

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

Колованов Р.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

Для исследования организации управления основной 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.

Функции и структуры данных.

Разработанная программа использует следующие функции и структуры данных:

Название процедуры	Предназначение процедуры
TETR_TO_HEX	Переводит значение тетрады (4-ех младших битов регистра AL) в цифру 16-ичной СС и представляет ее в виде символа, который далее записывается в регистр AL.
BYTE_TO_HEX	Переводит значение байта (регистра AL) в число 16-ичной СС и представляет его в виде двух символов, которые далее записываются в регистры AL и AH.
WORD_TO_HEX	Переводит значение слова (регистра AX) в число 16-ичной СС и представляет его в виде четырех символов, которые далее записываются по адресу, на который указывает DI.
BYTE_TO_DEC	Переводит значение байта (регистра AL) в число 10-ичной СС и представляет его в виде символов,

	которые далее записываются по адресу, на который указывает SI.
WORD_TO_DEC	Переводит значение слова (регистра AX) в число 10-ичной СС и представляет его в виде символов, которые далее записываются по адресу, на который указывает SI.
PRINT	Вызывает функцию вывода строки на экран (функция 09h прерывания 21h).
CLEAR_BLOCK_SIZE	Очищает поле, содержащее размер MCB блока.
PRINT_AVAILABLE_MEMORY_SIZE	Печатает на экран размер доступной памяти. Для получения размера доступной памяти используется функция 4Ah прерывания 21h.
PRINT_EXTENDED_MEMORY_SIZE	Печатает на экран размер расширенной памяти. Размер расширенной памяти берется из ячеек 30h и 31h CMOS.
PRINT_MCB_LIST	Печатает на экран цепочку блоков управления памятью (MCB).
READ_AND_PRINT_MCB	Печатает на экран блок управления памятью (MCB).
CHECK_AND_PRINT_ERROR	Выводит сообщение об ошибке и завершает программу в случае, если при выполнении функции прерывания 21h происходит ошибка.
FREE_MEMORY	Очищает неиспользуемую программой память при помощи функции 4Ah прерывания 21h.
ALLOC_MEMORY	Выделяет свободную память программе при помощи функции 48h прерывания 21h.

Ход работы.

Для начала был написан текст исходного COM модуля lab3_1.asm. Далее при помощи транслятора MASM.EXE, компоновщика LINK.EXE и утилиты EXE2BIN.EXE был скомпилирован COM модуль lab3_1.com с генерацией файла листинга и карты памяти, после чего загрузочный модуль был протестирован.

Результаты работы программы:

Листинг 1.

```
The size of the available memory: 633 kilobytes 720 bytes
The size of the extended memory: 15360 kilobytes
```

```
MCB address: 016F
MCB type: 4D
Block owner: MS DOS
Block size: 0 kilobytes 16 bytes
Reserved:
```

```
MCB address: 0171
MCB type: 4D
Block owner: Free
Block size: 0 kilobytes 64 bytes
Reserved:
```

```
MCB address: 0176
MCB type: 4D
Block owner: 0040
Block size: 0 kilobytes 256 bytes
Reserved:
```

```
MCB address: 0187
MCB type: 4D
Block owner: 0192
Block size: 0 kilobytes 144 bytes
Reserved:
```

```
MCB address: 0191
MCB type: 5A
Block owner: 0192
Block size: 633 kilobytes 720 bytes
Reserved: LAB3_1
```

Далее текст исходного COM модуля lab3_1.asm был изменен так, что теперь программа освободит неиспользуемую ею память, и сохранен в файл lab3_2.asm. Далее при помощи транслятора MASM.EXE, компоновщика LINK.EXE и утилиты EXE2BIN.EXE был скомпилирован COM модуль lab3_2.com с генерацией файла листинга и карты памяти, после чего загрузочный модуль был протестирован. Результаты работы программы:

Листинг 2.

```
The size of the available memory: 633 kilobytes 720 bytes
```

```
The size of the extended memory: 15360 kilobytes
```

```
MCB address: 016F
MCB type: 4D
Block owner: MS DOS
Block size:   0 kilobytes   16 bytes
Reserved:
```

```
MCB address: 0171
MCB type: 4D
Block owner: Free
Block size:   0 kilobytes   64 bytes
Reserved:
```

```
MCB address: 0176
MCB type: 4D
Block owner: 0040
Block size:   0 kilobytes  256 bytes
Reserved:
```

```
MCB address: 0187
MCB type: 4D
Block owner: 0192
Block size:   0 kilobytes  144 bytes
Reserved:
```

```
MCB address: 0191
MCB type: 4D
Block owner: 0192
Block size:   4 kilobytes   0 bytes
Reserved:   LAB3_2
```

```
MCB address: 0292
MCB type: 5A
Block owner: Free
Block size: 629 kilobytes  704 bytes
Reserved:  i  0x .□>
```

Сравнивая вывод программы с предыдущим, можно заметить, что теперь появился свободный блок памяти, а размер блока, выделенного программе, стал равен 4 килобайтам.

Далее текст исходного COM модуля lab3_2.asm был изменен так, что теперь программа после освобождения неиспользуемой памяти запрашивает 64 килобайта, и сохранен в файл lab3_3.asm. Далее при помощи транслятора MASM.EXE, компоновщика LINK.EXE и утилиты EXE2BIN.EXE был скомпилирован COM модуль lab3_3.com с генерацией файла листинга и карты памяти, после чего загрузочный модуль был протестирован. Результаты работы программы:

Листинг 3.

```

The size of the available memory: 633 kilobytes 720 bytes
The size of the extended memory: 15360 kilobytes

MCB address: 016F
MCB type: 4D
Block owner: MS DOS
Block size: 0 kilobytes 16 bytes
Reserved:

MCB address: 0171
MCB type: 4D
Block owner: Free
Block size: 0 kilobytes 64 bytes
Reserved:

MCB address: 0176
MCB type: 4D
Block owner: 0040
Block size: 0 kilobytes 256 bytes
Reserved:

MCB address: 0187
MCB type: 4D
Block owner: 0192
Block size: 0 kilobytes 144 bytes
Reserved:

MCB address: 0191
MCB type: 4D
Block owner: 0192
Block size: 4 kilobytes 0 bytes
Reserved: LAB3_3

MCB address: 0292
MCB type: 4D
Block owner: 0192
Block size: 64 kilobytes 0 bytes
Reserved: i LAB3_3

MCB address: 1293
MCB type: 5A
Block owner: Free
Block size: 565 kilobytes 688 bytes
Reserved:

```

Сравнивая вывод программы с предыдущим, можно заметить, что теперь появился еще один блок памяти размером 64 килобайта, принадлежащей программе.

Далее текст исходного COM модуля lab3_3.asm был изменен так, что теперь программа сначала запрашивает 64 килобайта памяти, а после чего освобождает ее, и сохранен в файл lab3_4.asm. Далее при помощи транслятора MASM.EXE, компоновщика LINK.EXE и утилиты EXE2BIN.EXE был скомпилирован COM модуль lab3_4.com с генерацией файла листинга и карты

памяти, после чего загрузочный модуль был протестирован. Результаты работы программы:

Листинг 4 .

```
The size of the available memory: 633 kilobytes 720 bytes
The size of the extended memory: 15360 kilobytes
Error! Code: 8
```

Сравнивая вывод программы с предыдущим, можно заметить, что программа завершилась после попытки выделить дополнительные 64 килобайта памяти. Поскольку свободной памяти больше не осталось, функция 48h прерывания 21h завершилась с ошибкой под номером 8, после чего программа завершила свое выполнение.

Результаты исследования проблем.

1. Что означает “доступный объем памяти”?

Это объем памяти, доступный для использования программой.

2. Где MCB блок Вашей программы в списке?

У всех MCB блоков моей программы поле block owner имеет значение 0192.

3. Какой размер памяти занимает программа в каждом случае?

В первом случае – 633 килобайта 720 байт, во втором случае – 4 килобайта, в третьем случае – 68 килобайт, в четвертом случае – 633 килобайта 720 байт (поскольку программа еще не успела освободить память).

Заключение.

Были изучены структуры данных и работа функций управления памятью ядра операционной системы.