

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

Студент гр. 0381

Березовская В. В.

Преподаватель

Ефремов М. А.

Санкт-Петербург

2022

Цель работы.

Исследование организации управления памятью, рассмотрение нестраничной памяти и способов управления динамическими разделами. Исследование структур данных и работы функций управления памятью ядра операционной системы.

Задание.

Шаг 1. Написать и отладить программный модуль .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 MSB 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.

Шаг 1: Была написана .COM программа на ассемблере, которая выводит требуемую информацию.

```
Available memory:
648912
Extended memory:
245760
MCB type is: 016F PSP address is: 0000 Size is: 16 SC/SD:
MCB type is: 0171 PSP address is: 0000 Size is: 64 SC/SD:
MCB type is: 0176 PSP address is: 0040 Size is: 256 SC/SD:
MCB type is: 0187 PSP address is: 0192 Size is: 144 SC/SD:
MCB type is: 0191 PSP address is: 0192 Size is: 648912 SC/SD: LAB3
```

Рисунок 1 - Пример работы программы

Шаг 2: Программа была изменена таким образом, чтобы неиспользуемая память освобождалась программой.

```
Available memory:
648912
Extended memory:
245760
MCB type is: 016F PSP address is: 0000 Size is: 16 SC/SD:
MCB type is: 0171 PSP address is: 0000 Size is: 64 SC/SD:
MCB type is: 0176 PSP address is: 0040 Size is: 256 SC/SD:
MCB type is: 0187 PSP address is: 0192 Size is: 144 SC/SD:
MCB type is: 0191 PSP address is: 0192 Size is: 6432 SC/SD: LAB3
MCB type is: 0324 PSP address is: 0000 Size is: 642464 SC/SD: .i6p
.AA
```

Шаг 3: Программа была изменена таким образом, чтобы неиспользуемая память освобождалась программой, после чего она запрашивала 64Кб памяти.

```
Available memory:
648912
Extended memory:
245760
MCB type is: 016F PSP address is: 0000 Size is: 16 SC/SD:
MCB type is: 0171 PSP address is: 0000 Size is: 64 SC/SD:
MCB type is: 0176 PSP address is: 0040 Size is: 256 SC/SD:
MCB type is: 0187 PSP address is: 0192 Size is: 144 SC/SD:
MCB type is: 0191 PSP address is: 0192 Size is: 6432 SC/SD: LAB3
MCB type is: 0324 PSP address is: 0192 Size is: 65536 SC/SD: LAB3
MCB type is: 1325 PSP address is: 0000 Size is: 576912 SC/SD:
```

Шаг 4: Программа была изменена таким образом, чтобы запрос 64Кб памяти совершался до освобождения памяти. Также был написан обработчик завершения функции ядра, проверяющий флаг CF.

```

Available memory:
648912
Extended memory:
245760
CAN NOT REQUEST MEMORY
MCB type is: 016F PSP adress is: 0008 Size is: 16 SC/SD:
MCB type is: 0171 PSP adress is: 0000 Size is: 64 SC/SD:
MCB type is: 0176 PSP adress is: 0040 Size is: 256 SC/SD:
MCB type is: 0187 PSP adress is: 0192 Size is: 144 SC/SD:
MCB type is: 0191 PSP adress is: 0192 Size is: 6432 SC/SD: LAB3
MCB type is: 0324 PSP adress is: 0000 Size is: 642464 SC/SD: .i6p
.ñ▲

```

Выводы.

В результате проделанной работы была исследована организация управления памятью. Было рассмотрено устройство нестраничной памяти и способы управления динамическими разделами. Были исследованы структуры данных и работы функций управления памятью ядра операционной системы.

ВОПРОСЫ

1. Что означает “Доступный объем памяти”?

Доступный объем памяти – это область памяти, которая выделяется для работы управляющей программы.

2. Где МСВ блок Вашей программы в списке?

На шагах 1, 2 и 4 МСВ блок находится на 5ой строчке в таблице на скриншотах. На третьем шаге мы вручную запросили еще один блок памяти, так что для него будет 5 и бая строка в таблице на скриншотах.

3. Какой размер памяти занимает программа в каждом случае?

В первом случае программу занимает всю доступную память.

Во втором – необходимый ей объем памяти, т.е. 6432 байта.

В третьем случае – необходимый объем памяти + 64Кб, т.е. 6432 + 65536 байт памяти.

В четвертом случае программа занимает необходимый объём памяти, т.е. 6432 байт, потому что выделить еще 64Кб не удалось.