МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждения образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий

Кафедра программной инженерии

Специальность 1-40 01 01 Программное обеспечение информационных технологий

Направление специальности 1-40 01 01 10 Программное обеспечение информационных технологий (программирование интернет-приложений)

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ:

по дисциплине «Операционные системы»

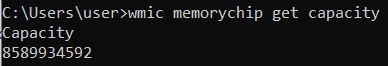
Исполнитель

студент (ка) 3 курса группы 6 Розель Станислав Александрович

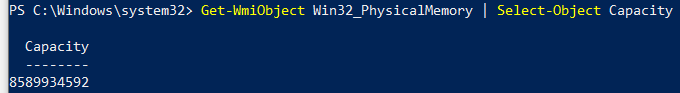
(Ф.И.О.)

Минск 2024

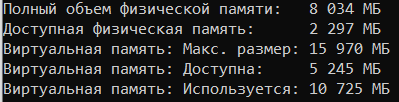
**Задание 1.1**

****

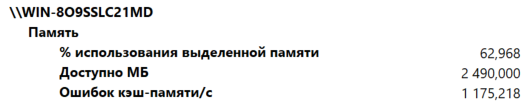
**Задание 1.2**

****

**Задание 1.3**

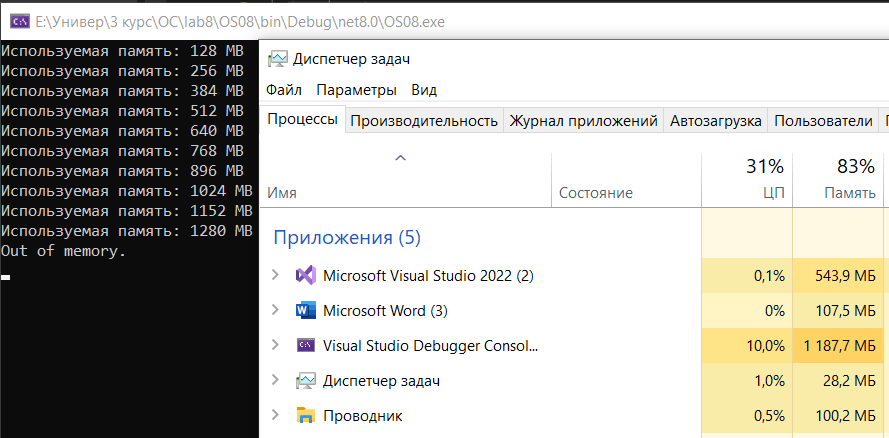
****

**Задание 1.4**

****

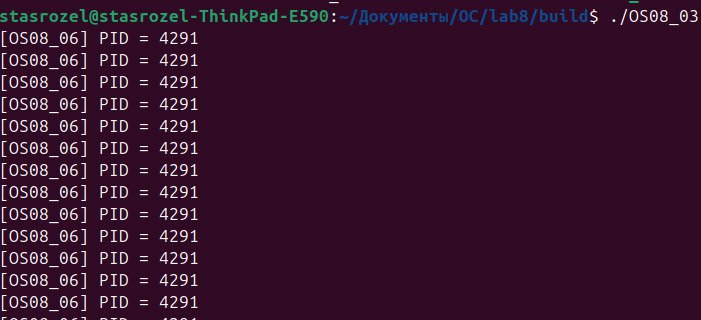
**Задание 2.1**

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Diagnostics;  using System.Threading;  using System.Threading.Tasks;  class Program  {  static void Main(string[] args)  {  List<YesRicoCabum> BUMBUMs = new List<YesRicoCabum>();  while (true)  {  YesRicoCabum BUMBUM = new YesRicoCabum();  BUMBUMs.Add(BUMBUM);  Task.Run(() => BUMBUM.Cabum());  long memoryUsed = GC.GetTotalMemory(true);  Console.WriteLine("Используется памяти: {0} MB", (memoryUsed / (1024 \* 1024)));  Thread.Sleep(1000);  }  }  }  class YesRicoCabum  {  public Int32[] IntArray;  public YesRicoCabum()  {  IntArray = new int[128 \* 1024 \* 1024 / sizeof(int)];  }  public void Cabum()  {  Random rand = new Random();  for (int i = 0; i < IntArray.Length; i++)  {  IntArray[i] = rand.Next();  }  }  } |

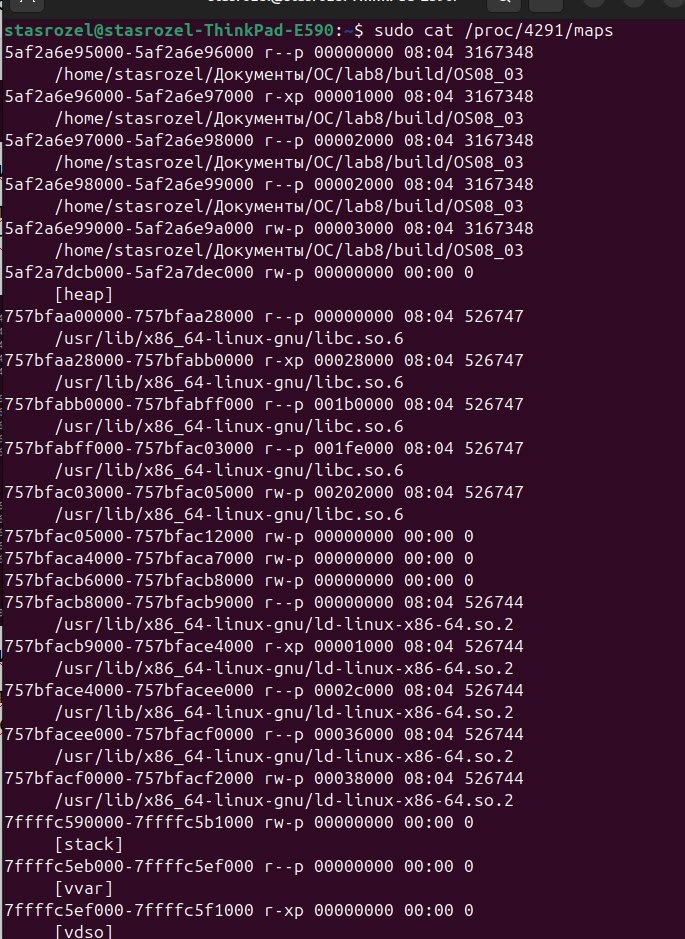
****

**Задание 3**

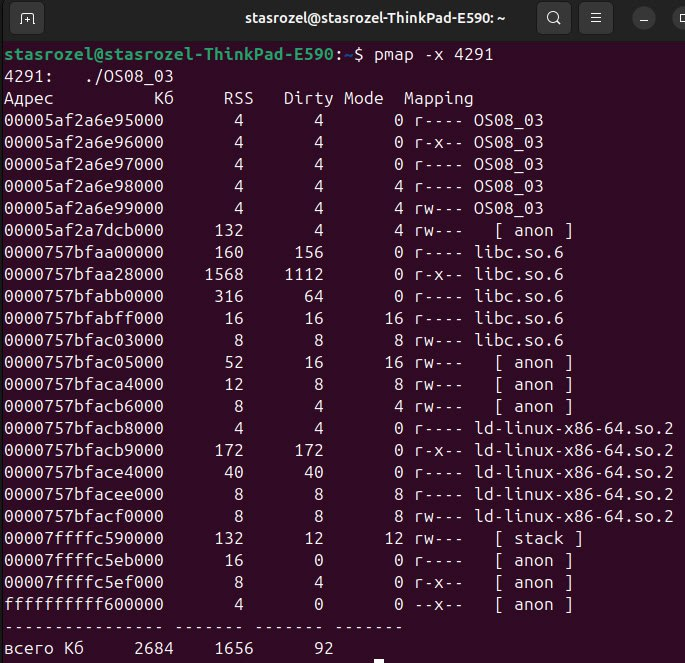
|  |
| --- |
| #include <stdio.h>  #include <unistd.h>  #include <stdlib.h>  int main() {  pid\_t pid = getpid();  nice(0);  for (int i = 0; i < 10000000; i++) {  printf("[OS08\_06] PID = %d\n", pid);  sleep(1);  }  exit(0);  } |

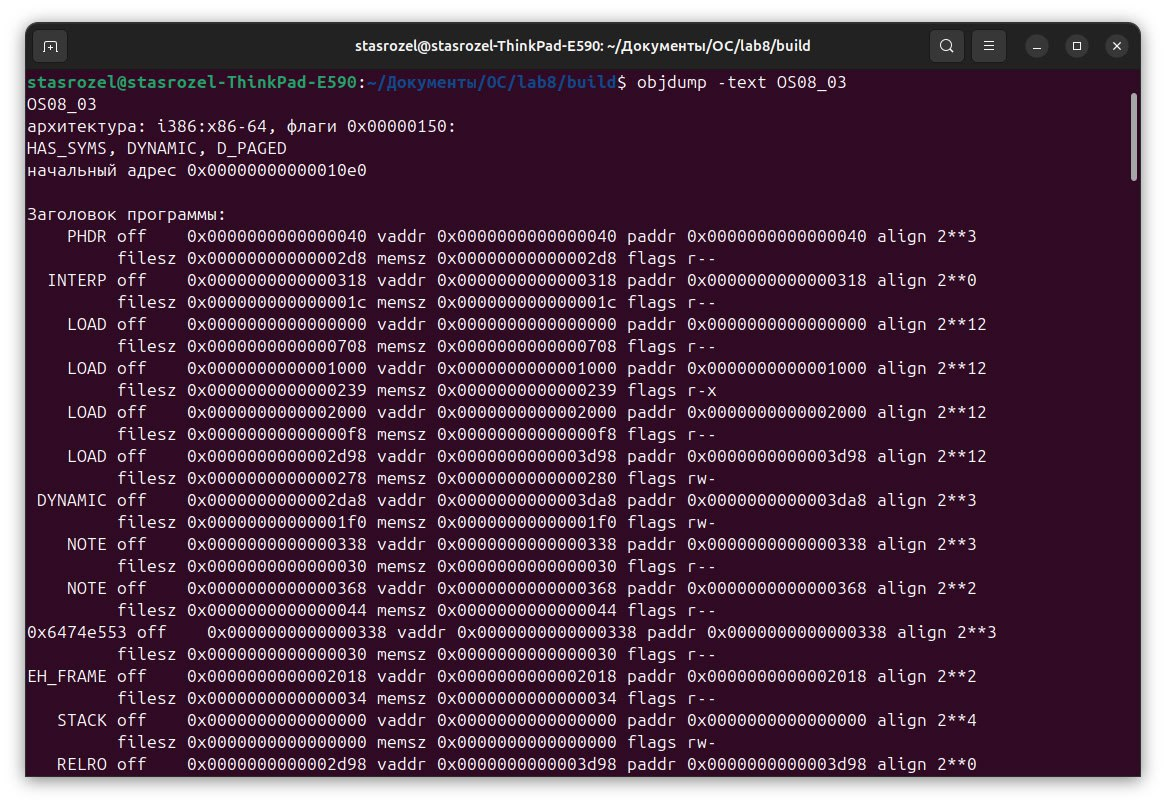


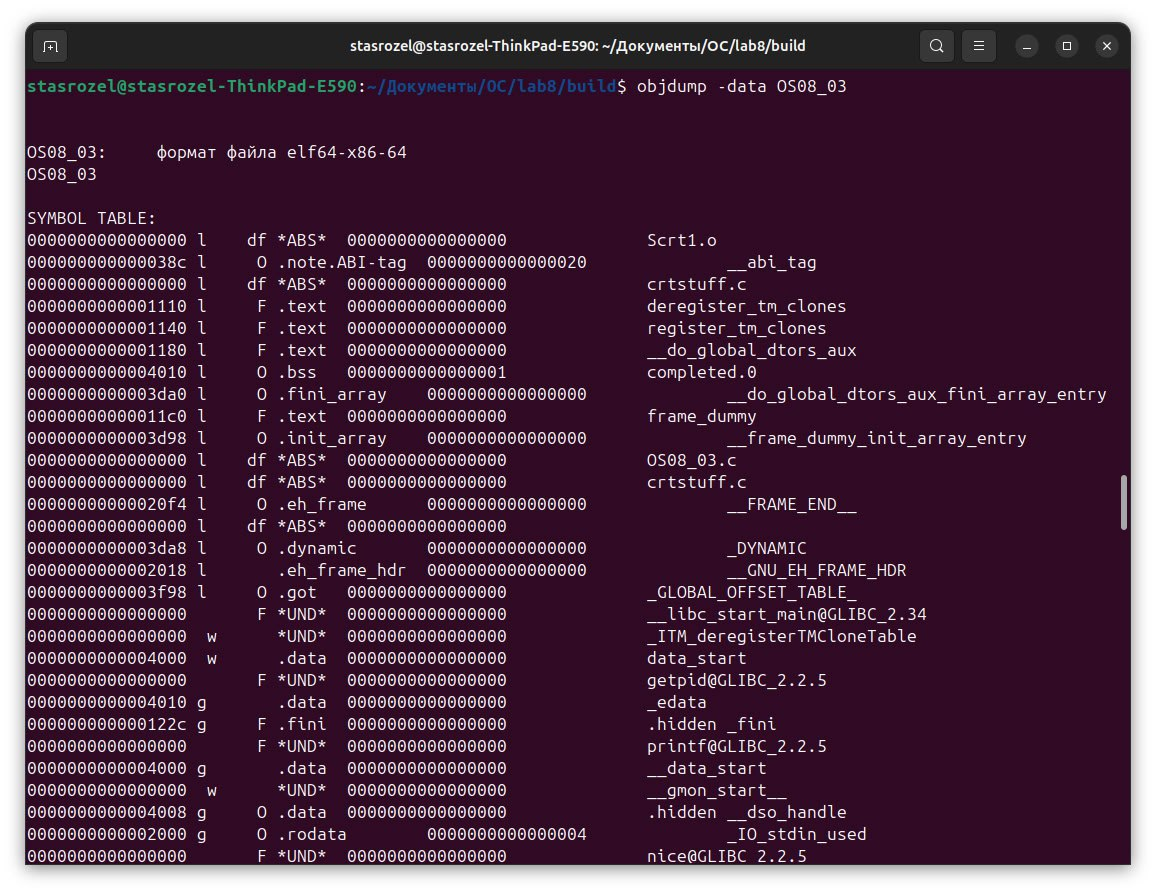
Вывод адресного пространства с помощью функции proc



Структура адресного пространства с помощью pmap







Вариант 3. Укажите содержимое байта по виртуальному адресу 0x001700A5, если регистр CR3 содержит значение 0x13907000.

0х означает что адреса нам представлены в шестнадцатиричной системе исчисления и мы по-быстрому всё переводим в двоичную:

0x001700A5 - 0000 0000 0001 0111 0000 0000 1010 0101

0x13907000 - 0001 0011 1001 0000 0111 0000 0000 0000 - PGD

Дальше мы разбиваем наш адрес на 3 секции

Старшие 10 бит – индекс каталога страниц - 0000 0000 00 – 0х0

Средние 10 бит – индекс таблицы страниц - 01 0111 0000 – 0х170

Младшие 12 бит – смещение по странице - 0000 1010 0101 – 0хА5

Физический адрес записи в каталоге страниц = адрес корневой таблицы страниц + индекс каталога страниц \* 4(размер одной записи)

0x13907000 + 0х0 \* 4 = 0x13907000 – далее ищем запись в нашем дампе памяти, по этому адресу



Следовательно, дальше мы будем использовать адрес 0х13909127, так как нет смещения

По такому же принципу как вычислялся адрес таблицы страниц, вычисляем и адрес страницы:

Физический адрес страницы = Физический адрес таблицы страниц + (Индекс таблицы страниц \* размер страницы)

0х13909000 + 0х170\*4 = 0х13909000 + 0х05С0 = 0х139095С0



**Базовый адрес физической страницы памяти**: 0x26003000

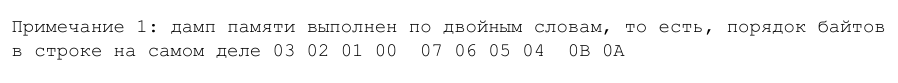
Искомый адрес мы находим по адресу страницы с использованием смещения по этой странице: 0x26003000 + 0хА5

Искомый физический адрес: 0x260030А5

Так как, нам представлены диапазоны адресов и адреса 0x260030А5 в пуле нету, то смотрим диапазон, в который он будет входить, то есть диапазон начинающийся с 0x260030А0. Такой диапазон в дампе присутствует. Рассмотрим его:  


Данный адрес предоставляет нам строку, содержащую 8 двухбайтовых слов с обратным (сначала младшие биты – потом старшие) порядком байтов.

По примеру:



Узнаем что на месте 5, которое указывает на смещение, у нас находятся данные F0

Требуемые данные: 0xF0