

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. 9382

Савельев И.С.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

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 не страничная 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.

Задание.

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 блоков управления памятью.

Адреса при выводе представляются шестнадцатеричными числами. Объем памяти функциями управления памятью 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. Запустите программу и внимательно оцените результаты. Сохраните результаты, полученные программой, и включите их в отчет в виде скриншота.

Шаг 2. Измените программу таким образом, чтобы она освобождала память, которую она не занимает. Для этого используйте функцию 4Ah прерывания 21h (пример в разделе «Использование функции 4AH»). Повторите эксперимент, запустив модифицированную программу. Сравните выходные

данные с результатами, полученными на предыдущем шаге. Сохраните результаты, полученные программой, и включите их в отчет в виде скриншота.

Шаг 3. Измените программу еще раз таким образом, чтобы после освобождения памяти, программа запрашивала 64Кб памяти функцией 48H прерывания 21H. Повторите эксперимент, запустив модифицированную программу. Сравните выходные данные с результатами, полученными на предыдущих шагах. Сохраните результаты, полученные программой, и включите их в отчет в виде скриншота.

Шаг 4. Измените первоначальный вариант программы, запросив 64Кб памяти функцией 48H прерывания 21H до освобождения памяти. Обязательно обрабатывайте завершение функций ядра, проверяя флаг CF. Сохраните результаты, полученные программой, и включите их в отчет в виде скриншота.

Шаг 5. Оцените результаты, полученные на предыдущих шагах. Ответьте на контрольные вопросы и оформите отчет.

Выполнение работы.

В результате выполнения лабораторной работы были написаны 4 версии программы, которая выводит количество доступной памяти, размер расширенной памяти, цепочку блоков управления памятью.

Результат запуска первой версии программы.

```
D:\>I3_1.com
Available memory in bytes: 648912
Expanded memory in kilobytes: 15420
MCB address: 016F PSP address: 0008 Size: 16 SC/SD:
MCB address: 0171 PSP address: 0000 Size: 64 SC/SD:
MCB address: 0176 PSP address: 0040 Size: 256 SC/SD:
MCB address: 0187 PSP address: 0192 Size: 144 SC/SD:
MCB address: 0191 PSP address: 0192 Size: 648912 SC/SD:L3_1
```

Результат запуска второй версии программы, теперь она освобождает память которую не занимает. В результате чего создается новый блок размером 641520 байт.

```

D:\>l3_2.com
Available memory in bytes: 648912
Expanded memory in kilobytes: 15420
MCB address: 016F PSP address: 0008 Size: 16 SC/SD:
MCB address: 0171 PSP address: 0000 Size: 64 SC/SD:
MCB address: 0176 PSP address: 0040 Size: 256 SC/SD:
MCB address: 0187 PSP address: 0192 Size: 144 SC/SD:
MCB address: 0191 PSP address: 0192 Size: 7376 SC/SD:L3_2
MCB address: 035F PSP address: 0000 Size: 641520 SC/SD:¼< t♥~δ

```

Результат запуска третьей версии программы. Теперь она после того как освобождает не используемую память, запрашивает 64КБ. В результате чего появляется еще один блок размером 65536 байт.

```

D:\>l3_3.com
Available memory in bytes: 648912
Expanded memory in kilobytes: 15420
MCB address: 016F PSP address: 0008 Size: 16 SC/SD:
MCB address: 0171 PSP address: 0000 Size: 64 SC/SD:
MCB address: 0176 PSP address: 0040 Size: 256 SC/SD:
MCB address: 0187 PSP address: 0192 Size: 144 SC/SD:
MCB address: 0191 PSP address: 0192 Size: 7376 SC/SD:L3_3
MCB address: 035F PSP address: 0192 Size: 65536 SC/SD:L3_3
MCB address: 1360 PSP address: 0000 Size: 575968 SC/SD:

```

Результат запуска четвертой версии программы. Теперь программа запрашивает память, до её освобождения. Что приводит к ошибке, так как вся доступная память занята.

```

D:\>l3_4.com
Available memory in bytes: 648912
Expanded memory in kilobytes: 15420
Error
MCB address: 016F PSP address: 0008 Size: 16 SC/SD:
MCB address: 0171 PSP address: 0000 Size: 64 SC/SD:
MCB address: 0176 PSP address: 0040 Size: 256 SC/SD:
MCB address: 0187 PSP address: 0192 Size: 144 SC/SD:
MCB address: 0191 PSP address: 0192 Size: 7504 SC/SD:L3_4
MCB address: 0367 PSP address: 0000 Size: 641392 SC/SD:♦iM♠èu

```

Вывод.

В процессе выполнения данной лабораторной работы была исследована не страничная память и способы управления динамическими разделами.

Приложение А. Ответы на контрольные вопросы.

1. Что означает "доступный объем памяти"?

Доступный объем памяти - это тот объем, который занимает программа и может использовать.

2. Где МСВ блок вашей программы в списке?

Расположение МСВ блок можно отследить по пункту PSP address: 0192.

3. Какой размер памяти занимает программа в каждом случае?

На 1 шаге это вся доступная память: $648912 + 144 = 649056$ байт

На 2 шаге: $7376 + 144 = 7520$ байт

На 3 шаге: $65\,536 + 7\,376 + 144 = 73056$ байт

На 4 шаге: $7504 + 144 = 7648$ байт