

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЁТ
по лабораторной работе №3
по дисциплине «Операционные системы»
Тема: Исследование организации управления основной памятью

Студентка гр. 8381
Преподаватель

Звегинцева Е.Н.
Ефремов М.А.

Санкт-Петербург
2020

Цель работы

Исследовать структуры данных и работу функций управления памятью ядра операционной системы.

Основные теоретические положения.

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 на рис. 1.

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

Рисунок 1 — Структура MCB

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.

Ход выполнения работы

Написана программа, которая выводит следующие сведения:

1. Объем основной памяти. Для его получения используется то, что функция AH=4Ah (*Изменить размер блока памяти*) прерывания 21h при слишком большом запрошенном размере блока возвращает в BX общий объем доступной памяти.

Имя функции в коде программы: WRITE_AVAIL_MEM.

2. Объем расширенной памяти. Он считывается из ячеек 0x30–0x31 памяти CMOS.

Имя функции: WRITE_EXTENDED_MEM.

3. Цепочка MCB – блоков управления памятью (функция WRITE_MEM_BLOCKS). Для каждого блока выводится его сегментный адрес (at seg:), сегментный адрес PSP его процесса-владельца (who:), размер блока в параграфах (size:), а также имя блока (name:).

Вместо адреса PSP процесса может MCB может содержать какое-либо специальное значение, обозначающее, например свободный блок памяти (0), ядро MS-DOS (8), либо какую-нибудь конкретную программу (как правило, управляющую «верхней памятью»). Если это обнаруживается, выводится обозначение для данного специального значения (known:)

Имя интерпретируется как нуль-терминированная строка с максимальной длиной 8.

В модификации на шаге 2 была добавлена функция `CHANGE_BLOCK_LENGTH`, вызывающая `int 21h`, `AH=4Ah` для единственного блока программы. Размер блока задаётся в регистре `BX`. В `MAIN` задаётся размер блока 10000_{16} байт, т.е. 1000_{16} параграфов. Таким образом, дальний конец стека попадает в этот блок.

Эта функция проверяет, не произошла ли ошибка, по флагу переноса (`C`, `Carry`). Если после возврата из обработчика прерывания этот флаг поднят, значит функция выводит сообщение об ошибке и возвращает FF_{16} в `AX`.

Функцию, изменяющую размер блока памяти процесса, нельзя вызывать перед `WRITE_AVAIL_MEM`, поскольку последняя использует ошибку, возникающую при выполнении той же функции ядра с заведомо слишком большим аргументом. Эта ошибка приведёт к тому, что блок расширится до максимального размера (который и хочет узнать эта функция).

В модификациях 3 и 4 была добавлена функция `ALLOCATE_BLOCK`. Она использует `int 21h`, `AH=48h` для выделения нового блока памяти с размером, заданным в `BX` (из `MAIN` вызывается с $64 \cdot 400_{16}/10_{16} = 1000_{16}$ параграфов). Эта функция точно так же проверяет, не произошла ли ошибка, но возвращает, как и обработчик функции `48h` прерывания сегментный адрес нового блока памяти, либо 0 в случае ошибки.

```
47672 + 441090 Bytes symbol space free

0 Warning Errors
0 Severe Errors

D:\LAB3\LAB3-1>link lab3,;

Microsoft (R) Overlay Linker Version 3.64
Copyright (C) Microsoft Corp 1983-1988. All rights reserved.

LINK : warning L4021: no stack segment

D:\LAB3\LAB3-1>exe2bin lab3,lab3.com

D:\LAB3\LAB3-1>lab3
Available memory: 648848 bytes
Extended memory: 245760 bytes
((at seg: 016F who: 0008 known: "MS-DOS" size: 16 bytes name: ""))
(at seg: 0171 who: 0000 known: "Free" size: 64 bytes name: "")
(at seg: 0176 who: 0040 size: 256 bytes name: "")
(at seg: 0187 who: 0196 size: 208 bytes name: "")
(at seg: 0195 who: 0196 size: 648848 bytes name: "LAB3"))
D:\LAB3\LAB3-1>
```

Рисунок 2 - Результат выполнения программы с шага 1 в Dosbox

```

47672 + 441092 Bytes symbol space free

    0 Warning Errors
    0 Severe Errors

D:\LAB3\LAB3-2>link lab3,;,;

Microsoft (R) Overlay Linker Version 3.64
Copyright (C) Microsoft Corp 1983-1988. All rights reserved.

LINK : warning L4021: no stack segment

D:\LAB3\LAB3-2>exe2bin lab3,lab3.com

D:\LAB3\LAB3-2>lab3
Available memory: 648848 bytes
Extended memory: 245760 bytes
Changing our memory block length to 65536 bytes...
((at seg: 016F who: 0008 known: "MS-DOS" size: 16 bytes name: "")
 (at seg: 0171 who: 0000 known: "Free" size: 64 bytes name: "")
 (at seg: 0176 who: 0040 size: 256 bytes name: "")
 (at seg: 0187 who: 0196 size: 208 bytes name: "")
 (at seg: 0195 who: 0196 size: 65536 bytes name: "LAB3")
 (at seg: 1196 who: 0000 known: "Free" size: 583296 bytes name: "ength  "))
D:\LAB3\LAB3-2>

```

Рисунок 3 - Результат выполнения программы с шага 2 в Dosbox

В модификации 2 происходит сужение блока, выделенного процессу при запуске и занимающего всю доступную системе основную память. При этом в списке блоков появляется свободный блок, содержащий всю освобождённую при этом память.

```

    0 Warning Errors
    0 Severe Errors

D:\LAB3\LAB3-3>link lab3,;,;

Microsoft (R) Overlay Linker Version 3.64
Copyright (C) Microsoft Corp 1983-1988. All rights reserved.

LINK : warning L4021: no stack segment

D:\LAB3\LAB3-3>exe2bin lab3,lab3.com

D:\LAB3\LAB3-3>lab3
Available memory: 648848 bytes
Extended memory: 245760 bytes
Changing our memory block length to 65536 bytes...
Allocating block of length 65536 bytes...segment address: 1197
((at seg: 016F who: 0008 known: "MS-DOS" size: 16 bytes name: "")
 (at seg: 0171 who: 0000 known: "Free" size: 64 bytes name: "")
 (at seg: 0176 who: 0040 size: 256 bytes name: "")
 (at seg: 0187 who: 0196 size: 208 bytes name: "")
 (at seg: 0195 who: 0196 size: 65536 bytes name: "LAB3")
 (at seg: 1196 who: 0196 size: 65536 bytes name: "LAB3")
 (at seg: 2197 who: 0000 known: "Free" size: 517744 bytes name: "+"))
D:\LAB3\LAB3-3>

```

Рисунок 4 - Результат выполнения программы с шага 3 в Dosbox

В модификации 3 вначале изменяется размер основного блока, а затем выделяется новый. При этом DOS берёт для него память из появившегося при сужении основного блока свободного блока.

```
D:\LAB3\LAB3-4>masm lab3,;;
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988. All rights reserved.

47672 + 441092 Bytes symbol space free

0 Warning Errors
0 Severe Errors

D:\LAB3\LAB3-4>link lab3,;;

Microsoft (R) Overlay Linker Version 3.64
Copyright (C) Microsoft Corp 1983-1988. All rights reserved.

LINK : warning L4021: no stack segment

D:\LAB3\LAB3-4>exe2bin lab3,lab3.com

D:\LAB3\LAB3-4>lab3
Available memory: 648848 bytes
Extended memory: 245760 bytes
Allocating block of length 65536 bytes...ERROR

D:\LAB3\LAB3-4>_
```

Рисунок 5 - Результат выполнения программы с шага 4 в Dosbox

В модификации 4 мы запрашиваем память под новый блок до сужения основного. Из-за этого программа на шаге 4 всегда завершается с ошибкой: запрашивается дополнительный блок памяти, но в это время её единственному блоку выделена вся доступная основная память в системе.

Контрольные вопросы

1. Что означает „доступный объём памяти“?

Это объём основной памяти (до 640 КБ), который есть в системе.

2. Где MCB блок Вашей программы в списке?

Это все блоки, у которых имя “LAB3”. Кроме того, у них одно и то же значение сегментного адреса PSP владельца, равное адресу сегмента, в который моя программа была загружена (на скриншотах это всегда 0x0196).

3. Какой размер памяти занимает программа в каждом случае?

В первом и четвёртом (во время завершения с ошибкой) случаях – процесс занимает всю память, начиная с сегмента, куда программа была загружена. Это 648848 байт.

Во втором случае, занимается только 65536 байт. В третьем случае, занимается 2 блока по 65536 байт.

Каждый блок, кроме того, занимает 16 байт дополнительно под МСВ. Это значит, что в третьем случае можно прибавить 32 байта, а во всех остальных – по 16.

Вывод

В ходе выполнения лабораторной работы был исследован принцип организации управления памятью в ОС с сегментной организацией памяти.