

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

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

Студент гр. 8382

\_\_\_\_\_

Вербин К.М.

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.

### **Необходимые сведения для составления программы.**

Учёт занятой и свободной памяти ведется при помощи списка блоков управления памятью MCB (Memory Control Block). MCB занимает 16 байт (параграф) и располагается всегда с адреса кратного 16 (адрес сегмента ОП) и находится в адресном пространстве непосредственно перед тем участком памяти, которым он управляет.

MCB имеет следующую структуру:

Смещение	Длина поля (байт)	Содержимое поля
00h	1	тип MCB: 5Ah, если последний в списке, 4Dh, если не последний
01h	2	Сегментный адрес PSP владельца участка памяти, либо 0000h - свободный участок, 0006h - участок принадлежит драйверу OS XMS UMB 0007h - участок является исключенной верхней памятью драйверов

		0008h - участок принадлежит MS DOS FFFAh - участок занят управляющим блоком 386MAX UMB FFFDh - участок заблокирован 386MAX FFFEh - участок принадлежит 386MAX UMB
03h	2	Размер участка в параграфах
05h	3	Зарезервирован
08h	8	"SC" - если участок принадлежит MS DOS, то в нем системный код "SD" - если участок принадлежит MS DOS, то в нем системные данные

По сегментному адресу и размеру участка памяти, контролируемого этим MCB можно определить местоположение следующего MCB в списке.

Адрес первого MCB хранится во внутренней структуре MS DOS, называемой "List of Lists" (список списков). Доступ к указателю на эту структуру можно получить, используя функцию f52h "Get List of Lists" int 21h. В результате выполнения этой функции ES:BX будет указывать на список списков. Слово по адресу ES:[BX-2] и есть адрес самого первого MCB.

Размер расширенной памяти находится в ячейках 30h, 31h CMOS. CMOS это энергонезависимая память, в которой хранится информация о конфигурации ПЭВМ. Объем памяти составляет 64 байта. Размер расширенной памяти в Кбайтах можно определить обращаясь к ячейкам CMOS следующим образом:

```

mov AL,30h ; запись адреса ячейки CMOS
out 70h,AL
in AL,71h ; чтение младшего байта
mov BL,AL ; размера расширенной памяти

```

mov AL,31h ; запись адреса ячейки CMOS

out 70h,AL

in AL,71h ; чтение старшего байта размера расширенной памяти

### **Постановка задачи.**

**Шаг 1.** Необходимо написать и отладить программный модуль типа .COM, 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.

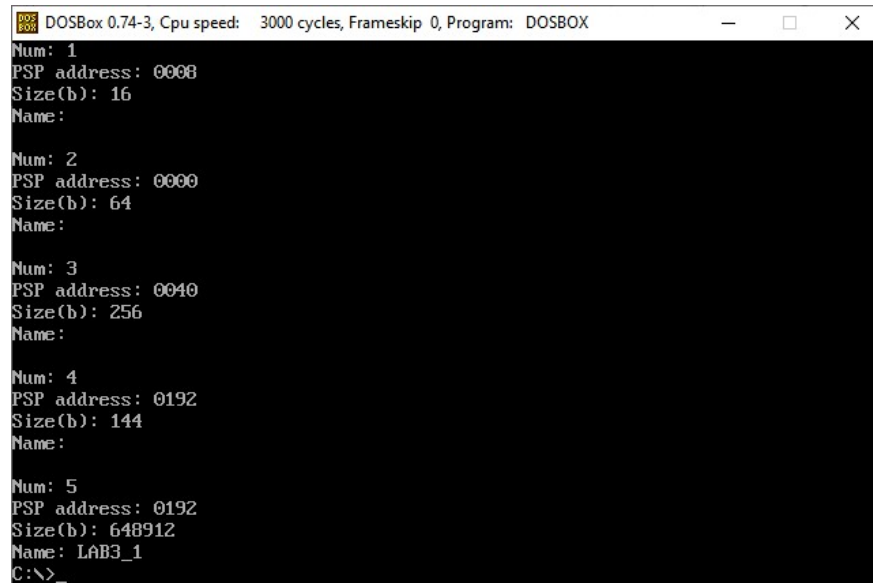
**Шаг 2.** Далее необходимо изменить программу таким образом, чтобы она освобождала память, которую она не занимает. Для этого используйте функцию 4Ah прерывания 21h (пример в разделе «Использование функции 4AH»).

**Шаг 3.** Затем необходимо изменить программу еще раз таким образом, чтобы после освобождения памяти, программа запрашивала 64Кб памяти функцией 48H прерывания 21H.

**Шаг 4.** Далее нужно изменить первоначальный вариант программы, запросив 64Кб памяти функцией 48H прерывания 21H до освобождения памяти.

**Шаг 5.** Оформить отчёт и ответить на контрольные вопросы.

## Ход выполнения работы.



```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX
Num: 1
PSP address: 0008
Size(b): 16
Name:

Num: 2
PSP address: 0000
Size(b): 64
Name:

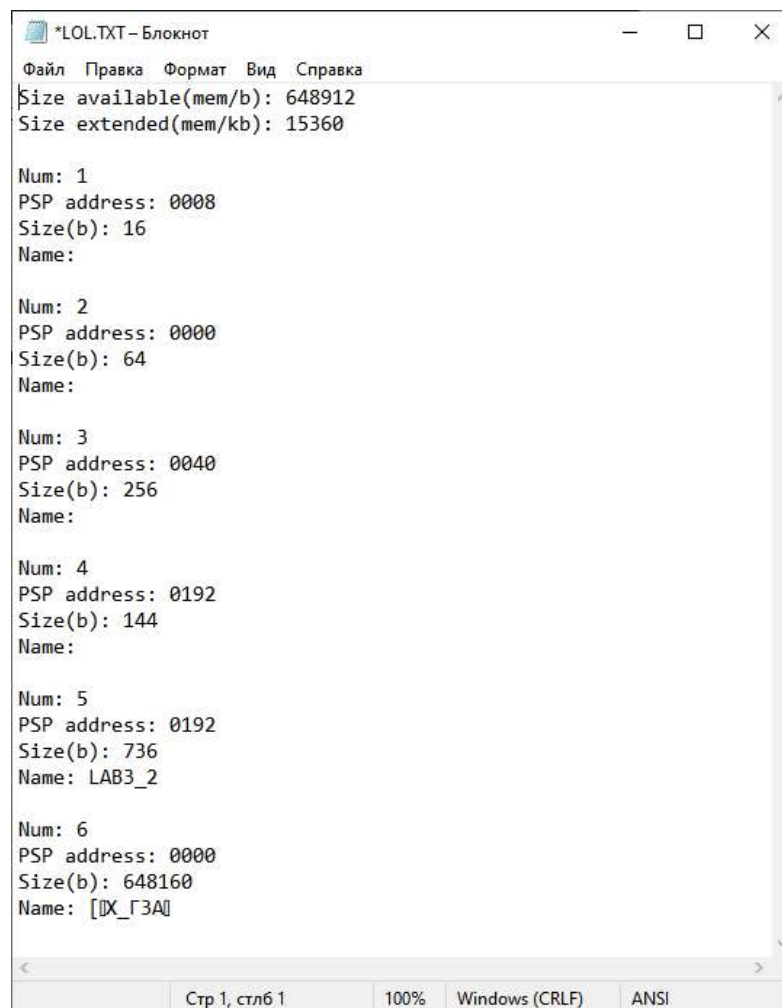
Num: 3
PSP address: 0040
Size(b): 256
Name:

Num: 4
PSP address: 0192
Size(b): 144
Name:

Num: 5
PSP address: 0192
Size(b): 648912
Name: LAB3_1
C:\>_
```

Рисунок 1 – Результат работы lab3\_1.com

Все доступные 648912 байт отдаются программе(рис.1).



```
*LOL.TXT – Блокнот
Файл Правка Формат Вид Справка
Size available(mem/b): 648912
Size extended(mem/kb): 15360

Num: 1
PSP address: 0008
Size(b): 16
Name:

Num: 2
PSP address: 0000
Size(b): 64
Name:

Num: 3
PSP address: 0040
Size(b): 256
Name:

Num: 4
PSP address: 0192
Size(b): 144
Name:

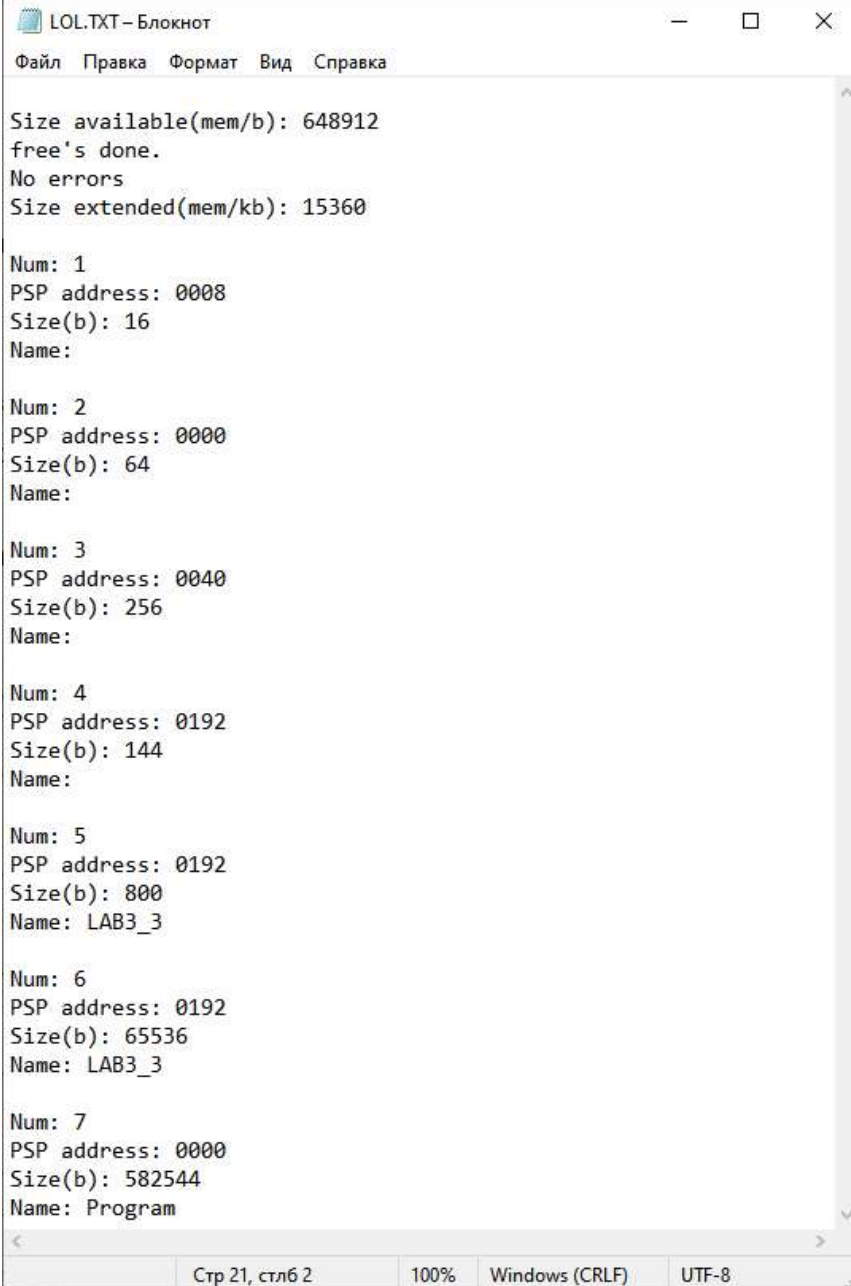
Num: 5
PSP address: 0192
Size(b): 736
Name: LAB3_2

Num: 6
PSP address: 0000
Size(b): 648160
Name: [OX_Г3A]

Стр 1, столб 1 100% Windows (CRLF) ANSI
```

Рисунок 2 – Результат работы lab3\_2.com

Исходный код программы был изменён: теперь программа освобождает не занимаемую ею память. Создается новый блок, который обозначен как свободный участок, размером 648160 байт. Результат предоставлен на рис.2.



```
LOL.TXT - Блокнот
Файл  Правка  Формат  Вид  Справка

Size available(mem/b): 648912
free's done.
No errors
Size extended(mem/kb): 15360

Num: 1
PSP address: 0008
Size(b): 16
Name:

Num: 2
PSP address: 0000
Size(b): 64
Name:

Num: 3
PSP address: 0040
Size(b): 256
Name:

Num: 4
PSP address: 0192
Size(b): 144
Name:

Num: 5
PSP address: 0192
Size(b): 800
Name: LAB3_3

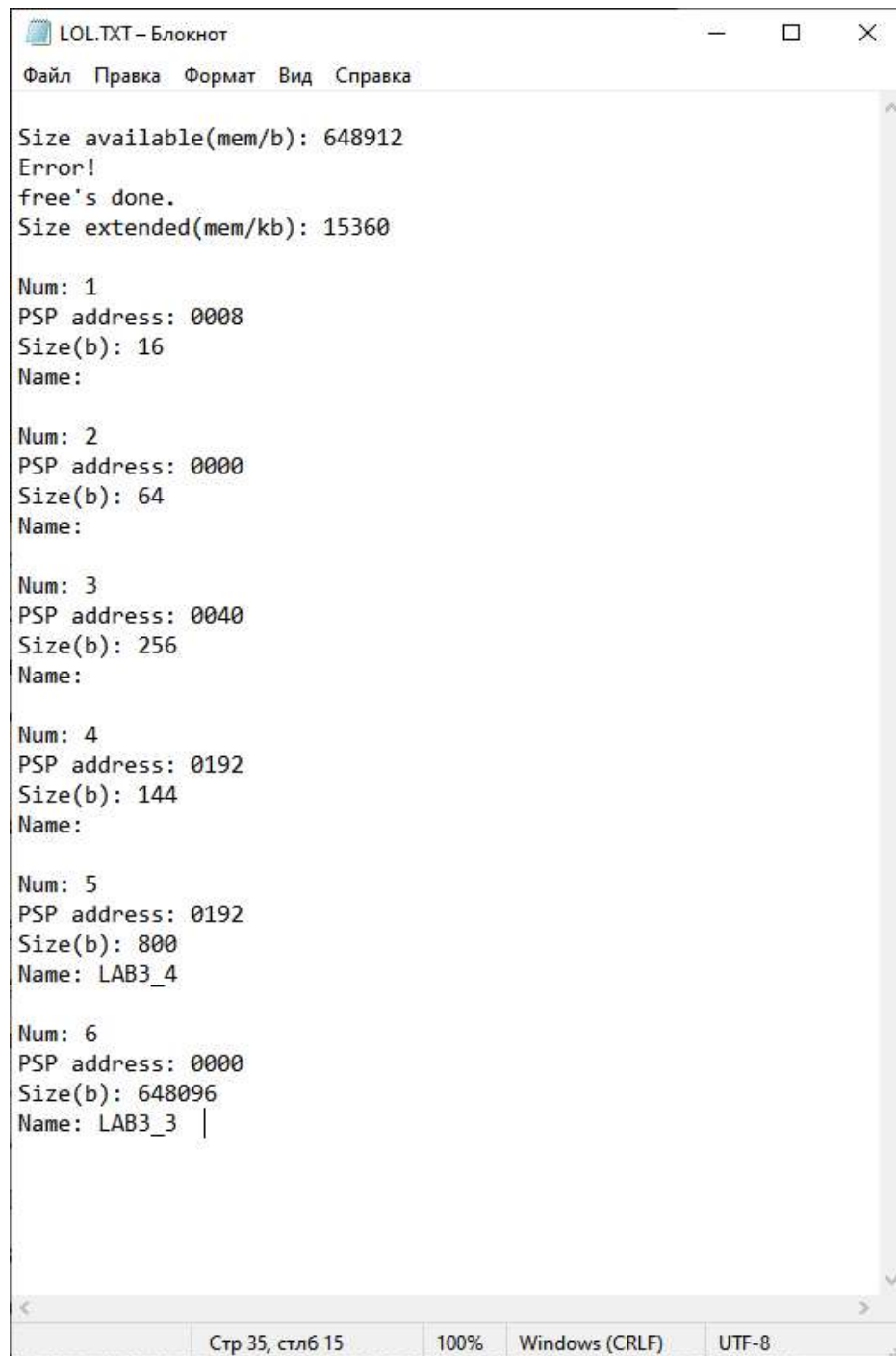
Num: 6
PSP address: 0192
Size(b): 65536
Name: LAB3_3

Num: 7
PSP address: 0000
Size(b): 582544
Name: Program

Стр 21, стлб 2    100%    Windows (CRLF)    UTF-8
```

Рисунок 3 – Результат работы lab3\_3.com

Код программы снова был изменён (рис.3). Вначале происходит то же самое, что и во втором случае (освобождение памяти). Затем программа запрашивает 64 Кбайт (65536 байт) памяти. На свободном участке создается новый блок, который следует за основным блоком программы и занимает 65536 байт.



```
LOL.TXT – Блокнот
Файл  Правка  Формат  Вид  Справка

Size available(mem/b): 648912
Error!
free's done.
Size extended(mem/kb): 15360

Num: 1
PSP address: 0008
Size(b): 16
Name:

Num: 2
PSP address: 0000
Size(b): 64
Name:

Num: 3
PSP address: 0040
Size(b): 256
Name:

Num: 4
PSP address: 0192
Size(b): 144
Name:

Num: 5
PSP address: 0192
Size(b): 800
Name: LAB3_4

Num: 6
PSP address: 0000
Size(b): 648096
Name: LAB3_3 |

Стр 35, стлб 15    100%    Windows (CRLF)    UTF-8
```

Рисунок 4 – Результат работы lab3\_4.com

Код программы снова был изменён (рис.4). Происходит запрос 64 Кбайт до освобождения памяти. Однако выдаётся ошибка(во второй строчке), так как запрос памяти происходит в тот момент, когда вся доступная память занята программой. Затем происходит освобождение памяти, аналогично второму случаю.

## **Ответы на контрольные вопросы.**

### **1) Что означает «доступный объём памяти»?**

Доступный объём памяти – это объём базовой или стандартной памяти (conventional memory), эта память представляет собой "нижние" 640 Кбайт ОЗУ. Для использования базовой памяти не нужны никакие дополнительные драйверы, поскольку операционная система MS DOS изначально создана для работы в адресах 0 - 640 Кбайт

### **2) Где МСВ блок Вашей программы в списке?**

На рисунках 1, 2, 4 - это блоки под номерами 4 и 5.

На рисунке 3 - это блоки под номерами 4, 5, 6.

### **3) Какой размер памяти занимает программа в каждом случае?**

1. 648912 байт.
2.  $144 + 736 = 880$  байт.
3.  $144 + 800 + 65536 = 66480$  байт.
4.  $144 + 800 = 944$  байт.

## **Вывод.**

В ходе работы было проведено исследование структуры данных и работы функций управления памятью ядра операционной системы, а также рассмотрены нестраничная память и способы управления динамическими разделами.