**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. 6381 |  | Дайнович А.Ю. |
| Преподаватель |  | Губкин А.Ф. |

Санкт-Петербург

2018

**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.

**Сведения о функциях и структурах данных управляющей программы:**

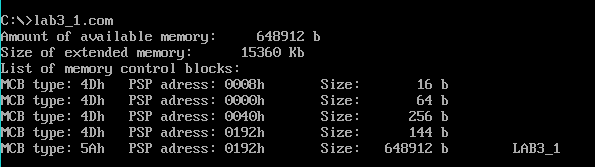
* Функция *tetr\_to\_hex* осуществляет перевод половины байта в символ.
* Функция *byte\_to\_hex* осуществляет перевод байта, помещенного в al, в два символа в шестнадцатеричной системе счисления, помещая результат в ax.
* Функция *wrd\_to\_hex* осуществляет перевод числового значения, помещенного в регистр AX, в символьную строку в шестнадцатеричной системе счисления, помещая результат в регистр di.
* Функция *byte\_to\_dec* осуществляет перевод байта, помещенного в al, в два символа в десятичной системе счисления, помещая результат в si.
* Функция *wrd\_to\_dec* осуществляет перевод слова, помещенного в ax, в последовательность символов в десятичной системе счисления, помещая результат в si. (по аналогии с byte\_to\_dec)
* Функция *print* осуществляет вывод строки на экран.
* Функция *print\_symb* осуществляет вывод символа на экран (используя функцию DOS 02h, прерывания 21).

**Последовательность действий, выполняемых утилитой: программа определяет и выводит на экран:**

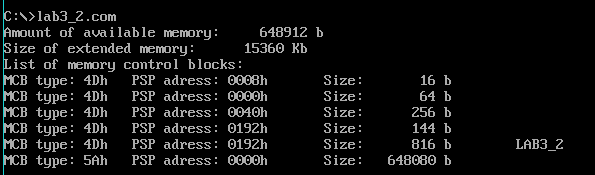
* Количество доступной памяти
* Размер расширенной памяти
* Цепочку блоков управления памятью (Memory Control Block)
* Содержимое области среды (отображается построчно)
* Путь загружаемого модуля

**Результаты работы:**

Результат работы обычной версии программы:

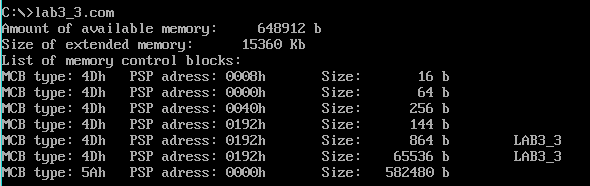


Результат после освобождения неиспользуемой памяти:



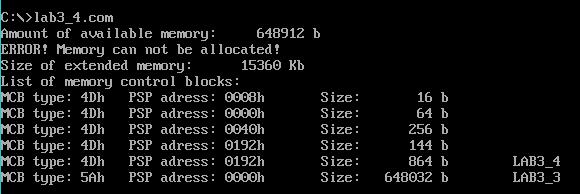
Как видно, размер блока, отведенного программе, уменьшился и образовался свободный участок памяти

Результат после освобождения неиспользуемой памяти и запроса 64Кб:



Как видно, размер участка свободной памяти уменьшился на 64 Кб и образовался участок памяти в 64Кб, запрошенный программой

Попытка выделить запросить 64Кб памяти до освобождения неиспользуемой памяти:



Попытка выделить память при отсутствии свободной памяти завершается неудачей

**Ответы на контрольные вопросы:**

1. Что означает «доступный объем памяти»?

Доступный объем памяти – область основной памяти, выделенная программе (включая префикс программного сегмента)

1. Где MCB блок Вашей программы в списке?

В первой версии программы он расположен в конце списка т.к. программа была последней загружена в память и обладает всем объемом свободной ранее памяти

Во второй версии, MCB блок так же седьмой по списку, но уже не последний. Последним является блок, освобожденной программой памяти

В третьей версии, блок так же седьмой в списке, но после него располагаются блок, памяти в 64Кб, выделенный по запросу программы и после – блок свободной памяти.

В четвертой версии ситуация аналогична ситуации во второй версии, так как запрос выделения памяти завершился неудачей

1. Какой размер памяти занимает программа в каждом списке?

В первом случае программа занимает всю выделенную ей память

Во втором – только объем, занимаемый самой программой

В третьем – объем, занимаемый самой программой и 64 Кб, выделенные ей по требованию

В четвертом, так же как и во втором – только объем, занимаемый самой программой