**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МОЭВМ**

Отчет

по лабораторной работе №3

по дисциплине «Операционные системы»

на тему

# «Исследование организации управления основной памятью»

Студент группы 6383 Базаров И.А.  
Преподаватель Губкин А.Ф.

Санкт-Петербург

2018

**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, просматривают и преобразуют этот список.

В лабораторной работе исследуются структуры данных и работа функций управления памятью ядра операционной системы.

**Необходимые сведения для составления программы**

Учет занятой и свободной памяти ведется при помощи списка блоков управления памятью MCB (Memory Control Block). MCB занимает 16 байт (параграф) и располагается всегда с адреса кратного 16 (адрес сегмента ОП) и находится в адресном пространстве непосредственно перед тем участком памяти, которым он управляет.

MCB имеет следующую структуру:

|  |  |  |
| --- | --- | --- |
| Смещение | Длина поля (байт) | Содержимое поля |
| 00h | 1 | тип MCB:  5Ah, если последний в списке,  4Dh, если не последний |
| 01h | 2 | Сегментный адрес PSP владельца участка памяти, либо  0000h - свободный участок,  0006h - участок принадлежит драйверу  OS XMS UMB  0007h - участок является исключенной верхней памятью драйверов  0008h - участок принадлежит MS DOS  FFFAh - участок занят управляющим блоком 386MAX UMB  FFFDh - участок заблокирован 386MAX  FFFEh - участок принадлежит 386MAX UMB |
| 03h | 2 | Размер участка в параграфах |
| 05h | 3 | Зарезервирован |
| 08h | 8 | "SC" - если участок принадлежит MS DOS, то в нем системный код  "SD" - если участок принадлежит MS DOS, то в нем системные данные |

По сегментному адресу и размеру участка памяти, контролируемого этим MCB можно определить местоположение следующего MCB в списке.

Адрес первого MCB хранится во внутренней структуре MS DOS, называемой "List of Lists" (список списков). Доступ к указателю на эту структуру можно получить используя функцию 52h "Get List of Lists" int 21h. В результате выполнения этой функции ES:BX будет указывать на список списков. Слово по адресу ES:[BX-2] и есть адрес самого первого MCB.

Размер расширенной памяти находится в ячейках 30h, 31h CMOS. CMOS это энергонезависимая память, в которой хранится информация о конфигурации ПЭВМ. Объем памяти составляет 64 байта. Размер расширенной памяти в Кбайтах можно определить, обращаясь к ячейкам CMOS.

**Использование функции 4АH**

Использование функции 4Ah позволяет освободить неиспользуемую программой память, поскольку при загрузке модуля вся память выделяется этому модулю, в соответствии со стратегией однопрограммных ОС.

Для этого следует использовать функции 4Ah прерывания 21h. Пример обращения к этой функции:

MOV AH, 4AH

MOV BX, mem-size ;размер памяти программы

INT 21H

В регистр BX заносится размер памяти в параграфах, который необходимо оставить программе.

Если занести заведомо больший размер памяти, чем может предоставить OС, то в регистре BX возвращается размер доступной памяти в параграфах. В этом случае следует использовать следующее обращение:

MOV AH, 4AH

MOV BX, 0FFFFH ; заведомо большая память

INT 21H

**Сведения о функциях и структурах управляющей программы**

|  |  |
| --- | --- |
| **Название процедуры** | **Описание процедуры** |
| WRT | Вывод строки на экран |
| AVAILABLE\_MEMORY | Определение количества доступной памяти |
| SIZE\_EXPAN\_MEM | Определение размера расширенной памяти |
| MCB\_LIST | Вывод цепочки блоков управления памятью |
| PARAGRAPH | Перевод из параграфов в байты в виде десятичных чисел |
| DWRD\_TO\_DEC | Перевод двойного слова из двоичной системы счисления в десятичную |
| TETR\_TO\_HEX | Перевод из четверичной системы счисления в шестнадцатеричную |
| BYTE\_TO\_HEX | Перевод из двоичной системы счисления в шестнадцатеричную |
| WRD\_TO\_HEX | Перевод слова (2 байта) шестнадцатеричную систему счисления |
| BYTE\_TO\_DEC | Перевод из двоичной системы счисления в десятичную |

**Порядок выполнения работы**

1. Написание исходного .СОМ модуля

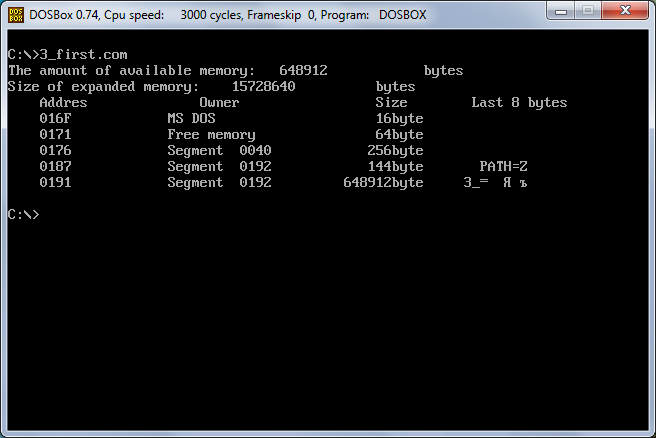
Для выполнения лабораторной работы был написан и отлажен исходный **.СОМ** модуль, который выбирает и распечатывает следующую информацию:

-количество доступной памяти

-размер расширенной памяти

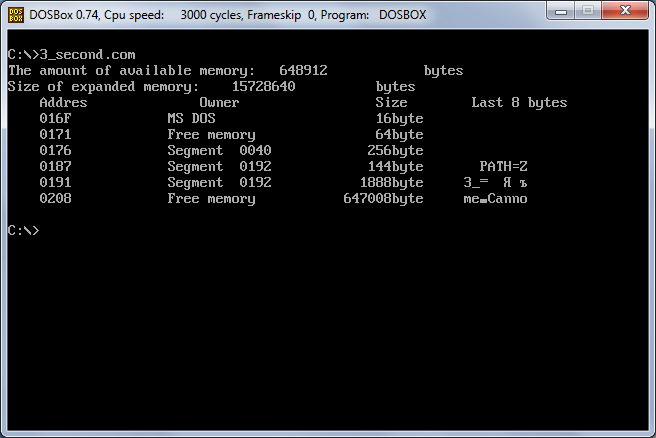
-выводит цепочку блоков управления памятью

Адреса при выводе представляются шестнадцатеричными числами. Объём памяти функциями управления памятью выводятся в параграфах. Необходимо преобразовать его в байты и выводить в виде десятичных чисел. Последние восемь байт МСВ выводятся как символы, не следует преобразовывать их в шестнадцатеричные числа. Результаты, полученные программой:

  
Рис.1

1. Изменение программы с освобождением памяти

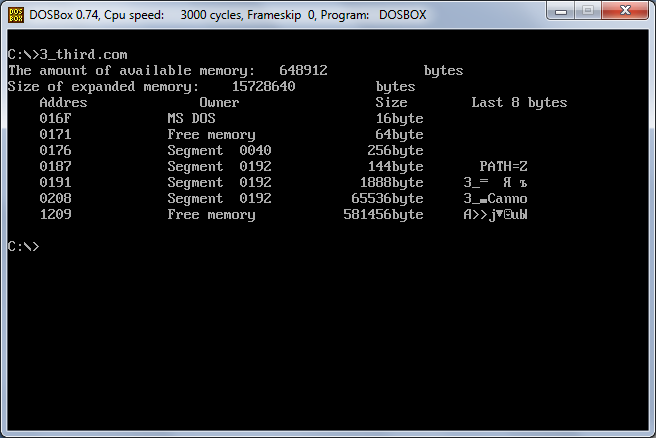
Программа была изменена таким образом, чтобы она освобождала память, которую она не занимает. Для этого была использована функция 4АH прерывания 21H. Эксперимент повторен при запуске модифицированной программы. Результаты, полученные программой:

  
Рис.2

По сравнению с результатами на предыдущем шаге размер блока 0191 уменьшился, при освобождении памяти добавился еще один блок по адресу 0208, память которого свободна.

1. Изменение программы с освобождением и запросом памяти

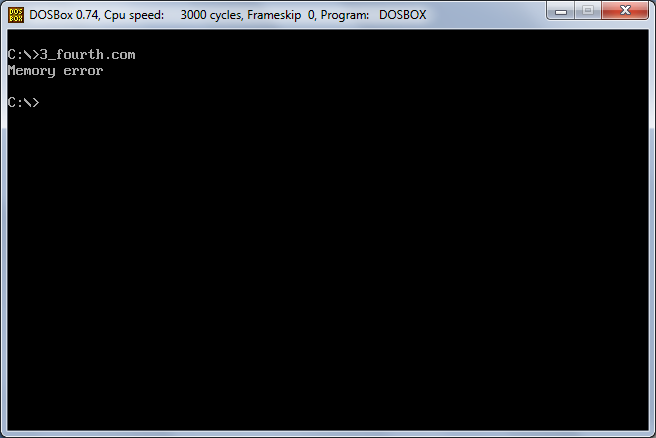
Программа была изменена таким образом, чтобы после освобождения памяти, программа запрашивала 64Кб памяти функцией 48H прерывания 21H. Эксперимент повторен при запуске модифицированной программы. Результаты, полученные программой:

  
Рис.3

В результате выполнения данной программы размер блока памяти по адресу 0208 стал 64Кб, а размер последнего блока свободной памяти уменьшился.

1. Изменение программы с запросом и освобождением памяти

Изменен первоначальный вариант программы, запросив 64Кб памяти функцией 48H прерывания 21H до освобождения памяти. Обработано завершение функций ядра, проверив флаг CF. Результаты, полученные программой:

  
Рис.4

Попытка выделить память без её предварительного освобождения привела к ошибке (определено по состоянию флага CF).

**Ответы на вопросы**

1. Что означает «доступный объём памяти»?

Доступный объём памяти -объём оперативной памяти, который используется программой (где программа хранит некоторые свои данные).

1. Где МСВ блок Вашей программы в списке?

1-й пункт: блок программы по адресу 0191 и занимает всю доступную память;

2-й пункт: блок программы по адресу 0191;

3-й пункт: блоки по адресам 0191 и 01F7.

1. Какой размер памяти занимает программа в каждом случае?

1-й пункт: 649056 байт (все свободную память).

2-й пункт: 2032 байта (необходимый объем памяти).

3-й пункт: 67568 байт (необходимый объем + 64Кб).