**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра Математического Обеспечения ЭВМ**

отчет

**по лабораторной работе №3**

**по дисциплине «Операционные системы»**

Тема: **Исследование организации управления основной памятью.**

|  |  |  |
| --- | --- | --- |
| Студент гр. 6383 |  | Азаревич А.Д. |
| 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 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. Структура MCB представлена в табл. 1.

Табл. 1 – Структура MCB.

|  |  |  |
| --- | --- | --- |
| 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 |

Продолжение таблицы 1

|  |  |  |
| --- | --- | --- |
|  |  | 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

# Интерфейс функций и структур управляющей программы

Функции:

TETR\_TO\_HEX – переводит число из младшей половины регистра AL в его символьное обозначение (помещается в AL);

BYTE\_TO\_HEX – переводит число из регистра AL в его символьное обозначение (помещается в AX);

WRD\_TO\_HEX – переводит число из регистра AX в его символьное обозначение (помещается в память, на конец которой указывает DI);

QW\_TO\_DEC – переводит число из DX:AX в десятичное символьное обозначение (помещается в память, на конец которой указывает SI);

Pr\_1 – помещает в «выводящуюся строку» информацию о количестве доступной программе памяти в байтах;

Pr\_2 – помещает в «выводящуюся строку» информацию о размере расширенной памяти

Pr\_3 – печатает цепочку MCB(выводит информацию о типе MCB, адресе PSP, размере участка в байтах, SC/SD);

FREE\_MEM – освобождает не использующуюся программой память;

ALLOC\_MEM – выделяет программе память (если ОС не дало выделить память – выводит сообщение об ошибке).

Структуры:

# AoAM – строка, содержащая информацию о количестве доступной памяти;

SEM – строка, содержащая информацию о количестве расширенной памяти;

CMCB – «шапка» для вывода MCB;

MCB – строка, печатающая содержание MCB;

ERR\_MES – строка, печатающая сообщение об ошибке выделения памяти.

# Последовательность действий, выполняемых утилитой

В ходе лабораторной работы было разработано 4 программы схожего действия. Программа 1 (результат работы представлен на рис. 1) печатает в консоль информацию о доступной и расширенной памяти, цепочку MCB.

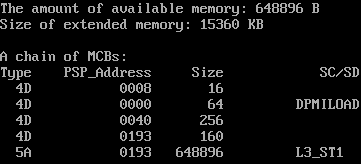


Рис. 1 – Результат работы программы 1

Программа 2 (результат работы представлен на рис. 2), помимо выполнения задач программы 1, освобождает неиспользуемую программой память.

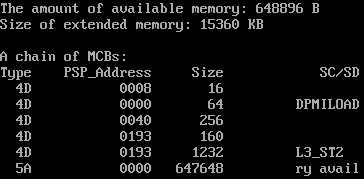


Рис. 2 – Результат работы программы 2

Программа 3 (результат работы представлен на рис. 3), помимо выполнения задач программы 2, выделяет 64КБ (после освобождения неиспользуемой памяти).

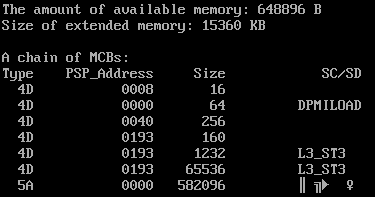


Рис. 3 – Результат работы программы 3

Программа 4 (результат работы представлен на рис. 4), помимо выполнения задач программы 2, выделяет 64КБ (до освобождения неиспользуемой памяти).

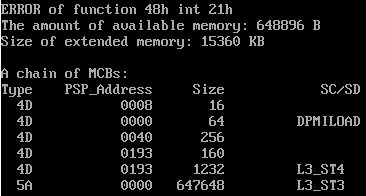


Рис. 4 – Результат работы программы 4

# Результаты исследования

1. Доступный объём памяти – объём оперативной памяти, который ОС даёт программе в личное пользование. Запущенной программе выделяется весь объём свободной памяти.
2. MCB блок программы помещается в конец списка подобных блоков.
3. Программа 1 занимает 648896 байт.

Программа 2 после освобождения памяти занимает 1232 байт.

Программа 3 после освобождения и последующего выделения памяти занимает 1232+65536 = 66768 байт.

Программа 4 после запроса памяти и последующего освобождения памяти занимает 1232 байт (память выделена не была).

# Вывод

В ходе работы была рассмотрена нестраничная память и способ управления динамическими разделами. Были рассмотрены список занятых и свободных участков память и функции ядра, обеспечивающие управление основной памятью, просматривающих и преобразующих этот список.

Так же было выявлено, что ОС не может выделить программе память, если вся память уже «роздана» (даже если большая её часть уже отдана запрашивающей программе).