

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. 0381

Прохоров Б.В.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

Исследование организации управления нестраничной памятью и динамическими разделами.

Постановка задачи.

Шаг 1. Для выполнения лабораторной работы необходимо написать и отладить программный модуль типа .COM, который выбирает и распечатывает следующую информацию:

- 1) Количество доступной памяти.
- 2) Размер расширенной памяти.
- 3) Выводит цепочку блок управления памятью.

Адреса при выводе представляются шестнадцатеричными числами. Объём памяти функциями управления памяти 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. Измените программу таким образом, чтобы она освобождала память, которую она не занимает. Для этого используйте функцию 4Ah прерывания 21h. Повторите эксперимент, запустив модифицированную программу. Сравните выходные данные с результатами, полученными на предыдущем шаге. Сохраните результаты, полученные программой, и включите их в отчёт в виде скриншота.

Шаг 3. Измените программу ещё раз таким образом, чтобы после освобождения памяти, программа запрашивала 64Кб памяти функцией 48h прерывания int 21h. Повторите эксперимент, запустив модифицированную программу. Сравните выходные данные с результатами, полученными на предыдущем шаге. Сохраните результаты, полученные программой, и включите их в отчёт в виде скриншота.

Шаг 4. Измените первоначальный вариант программы, запросив 64Кб памяти функцией 48h прерывания int 21h до освобождения памяти. Обязательно обрабатывайте завершение функции ядра, проверяя флаг CF.

Сохраните результаты, полученные программой, и включите их в отчёт в виде скриншота.

Исходные данные.

За основу был взят предоставленный шаблон, содержащий процедуры:
TETR_TO_HEX, BYTE_TO_HEX_WRD_TO_HEX, BYTE_TO_DEC.

Таблица 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 исключительной 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	<p>“SC” – если участок принадлежит MS DOS, то в нём системный код</p> <p>“SD” – если участок принадлежит MS DOS, то в нём системные данные</p>
-----	---	--

Выполнение работы.

Шаг 1.

Для вывода строк и символов написаны процедуры PUTS и PUTC соответственно.

Подготовлены строки для вывода требуемых сообщений.

Для распечатывания требуемой информации написаны следующие процедуры:

- AVAILABLEMEMORYINFO, выводящая количество доступной памяти.
- EXTENDEDMEMORYINFO, выводящая количество расширенной памяти.
- BLOCKLISTINFO, выводящая содержание MCB-блоков.
- WRD_TO_DEC, переводящая размер памяти в параграфах в байты и добавляет их десятичное представление в строку.

Вывод программы выглядит следующим образом:

```

Copyright (C) Microsoft Corp 1983-1988. All rights reserved.

Run File [LB3_1.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
LINK : warning L4021: no stack segment

F:\>exe2bin lb3_1.exe lb3_1.com

F:\>lb3_1.com

Available memory size: 64 bytes
Extended memory size: 15728640 bytes
MCB Type: 4Dh Address: 016Fh Owner: MS DOS Area size: 16 bytes
SC/SD:
MCB Type: 4Dh Address: 0171h Owner: Free Area size: 64 bytes
SC/SD:
MCB Type: 4Dh Address: 0176h Owner: 0040h Area size: 256 bytes
SC/SD:
MCB Type: 4Dh Address: 0187h Owner: 0192h Area size: 144 bytes
SC/SD:
MCB Type: 5Ah Address: 0191h Owner: 0192h Area size: 648912 bytes
SC/SD: LB3_1

```

Рисунок 1 – Вывод программы при выполнении первого пункта задания

Программа выводит размер доступной памяти равный 64 байт, поскольку она занимает весь объём памяти, но есть свободный блок памяти, идущий после блока MS DOS, который имеет размер 64 байт. В списке MCB содержится 5 блоков. Размер расширенной памяти почти равен 16 МБ.

Шаг 2. В данной версии программа освобождает память, которую не использует, поэтому объём свободной памяти увеличился. В выводе списка MCB появился новый свободный блок.

```

Run File [LB3_2.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
LINK : warning L4021: no stack segment

F:\>exe2bin lb3_2.exe lb3_2.com

F:\>lb3_2.com

Available memory size: 583360 bytes
Extended memory size: 15728640 bytes
MCB Type: 4Dh Address: 016Fh Owner: MS DOS Area size: 16 bytes
SC/SD:
MCB Type: 4Dh Address: 0171h Owner: Free Area size: 64 bytes
SC/SD:
MCB Type: 4Dh Address: 0176h Owner: 0040h Area size: 256 bytes
SC/SD:
MCB Type: 4Dh Address: 0187h Owner: 0192h Area size: 144 bytes
SC/SD:
MCB Type: 4Dh Address: 0191h Owner: 0192h Area size: 65536 bytes
SC/SD: LB3_2
MCB Type: 5Ah Address: 1192h Owner: Free Area size: 583360 bytes
SC/SD: ength

```

Рисунок 2 – Вывод программы при выполнении второго пункта задания

Шаг 3. В данной версии программа после освобождения памяти запрашивает ещё 64 Кб памяти, поэтому в выводе списка MCB появился ещё один блок размером 64 Кб, принадлежащий программе.

```
Libraries [LIB]:
LINK : warning L4021: no stack segment

F:\>exe2bin lb3_3.exe lb3_3.com

F:\>lb3_3.com

Available memory size: 517808 bytes
Extended memory size: 15728640 bytes
MCB Type: 4Dh Address: 016Fh Owner: MS DOS Area size: 16 bytes
SC/SD:
MCB Type: 4Dh Address: 0171h Owner: Free Area size: 64 bytes
SC/SD:
MCB Type: 4Dh Address: 0176h Owner: 0040h Area size: 256 bytes
SC/SD:
MCB Type: 4Dh Address: 0187h Owner: 0192h Area size: 144 bytes
SC/SD:
MCB Type: 4Dh Address: 0191h Owner: 0192h Area size: 65536 bytes
SC/SD: LB3_3
MCB Type: 4Dh Address: 1192h Owner: 0192h Area size: 65536 bytes
SC/SD: LB3_3
MCB Type: 5Ah Address: 2193h Owner: Free Area size: 517808 bytes
SC/SD:
```

Рисунок 3 – Вывод программы при выполнении третьего пункта задания

Шаг 4. В данной версии программа запрашивает дополнительную память, предварительно не освободив её, поэтому функция 48h устанавливает флаг CF, который говорит о том, что невозможно выделить память. Программа выводит сообщение о том, что памяти недостаточно.

```
Copyright (C) Microsoft Corp 1983-1988. All rights reserved.

Run File [LB3_4.EXE]:
List File [NUL.MAP]:
Libraries [LIB]:
LINK : warning L4021: no stack segment

F:\>exe2bin lb3_4.exe lb3_4.com

F:\>lb3_4.com
There is not enough memory
Available memory size: 64 bytes
Extended memory size: 15728640 bytes
MCB Type: 4Dh Address: 016Fh Owner: MS DOS Area size: 16 bytes
SC/SD:
MCB Type: 4Dh Address: 0171h Owner: Free Area size: 64 bytes
SC/SD:
MCB Type: 4Dh Address: 0176h Owner: 0040h Area size: 256 bytes
SC/SD:
MCB Type: 4Dh Address: 0187h Owner: 0192h Area size: 144 bytes
SC/SD:
MCB Type: 5Ah Address: 0191h Owner: 0192h Area size: 648912 bytes
SC/SD: LB3_4
```

Рисунок 4 – Вывод программы при выполнении четвёртого пункта задания

Ответы на вопросы см. в разделе «Вопросы».

Выводы.

Была исследована организация управления нестраничной памятью и динамическими разделами.

ВОПРОСЫ

- 1) Что означает “доступный объём памяти”?

Доступный объём памяти – это размер наибольшего свободного участка памяти, то есть максимальный размер блока, который может запросить программа.

- 2) Где МСВ блок Вашей программы в списке?

В первом случае блок программы находится в конце списка, см. рис. 1.

Во втором случае блок программы предпоследний в списке, после него идёт свободный блок, см. рис. 2.

В третьем случае программе принадлежат 5 и 6 блоки, последний в списке свободный, см. рис. 3.

- 3) Какой размер памяти занимает программа в каждом случае?

В первом случае программа занимает всё доступную память – 648912 байт.

Во втором случае программа занимает 65536 байт.

В третьем случае программе принадлежат 5 и 6 блоки, следовательно, она занимает $65536 + 65536 = 131072$ байт.

Четвёртый случай аналогичен первому, поскольку невозможно выделить дополнительную память.