

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. 0381

Ибатов Н.Э.

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 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, просматривают и преобразуют этот список.

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 MSB выводятся как символы, не следует преобразовывать их в шестнадцатеричные числа. Запустите программу и внимательно оцените результаты. Сохраните результаты, полученные программой, и включите их в отчет в виде скриншота.

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. 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.

Шаг 1. При запуске программы вызывается процедура `print_available`, которая выводит доступную память. Для этого была использована функция 48h прерывания 21h. Значение, полученное данной процедурой, считается в параграфах, далее переводится в байты и выводится на экран.

В процедуре `print_extended`, выводится объем расширенной памяти, для этого программа обращается к ячейкам 30h и 31h памяти CMOS. Полученное значение является размером расширенной памяти в килобайтах, далее переводится в байты и выводится на экран.

Затем программа находит адрес первого MCB (функция 52h int 21h). Вызывается функция `mcb_info`, которая копирует содержимое блока, на который указывает `es`, в таблицу, расположенную в сегменте данных программы. Данные из таблицы выводятся.

Результат работы программы:

Доступная память равна 64 байтам, так как программа занимает весь объем памяти, но есть один свободный блок памяти после блока DOS длиной в 64 байта. Размер расширенной памяти примерно 16 Мб.

```

F:\>lb3.com
Avaliable memory: 64 bytes
Extended memory: 15728640 bytes
MCB Type: 4Dh MCB Address: 016Fh Owner: 0008 Size: 16 bytes
Reserved: 00h SD/SC:
MCB Type: 4Dh MCB Address: 0171h Owner: 0000 Size: 64 bytes
Reserved: 00h SD/SC:
MCB Type: 4Dh MCB Address: 0176h Owner: 0040 Size: 256 bytes
Reserved: 00h SD/SC:
MCB Type: 4Dh MCB Address: 0187h Owner: 0192 Size: 144 bytes
Reserved: 00h SD/SC:
MCB Type: 5Ah MCB Address: 0191h Owner: 0192 Size: 648912 bytes
Reserved: 00h SD/SC: LB3
F:\>

```

Шаг 2. Модифицированная программа освобождает память, которую не использует, поэтому доступная память увеличилась и освободился блок.

```

F:\>LB3_R.COM
Avaliable memory: 583360 bytes
Extended memory: 15728640 bytes
MCB Type: 4Dh MCB Address: 016Fh Owner: 0008 Size: 16 bytes
Reserved: 00h SD/SC:
MCB Type: 4Dh MCB Address: 0171h Owner: 0000 Size: 64 bytes
Reserved: 00h SD/SC:
MCB Type: 4Dh MCB Address: 0176h Owner: 0040 Size: 256 bytes
Reserved: 00h SD/SC:
MCB Type: 4Dh MCB Address: 0187h Owner: 0192 Size: 144 bytes
Reserved: 00h SD/SC:
MCB Type: 4Dh MCB Address: 0191h Owner: 0192 Size: 65536 bytes
Reserved: 00h SD/SC: LB3_R
MCB Type: 5Ah MCB Address: 1192h Owner: 0000 Size: 583360 bytes
Reserved: 20h SD/SC: ength
F:\>_

```

Шаг 3. После освобождения программа запрашивает дополнительно 64 Кб памяти, поэтому добавился еще один блок.

```

F:\>lb3_h.com
Avaliable memory: 517808 bytes
Extended memory: 15728640 bytes
MCB Type: 4Dh MCB Address: 016Fh Owner: 0008 Size: 16 bytes
Reserved: 00h SD/SC:
MCB Type: 4Dh MCB Address: 0171h Owner: 0000 Size: 64 bytes
Reserved: 00h SD/SC:
MCB Type: 4Dh MCB Address: 0176h Owner: 0040 Size: 256 bytes
Reserved: 00h SD/SC:
MCB Type: 4Dh MCB Address: 0187h Owner: 0192 Size: 144 bytes
Reserved: 00h SD/SC:
MCB Type: 4Dh MCB Address: 0191h Owner: 0192 Size: 65536 bytes
Reserved: 00h SD/SC: LB3_H
MCB Type: 4Dh MCB Address: 1192h Owner: 0192 Size: 65536 bytes
Reserved: 20h SD/SC: LB3_H
MCB Type: 5Ah MCB Address: 2193h Owner: 0000 Size: 517808 bytes
Reserved: 00h SD/SC:
F:\>_

```

Шаг 4. Если попросить дополнительную память, заранее не освободив её, то функция 48h устанавливает флаг CF. Этот флаг говорит о том, что последующее выделение памяти невозможно. Программа выводит сообщение о том, что недостаточно памяти.

```

F:\>lb3_l.com
Not enough memory
F:\>

```

Выводы.

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

ПРИЛОЖЕНИЕ А

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.

- 2) Где МСВ блок вашей программы в списке?

В первом случае он находится в конце, во втором – предпоследний, после него следует свободный блок, в третьем – 5-й и 6-й.

- 3) Какой размер памяти занимает программа в каждом случае?

В первом – примерно 634 Кб, во втором – 64 Кб, в третьем – 128 Кб и в четвертом – примерно 634 Кб, так как не получилось выделить память.