

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

Студентка гр. 0381

Шиняева А.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы

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

Постановка задачи

1 шаг: Необходимо написать и отладить программный модуль типа .COM, который выбирает и распечатывает следующую информацию:

- 1) количество доступной памяти;
- 2) размер расширенной памяти;
- 3) цепочку блоков управления памятью.

2 шаг: Изменить программу таким образом, чтобы она освобождала память, которую она не занимает. Сравнить выходные данные с результатами, полученными предыдущей программой.

3 шаг: Изменить программу ещё раз таким образом, чтобы после освобождения памяти программа запрашивала 64Кб памяти. Сравнить выходные данные с результатами, полученными предыдущими программами.

4 шаг: Изменить первоначальный вариант программы, запросив 64Кб памяти до освобождения память.

Выполнение работы

1 шаг:

Был написан модуль типа .COM, который выбирает и распечатывает следующую информацию: количество доступной памяти, размер расширенной памяти и выводит цепочку блоков управления памятью. Адреса при выводе представляются шестнадцатеричными числами. Объем памяти функциями управления памятью выводится в виде десятичных чисел. Последние 8 байт MSB выводятся как символы.

Были реализованы процедуры:

- *PRINT_STR — печатает строку;
- *PRINT_BYTE — печатает байты;

- *MEMORY_AVAILABLE — определяет количество доступной памяти;
- *MEMORY_EXTENDED — определяет размер расширенной памяти;
- *MCB — выводит цепочки блоков управления памятью

Результат работы первой программы представлен на рисунке 1.

```
C:\>l1.com
Available memory (bytes): 648912
Extended memory (bytes): 245920
MCB table:
Address: 016F PSP address: 0008 Size: 16 SC/SD:
Address: 0171 PSP address: 0000 Size: 64 SC/SD:
Address: 0176 PSP address: 0040 Size: 256 SC/SD:
Address: 0187 PSP address: 0192 Size: 144 SC/SD:
Address: 0191 PSP address: 0192 Size: 648912 SC/SD: L1
```

Рис. 1 - Результат работы первой программы

2 шаг:

Во второй программе была очищена память, которую она не занимает. Результат работы второй программы представлен на рисунке 2. Реализована функция TASK2.

```
C:\>l2.com
Available memory (bytes): 648912
Extended memory (bytes): 245920
MCB table:
Address: 016F PSP address: 0008 Size: 16 SC/SD:
Address: 0171 PSP address: 0000 Size: 64 SC/SD:
Address: 0176 PSP address: 0040 Size: 256 SC/SD:
Address: 0187 PSP address: 0192 Size: 144 SC/SD:
Address: 0191 PSP address: 0192 Size: 6432 SC/SD: L2
Address: 0324 PSP address: 0000 Size: 642464 SC/SD:
```

Рис. 2 - Результат работы второй программы

3 шаг:

Третья программа после очищения памяти выделяет блок на 64Кб. Результат представлен на рисунке 3. Реализована функция TASK3.

```

C:\>l3.com
Available_memory (bytes): 648912
Extended_memory (bytes): 245920
MCB table:
Address: 016F PSP address: 0008 Size: 16 SC/SD:
Address: 0171 PSP address: 0000 Size: 64 SC/SD:
Address: 0176 PSP address: 0040 Size: 256 SC/SD:
Address: 0187 PSP address: 0192 Size: 144 SC/SD:
Address: 0191 PSP address: 0192 Size: 6432 SC/SD: L3
Address: 0324 PSP address: 0192 Size: 65536 SC/SD: L3
Address: 1325 PSP address: 0000 Size: 576912 SC/SD:

```

Рис. 3 - Результат работы третьей программы

4 шаг:

Четвёртая программа запрашивает 64Кб до очищения памяти (рисунок 4).

```

C:\>l4.com
Available_memory (bytes): 648912
Extended_memory (bytes): 245920
Error! Memory cant allocated
MCB table:
Address: 016F PSP address: 0008 Size: 16 SC/SD:
Address: 0171 PSP address: 0000 Size: 64 SC/SD:
Address: 0176 PSP address: 0040 Size: 256 SC/SD:
Address: 0187 PSP address: 0192 Size: 144 SC/SD:
Address: 0191 PSP address: 0192 Size: 6432 SC/SD: L4
Address: 0324 PSP address: 0000 Size: 642464 SC/SD: .ïþ
.ä

```

Рис. 4 - Результат работы четвертой программы

Контрольные вопросы

1. Что означает «доступный объём памяти»?

Доступный объём памяти – это часть оперативной памяти, занимаемая и используемая программой.

2. Где МСВ блок Вашей программы в списке?

МСВ блок первой программы находится на последнем месте списка (рисунок 1), поскольку программа не освобождает память, которую она не занимает.

МСВ блок второй программы находится на предпоследнем месте списка (рисунок 2), поскольку программа освободила память, которую она не занимает, соответственно, на последнем месте блок с освобождённой памятью.

МСВ блок третьей программы располагается сразу в двух предпоследних местах списка (рисунок 3), поскольку программа сперва освободила память, которую она не занимает, а затем выделила новый блок памяти.

МСВ блок четвёртой программы находится на предпоследнем месте списка (рисунок 4), поскольку программа не смогла выделить 64Кб ввиду того, что весь доступный объём памяти занимает сама программа.3. Можно ли в эту область памяти писать?

Программа может модифицировать содержимое этой области памяти, однако не стоит этого делать, так как эта область не отведена программе.

3. Какой размер памяти занимает программа в каждом случае?

В первом случае программа занимает весь доступный объём памяти.

Во втором случае программа занимает только необходимый объём памяти, то есть 6432 байт.

В третьем случае программа занимает необходимый объём памяти и запрошенные 64Кб памяти, то есть 71968.

В четвертом случае программа занимает только необходимый объём памяти, то есть 6432 байт, потому что была выделена память в 64Кб, и после этого сразу же освободили неиспользуемую память.

Вывод

В ходе работы были исследованы структуры данных и работа функций управления памятью ядра операционной системы.