

МИНИСТЕРСТВО ОБРАЗОВАНИЯ ОРЕНБУРГСКОЙ ОБЛАСТИ
Государственное автономное профессиональное образовательное
учреждение
**«ОРЕНБУРГСКИЙ КОЛЛЕДЖ ЭКОНОМИКИ И ИНФОРМАТИКИ»
(ГАПОУ ОКЭИ)**

Практическая работа

ОКЭИ 09.02.07 7026 07 У

Тема: *«Изменение выделяемой памяти процессу и анализ памяти с помощью VMMap»*

Выполнил: Гадияев Ислам Ильгамович

Оренбург 2026

Содержание

1 Цель работы 3

2 Скриншот структуры готового проекта..... 3

3 Листинг программы 3

4 Результаты работы программы..... 4

					ОКЭИ 09.02.07 7026 07 У									
Изм.	Лист	№ докум.	Подп.	Дата										
Разраб.	Гадилыев И. И.				Практическая работа						Лит.	Лист	Листов	
											У		2	
											Отделение программирования			

1 Цель работы

Цель работы: разобрать функционал по выделению памяти процессу.

2 Скриншот структуры готового проекта

На рисунке 1 представлен скриншот структуры готового проекта.

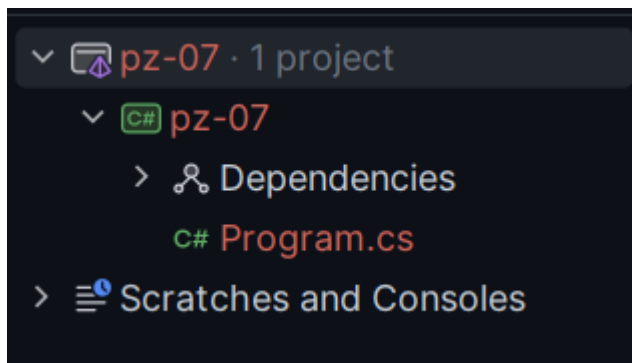


Рисунок 1 – Структура проекта

3 Листинг программы

Листинг файла Program.cs представлен ниже.

```
using System.Runtime.InteropServices;
using System.Threading;

namespace pz_07;

class Program
{
    [DllImport("kernel32.dll")]
    static extern IntPtr HeapCreate(uint flOptions, UIntPtr dwInitialSize, UIntPtr dwMaximumSize);

    [DllImport("kernel32.dll")]
    static extern IntPtr HeapAlloc(IntPtr hHeap, uint dwFlags, UIntPtr dwBytes);

    static void Main(string[] args)
    {
        Console.WriteLine($"{Environment.ProcessId}");

        Thread.Sleep(30000);

        for (int i = 0; i < 3; i++)
        {
            Thread thread = new Thread(Worker);
            thread.IsBackground = true;
            thread.Start(i);
        }

        IntPtr heap = HeapCreate(0, (UIntPtr)(50 * 1024 * 1024), UIntPtr.Zero);
```

```

        Console.WriteLine("Доп куча");

        IntPtr mem = HeapAlloc(
            heap,
            0,
            (UIntPtr)(100 * 1024 * 1024)
        );

        Console.WriteLine("Выделено 100 MB в Heap");

        Console.ReadLine();
    }

    static void Worker(object id)
    {
        Console.WriteLine($"Поток {id} запущен");

        while (true)
        {
            Thread.Sleep(1000);
        }
    }
}

```

4 Результаты работы программы

Результат выполнения программы показан на рисунке 2.

```

"C:\Program Files\JetBrains\JetBrains Rider 2024.2.7\plugins\
\JetBrains.DPA.Runner.exe" --handle=20360 --backend-pid=18988
ct-flags=67108622 --detach-event-name=dpa.detach.20360 C:/Use
ktop/sp-01.04/pz-07/pz-07/bin/Debug/net9.0/pz-07.exe
28004
Поток 0 запущен
Поток 1 запущен
Поток 2 запущен
Доп куча
Выделено 100 MB в Heap

Process finished with exit code -1,073,741,510.

```

Рисунок 2 – Результат работы программы

Как видно в коде, я усыпил процесс на 30 секунд, чтобы вся заскринить, на рисунке 3 результаты до создания дополнительного heap.

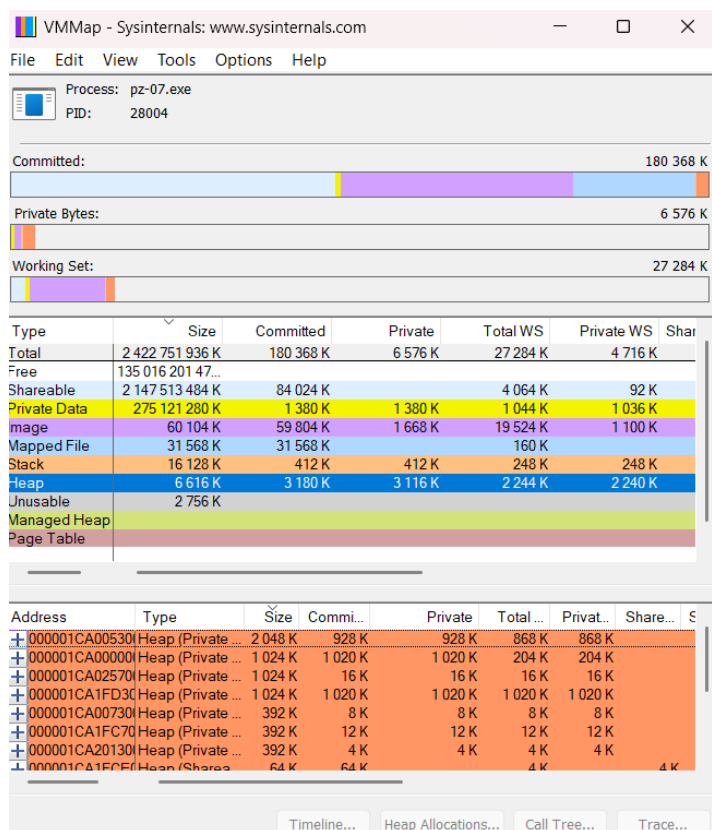


Рисунок 3 – Результат работы программы до heap

С помощью функции HeapCreate был создан дополнительный heap. После выделения памяти через HeapAlloc в VMMap зафиксировано увеличение объёма Heap и Private памяти, как показано на рисунке 4.

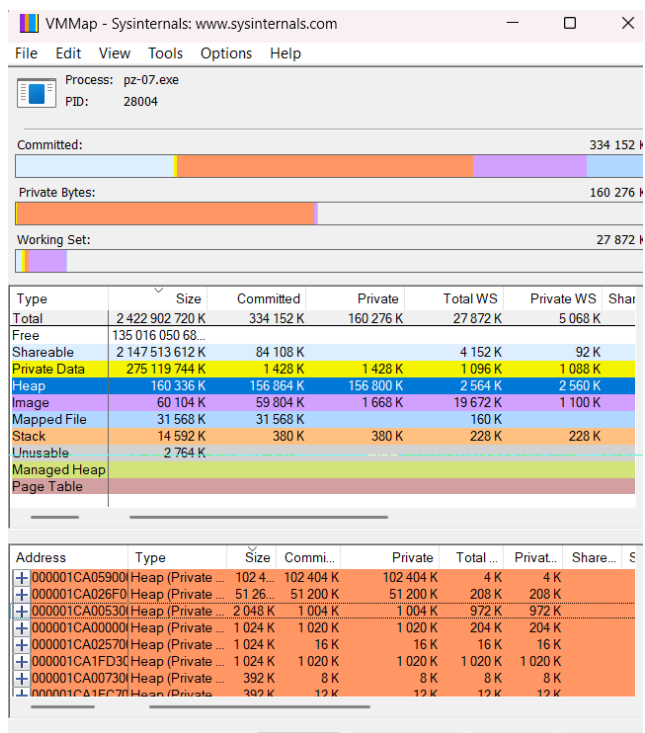


Рисунок 4 – Результат работы программы после HeapCreate

В области Image отображаются исполняемый файл приложения и загруженные DLL, включая clr.dll, ntdll.dll, kernel32.dll и другие системные библиотеки. Все это подтверждает, что Image-память используется для хранения кода и отображённых модулей. Таким образом, большая часть виртуальной памяти процесса зарезервирована, но не закоммичена. Основное потребление Working Set приходится на Image и Heap.

					ОКЭИ 09.02.07 7026 07 У	Лист
Изм.	Лист	№ докум.	Подп.	Дата		6