

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

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

Студентка гр. 0381

Короткина Е.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

Исследовать организации управления нестраничной памятью и динамическими разделами.

Постановка задачи.

1. Написать и отладить программный модуль типа .COM, который выбирает и распечатывает следующую информацию:

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

2. Изменить программу таким образом, чтобы она освобождала память, которую она не занимает. Сравнить выходные данные с результатами, полученными на предыдущем шаге.

3. Изменить программу еще раз таким образом, чтобы после освобождения памяти, программа запрашивала 64Кб памяти функцией 48H прерывания 21H. Сравните выходные данные с результатами, полученными на предыдущих шагах.

4. Изменить первоначальный вариант программы, запросив 64Кб памяти функцией 48H прерывания 21H до освобождения памяти

Исходные данные.

Для выполнения работы были использованы следующие данные из таблицы «Структура MCB» учебного пособия:

| 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 |

Табл. 1. Исходные данные.

Выполнение работы.

Для вывода сообщений написана процедура WRITEMESSAGE.

В файле lab3_1.asm написан код исходного .COM модуля. Подготовлены строки для вывода требуемых сообщений, обозначающих, какой пункт задания выводится в данный момент.

Написана процедура WRITEMEMORYSIZE, выводящая количество доступной памяти.

Написана процедура WRITEEXTMEMORYSIZE, выводящая размер расширенной памяти.

Написана процедура WRITEBLOCKLIST, выводящая содержание MCB-блоков. В метке printblock информация из блока сохраняется в специально выделенную память BLOCKDATA, чтобы работать с ней было удобнее. Для вывода размера в байтах написана процедура WRD_TO_DEC, которая переводит размер памяти в параграфах в байты и добавляет их десятичное

символьное представление в строку. Написана процедура DETECTBLOCKTYPE, которая определяет владельца блока на основе данных из 2-го и 3-го байтов.

Вывод программы выглядит следующим образом:

```
F:\>lab3.com
Available memory size: 648912 byte
Extended memory size: 245760 byte
MCB || Type: 4Dh | Address: 016Fh | Size: 16      bytes | Owner: MS DOS
      | Other:
MCB || Type: 4Dh | Address: 0171h | Size: 64      bytes | Owner: Free
      | Other:
MCB || Type: 4Dh | Address: 0176h | Size: 256     bytes | Owner: 0040h
      | Other:
MCB || Type: 4Dh | Address: 0187h | Size: 144     bytes | Owner: 0192h
      | Other:
MCB || Type: 5Ah | Address: 0191h | Size: 648912 bytes | Owner: 0192h
      | Other: LAB3
```

Рисунок 1. Вывод модуля при выполнении первого пункта задания.

Видно, что первый по счёту блок принадлежит MS DOS, последний блок имеет размер, равный размеру всей доступной памяти, это блок написанного модуля. Размер расширенной памяти - 240 Кб.

Для выполнения второго пункта задания перед вызовом процедуры, выводящей содержимое блоков памяти, была вызвана функция 4Ah прерывания int 21h, в BX занесен размер памяти, занимаемый программой. Теперь память, которую программа не занимает, освобождается. Как видно на рис.2, добавился еще один свободный блок памяти (шестой по счету) с размером 647664 байта, программа теперь занимает пятый по счету блок размером 1232 байта.

```
F:\>lab3_2.com
Available memory size: 648912 byte
Extended memory size: 245760 byte
MCB || Type: 4Dh | Address: 016Fh | Size: 16      bytes | Owner: MS DOS
      | Other:
MCB || Type: 4Dh | Address: 0171h | Size: 64      bytes | Owner: Free
      | Other:
MCB || Type: 4Dh | Address: 0176h | Size: 256     bytes | Owner: 0040h
      | Other:
MCB || Type: 4Dh | Address: 0187h | Size: 144     bytes | Owner: 0192h
      | Other:
MCB || Type: 4Dh | Address: 0191h | Size: 1232    bytes | Owner: 0192h
      | Other: LAB3_2
MCB || Type: 5Ah | Address: 01DFh | Size: 647664 bytes | Owner: Free
      | Other: XêşH CâJ
```

Рисунок 2. Вывод модуля при освобождении памяти.

Для выполнения третьего пункта задания после функции освобождения памяти была вызвана функция 48h прерывания 21h. В ВХ помещено значение 4096, т.к. 64 Кбайта это 4096 параграфов. Добавился ещё один блок памяти (на рис.3. 6) размеров 64 Кбайт, он был выделен из памяти, освобождённой в предыдущем пункте. Владелец этого блока - исполняемый модуль.

```
F:\>lab3_3.com
Available memory size: 648912 byte
Extended memory size: 245760 byte
MCB || Type: 4Dh | Address: 016Fh | Size: 16      bytes | Owner: MS DOS
      | Other:
MCB || Type: 4Dh | Address: 0171h | Size: 64      bytes | Owner: Free
      | Other:
MCB || Type: 4Dh | Address: 0176h | Size: 256     bytes | Owner: 0040h
      | Other:
MCB || Type: 4Dh | Address: 0187h | Size: 144     bytes | Owner: 0192h
      | Other:
MCB || Type: 4Dh | Address: 0191h | Size: 1248    bytes | Owner: 0192h
      | Other: LAB3_3
MCB || Type: 4Dh | Address: 01E0h | Size: 65536   bytes | Owner: 0192h
      | Other: LAB3_3
MCB || Type: 5Ah | Address: 11E1h | Size: 582096  bytes | Owner: Free
      | Other: r assign
```

Рисунок 3. Вывод модуля при выделении дополнительной памяти после освобождения памяти.

Для выполнения четвертого пункта задания функция 48h прерывания int 21h была вызвана до функции освобождения памяти. Как видно из рисунка 4, это вызвало ошибку: флаг CF был установлен в 1, и было выведено сообщение «Bad memory allocation». Это произошло из-за того, что в момент попытки выделения дополнительной памяти программа и так владеет всей доступной памятью, т.е. выделения дополнительного блока невозможно.

```
F:\>lab3_4.com
Available memory size: 648912 byte
Extended memory size: 245760 byte
Bad memory allocation
MCB || Type: 4Dh | Address: 016Fh | Size: 16      bytes | Owner: MS DOS
      | Other:
MCB || Type: 4Dh | Address: 0171h | Size: 64      bytes | Owner: Free
      | Other:
MCB || Type: 4Dh | Address: 0176h | Size: 256     bytes | Owner: 0040h
      | Other:
MCB || Type: 4Dh | Address: 0187h | Size: 144     bytes | Owner: 0192h
      | Other:
MCB || Type: 5Ah | Address: 0191h | Size: 648912  bytes | Owner: 0192h
      | Other: LAB3_4
```

Рисунок 3. Вывод модуля при выделении дополнительной памяти до освобождения памяти.

Ответы на контрольные вопросы:

1. Что означает «доступный объем памяти»?

Доступный объем памяти - это свободный участок памяти наибольшего размера, который может быть запрошен программой.

2. Где МСВ блок вашей программы в списке?

На первом и втором скриншотах пятые по счету, на третьем - пятый и шестой. У таких блоков в поле other название программы (LAB3, LAB3_2, LAB3_3).

3. Какой размер памяти занимает программа в каждом случае?

В первом случае - всю доступную память, 648912 байт.

Во втором - ту, что была выделена ей функцией 4Ah, т.е. 1232 байт.

В третьем - ту, что была выделена ей функцией 4Ah, т.е. 1248 байт, и ту, что была дополнительно запрошена, т.е. еще 65536 байт или 64 Кбайта.

Четвёртый случай аналогичен первому.

Вывод.

Была исследована организация управления нестраничной памятью и динамическими разделами.