

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

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

Студентка гр. 8382

Кулачкова М.К.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2020

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.

Ход выполнения работы.

Была реализована программа, которая выбирает и распечатывает количество доступной памяти в байтах, размер расширенной памяти в Кбайтах и цепочку блоков управления памятью (тип, сегментный адрес PSP владельца участка памяти либо специальный флаг, размер участка в байтах и содержимое последних 8 байтов блока посимвольно).

Количество доступной памяти извлекается с помощью функции 4Ah прерывания 21h, которая предназначена для изменения размера существующего блока памяти и возвращает размер наибольшего доступного блока (в параграфах) в регистр ВХ, если запросить заведомо большую память.

Размер расширенной памяти извлекается из ячеек 30h, 31h CMOS.

Адрес первого MCB извлекается из внутренней структуры MS DOS, называемой “List of Lists”. Содержимое полей MCB извлекается в соответствии с их смещением относительно начала MCB и преобразуется к требуемому виду. Адрес следующего MCB получается при добавлении к указателю на текущий блок размера участка памяти, которым он управляет.

Результат выполнения программы представлен на рисунке 1.

```

C:\>LR3_1.COM
Size of accessible memory: 648912 B
Size of expanded memory: 15360 KB
MCB:
    Block type: 4D, PSP address or special flag: 0008
    Size of memory block: 16 B, Last symbols:
MCB:
    Block type: 4D, PSP address or special flag: 0000
    Size of memory block: 64 B, Last symbols:
MCB:
    Block type: 4D, PSP address or special flag: 0040
    Size of memory block: 256 B, Last symbols:
MCB:
    Block type: 4D, PSP address or special flag: 0192
    Size of memory block: 144 B, Last symbols:
MCB:
    Block type: 5A, PSP address or special flag: 0192
    Size of memory block: 648912 B, Last symbols: LR3_1

```

Рисунок 1 - Результат выполнения модуля LR3_1.COM

Затем программа была изменена таким образом, чтобы она освобождала память, которую она не занимает. Для этого была использована функция 4Ah прерывания 21h. Занимаемый программой размер памяти определен при помощи метки, помещенной в конец программы. Результат выполнения программы представлен на рисунке 2.

```

C:\>LR3_2.COM
Size of accessible memory: 648912 B
Size of expanded memory: 15360 KB
MCB:
    Block type: 4D, PSP address or special flag: 0008
    Size of memory block: 16 B, Last symbols:
MCB:
    Block type: 4D, PSP address or special flag: 0000
    Size of memory block: 64 B, Last symbols:
MCB:
    Block type: 4D, PSP address or special flag: 0040
    Size of memory block: 256 B, Last symbols:
MCB:
    Block type: 4D, PSP address or special flag: 0192
    Size of memory block: 144 B, Last symbols:
MCB:
    Block type: 4D, PSP address or special flag: 0192
    Size of memory block: 12416 B, Last symbols: LR3_2
MCB:
    Block type: 5A, PSP address or special flag: 0000
    Size of memory block: 636480 B, Last symbols: -86L~*iF

```

Рисунок 2 - Результат выполнения модуля LR3_2.COM

Как видно на рисунках 1 и 2, последний участок памяти из первой программы в модифицированной программе был разделен на два участка, причем первый из них занят самой программой, а второй – свободный.

Программа была изменена еще раз таким образом, чтобы после освобождения памяти, которую программа не занимает, она запрашивала 64Кб

(1000h параграфов) памяти функцией 48h прерывания 21h. Результат выполнения программы представлен на рисунке 3.

На рисунке видно, что был выделен ещё один участок памяти по сравнению с последней модификацией программы, занимающий 64Кб (65536 б).

```
Size of accessible memory: 648912 B
Size of expanded memory: 15360 KB
MCB:
    Block type: 4D, PSP address or special flag: 0008
    Size of memory block: 16 B, Last symbols:
MCB:
    Block type: 4D, PSP address or special flag: 0000
    Size of memory block: 64 B, Last symbols:
MCB:
    Block type: 4D, PSP address or special flag: 0040
    Size of memory block: 256 B, Last symbols:
MCB:
    Block type: 4D, PSP address or special flag: 0192
    Size of memory block: 144 B, Last symbols:
MCB:
    Block type: 4D, PSP address or special flag: 0192
    Size of memory block: 12528 B, Last symbols: LR3_3
MCB:
    Block type: 4D, PSP address or special flag: 0192
    Size of memory block: 65536 B, Last symbols: LR3_3
MCB:
    Block type: 5A, PSP address or special flag: 0000
    Size of memory block: 570816 B, Last symbols: æπ¹èèâ†
```

Рисунок 3 - Результат выполнения модуля LR3_3.COM

Программы была изменена еще раз так, чтобы запрос 64Кб памяти осуществлялся до освобождения памяти. Была добавлена обработка завершения функции путем проверки флага CF – если при выделении памяти произошла ошибка, программа выведет соответствующее сообщение. Результат выполнения программы приведен на рисунке 4.

```
C:\>LR3_4.COM
Size of accessible memory: 648912 B
Memory allocation error
Size of expanded memory: 15360 KB
MCB:
    Block type: 4D, PSP address or special flag: 0008
    Size of memory block: 16 B, Last symbols:
MCB:
    Block type: 4D, PSP address or special flag: 0000
    Size of memory block: 64 B, Last symbols:
MCB:
    Block type: 4D, PSP address or special flag: 0040
    Size of memory block: 256 B, Last symbols:
MCB:
    Block type: 4D, PSP address or special flag: 0192
    Size of memory block: 144 B, Last symbols:
MCB:
    Block type: 4D, PSP address or special flag: 0192
    Size of memory block: 13088 B, Last symbols: LR3_4
MCB:
    Block type: 5A, PSP address or special flag: 0000
    Size of memory block: 635808 B, Last symbols: σ1|ÉUîœâ
```

Рисунок 4 - Результат выполнения модуля LR3_4.COM

На рисунке видно, что память выделена не была. Это связано с тем, что до освобождения памяти программа занимает весь доступный объем памяти.

Ответы на вопросы.

1. Что означает «доступный объем памяти»?

– Доступный объем памяти – это размер памяти, которая может быть использована программой.

2. Где MCB блок Вашей программы в списке?

– MCB программы – это последние 2-3 занятых блока в списке: блоки 4 и 5 в 1-ой, 2-ой и 4-ой программах, блоки 4, 5 и 6 в 3-ей программе. Они содержат указатель на PSP программы, а последний из них (последние два в случае 3-ей программы) также содержит название программы в последних байтах.

3. Какой размер памяти занимает программа в каждом случае?

– В первом случае программа занимает 648912 б, т.е. всю доступную память, во втором – 12416 б, в третьем – 12528 б + 64Кб, которые были выделены по запросу, в четвертом – 13088 б.

Выводы.

Была реализована программа, которая выбирает и распечатывает информацию об организации памяти. Программа была модифицирована несколько раз для лучшего ознакомления с организацией памяти и функциями управления памятью. Проведено сравнение результатов выполнения программ, были получены ответы на контрольные вопросы. В результате выполнения работы была исследована структура блоков управления памятью (MCB), получены знания об организации основной памяти, изучены функции управления памятью ядра операционной системы.