**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**

|  |  |  |
| --- | --- | --- |
| Студентка гр. 6381 |  | Герасимова Д.В. |
| 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 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.

**Постановка задачи:**

**Шаг 1.** Необходимо написать и отладить программный модуль типа .COM, 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 МСВ выводятся как символы, не следует преобразовывать их в шестнадцатеричные числа.

**Шаг 2.** Далее необходимо изменить программу таким образом, чтобы она освобождала память, которую она не занимает. Для этого используйте функцию 4Ah прерывания 21h (пример в разделе «Использование функции 4АН»).

**Шаг 3.** Затем необходимо изменить программу еще раз таким образом, чтобы после освобождения памяти, программа запрашивала 64Кб памяти функцией 48Н прерывания 21Н.

**Шаг 4.** Далее нужно изменить первоначальный вариант программы, запросив 64Кб памяти функцией 48Н прерывания 21Н до освобождения памяти.

**Шаг 5.** Оформить отчёт и ответить на контрольные вопросы.

**Необходимые сведения для составления программы:**

Учёт 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 f52h "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 следующим образом:

mov AL,30h ; запись адреса ячейки CMOS

out 70h,AL

in AL,71h ; чтение младшего байта

mov BL,AL ; размера расширенной памяти

mov AL,31h ; запись адреса ячейки CMOS

out 70h,AL

in AL,71h ; чтение старшего байта размера расширенной памяти

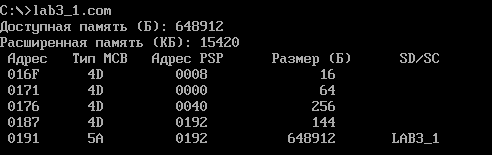
**В программе используются следующие процедуры:**

|  |  |
| --- | --- |
| **Название процедуры** | **Назначение** |
| TETR\_TO\_HEX | Перевод половины байта в шестнадцатеричную систему счисления |
| BYTE\_TO\_HEX | Перевод байта регистра AL в шестнадцатеричную систему счисления, помещая результат в AX |
| WRD\_TO\_HEX | Перевод двух байт регистра AX в шестнадцатеричную систему счисления, помещая результат в регистр DI |
| BYTE\_TO\_DEC | Перевод байта регистра AL в десятичную систему счисления, помещая результат в SI |
| GET\_MCB | Вывод таблицу MCB |
| PRINT | Вывод на экран |

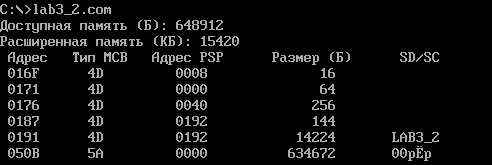
**Определение структуры данных:**

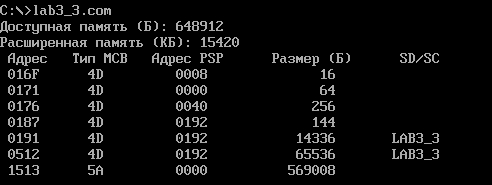
|  |  |  |
| --- | --- | --- |
| **Название поля данных** | **Тип** | **Назначение** |
| \_avl\_mem | db | Доступная память |
| \_ext\_mem | db | Расширенная память |
| \_table\_MCB | db | Таблица MCB |
| result | db | Строка для вывода |
| a | db | Строка для вывода |
| b | db | Строка для вывода |

**Результат работы программы:**

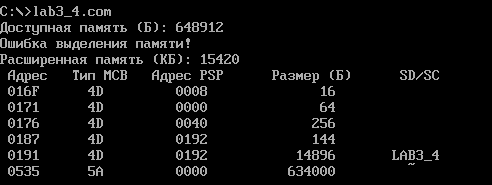


Все доступные 648912 байт отдаются программе.

  
Исходный код программы был изменён: теперь программа освобождает не занимаемую ею память. Создается новый блок, который обозначен как свободный участок, размером 634672 байт.



Код программы снова был изменён. Вначале происходит то же самое, что и во втором случае (освобождение памяти). Затем программа запрашивает 64 Кбайт (65536 байт) памяти. На свободном участке создается новый блок, который следует за основным блоком программы и занимает 65536 байт.

  
Код программы снова был изменён. Происходит запрос 64 Кбайт до освобождения памяти. Однако выдаётся ошибка, так как запрос памяти происходит в тот момент, когда вся доступная память занята программой. Затем происходит освобождение памяти, аналогично второму случаю.

**Ответы на контрольные вопросы:**

1. Что означает «доступный объём памяти»?

**Ответ:** доступный объём памяти – это объём базовой или стандартной памяти (conventional memory), эта память представляет собой "нижние" 640 Кбайт ОЗУ. Для использования базовой памяти не нужны никакие дополнительные драйверы, поскольку операционная система MS DOS изначально создана для работы в адресах 0 - 640 Кбайт

1. Где MCB блок Вашей программы в списке?

**Ответ:** по адресу 0192h.

1. Какой размер памяти занимает программа в каждом случае?

**Ответ:** 1) 648912 байт.

2) 648912 – 634672 – 16 = 14224 байт.

3) 648912 – 569008 – 65536 – 2\*16 = 14336 байт.

4) Произошла ошибка выделения дополнительной памяти,

648912 – 634000 – 16 = 14896 байт.

**Заключение:**

В ходе работы было проведено исследование структуры данных и работы функций управления памятью ядра операционной системы, а также рассмотрены нестраничная память и способы управления динамическими разделами.