завершается, когда достигнут последний блок в списке.

Результат работы программы:

Программа выводит размер доступной памяти равный 64 байтам, так как она занимает весь объем памяти, но есть свободный блок памяти после блока DOS длиной 64 байта. В цепочке MCB содержится 5 блоков. Размер расширенной памяти почти 16 МБ.

```
F:\>1.com
Avaliable memory: 64 bytes
Extended memory: 15728640 bytes
MCB Type: 4Dh MCB Address: 016Fh Owner: DOS Length: 16 bytes
Reserved: 00h Text:
MCB Type: 4Dh MCB Address: 0171h Owner: free Length: 64 bytes
Reserved: 00h Text:
MCB Type: 4Dh MCB Address: 0176h Owner: 0040h Length: 256 bytes
Reserved: 00h Text:
MCB Type: 4Dh MCB Address: 0187h Owner: 0192h Length: 144 bytes
Reserved: 00h Text:
MCB Type: 5Ah MCB Address: 0191h Owner: 0192h Length: 648912 bytes
Reserved: 00h Text: 1
F:\>_
```

Шаг 2. Модифицированная программа освобождает память, которую не использует, поэтому теперь объем свободной памяти увеличился и появился свободный блок в конце адресного пространства.

```

F:\>1.com
Avaliable memory: 583360 bytes
Extended memory: 15728640 bytes
MCB Type: 4Dh MCB Address: 016Fh Owner: DOS Length: 16 bytes
Reserved: 00h Text:
MCB Type: 4Dh MCB Address: 0171h Owner: free Length: 64 bytes
Reserved: 00h Text:
MCB Type: 4Dh MCB Address: 0176h Owner: 0040h Length: 256 bytes
Reserved: 00h Text:
MCB Type: 4Dh MCB Address: 0187h Owner: 0192h Length: 144 bytes
Reserved: 00h Text:
MCB Type: 4Dh MCB Address: 0191h Owner: 0192h Length: 65536 bytes
Reserved: 00h Text: 1
MCB Type: 5Ah MCB Address: 1192h Owner: free Length: 583360 bytes
Reserved: 20h Text: engh
F:\>

```

Шаг 3. Теперь программа после освобождения памяти запрашивает ещё 64 Кб памяти. В цепочке блоков появился ещё один блок размером 64 Кб, который принадлежит программе.

```

F:\>1.com
Avaliable memory: 517808 bytes
Extended memory: 15728640 bytes
MCB Type: 4Dh MCB Address: 016Fh Owner: DOS Length: 16 bytes
Reserved: 00h Text:
MCB Type: 4Dh MCB Address: 0171h Owner: free Length: 64 bytes
Reserved: 00h Text:
MCB Type: 4Dh MCB Address: 0176h Owner: 0040h Length: 256 bytes
Reserved: 00h Text:
MCB Type: 4Dh MCB Address: 0187h Owner: 0192h Length: 144 bytes
Reserved: 00h Text:
MCB Type: 4Dh MCB Address: 0191h Owner: 0192h Length: 65536 bytes
Reserved: 00h Text: 1
MCB Type: 4Dh MCB Address: 1192h Owner: 0192h Length: 65536 bytes
Reserved: 20h Text: 1
MCB Type: 5Ah MCB Address: 2193h Owner: free Length: 517808 bytes
Reserved: 00h Text:

```

Шаг 4. Если попытаться запросить дополнительную память, предварительно не освободив её, функция 48h устанавливает флаг CF. Этот флаг говорит о том, что невозможно выделить память. В этом случае программа выводит сообщение о том, что недостаточно памяти.

```
F:\>1.com
Not enough memory
Avaliable memory: 64 bytes
Extended memory: 15728640 bytes
MCB Type: 4Dh MCB Address: 016Fh Owner: DOS Length: 16 bytes
Reserved: 00h Text:
MCB Type: 4Dh MCB Address: 0171h Owner: free Length: 64 bytes
Reserved: 00h Text:
MCB Type: 4Dh MCB Address: 0176h Owner: 0040h Length: 256 bytes
Reserved: 00h Text:
MCB Type: 4Dh MCB Address: 0187h Owner: 0192h Length: 144 bytes
Reserved: 00h Text:
MCB Type: 5Ah MCB Address: 0191h Owner: 0192h Length: 648912 bytes
Reserved: 00h Text: 1
F:\>_
```

Выводы.

Была исследована организация управлением основной памятью, разработана программа, которая выводит информацию о использовании памяти в системе с помощью системных функций и обращений к блокам МСВ.

ВОПРОСЫ

1. Что означает "доступный объем памяти"?

Доступный объем памяти – это размер наибольшего свободного участка памяти, то есть, если есть несколько свободных участков памяти разного размера не расположенных подряд, доступным объемом памяти будет являться размер большего из этих участков. Таким образом, доступный объем памяти определяет максимальный размер блока, который может запросить программа.

2. Где MCB блок Вашей программы в списке?

В первом случае блок находится в конце списка.

Во втором случае блок программы предпоследний в списке, за ним следует свободный блок.

В третьем случае программе принадлежит 5 и 6 блоки, последний блок в списке свободный.

```
MCB Type: 4Dh MCB Address: 0191h Owner: 0192h Length: 65536 bytes  
Reserved: 00h Text: 1  
MCB Type: 4Dh MCB Address: 1192h Owner: 0192h Length: 65536 bytes  
Reserved: 20h Text: 1
```

3. Какой размер памяти занимает программа в каждом случае?

В первом случае программа занимает все доступное пространство, то есть 648912 байт.

Во втором случае программа занимает 65536 байт.

В третьем случае программа занимает 131072 байт.

В четвертом случае программа занимает 648912 байт, так как невозможно выделить дополнительную память.