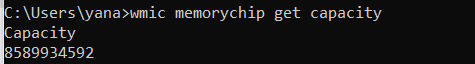
Лабораторная работа 08

Управление памятью

OC, ПОИТ-3

**Задание 01. Windows**

1. Получите с помощью утилиты **wmic** информации об физической оперативной памяти компьютера, поясните эту информацию.



Команда `wmic memorychip get capacity` показывает общий объем оперативной памяти, установленной в системе; это 8 ГБ (8589934592 байт). Оперативная память используется для временного хранения данных и инструкций, что влияет на производительность системы.

wmic: Windows Management Instrumentation Command-line, инструмент для управления и получения информации о системе.

memorychip: Класс, представляющий память в системе.

get: Команда для получения информации.

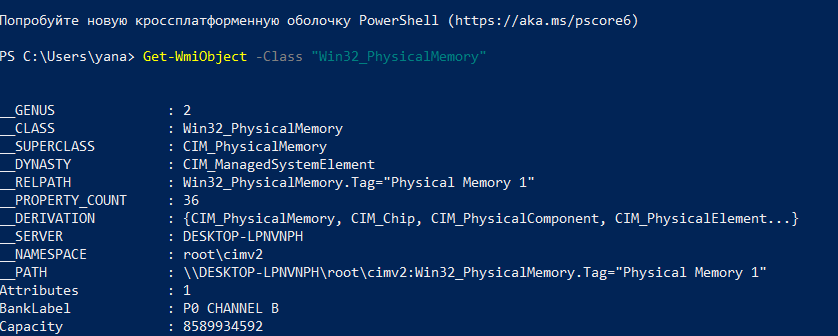
capacity: Свойство, отображающее ёмкость памяти (в байтах).

1. Получите с помощью утилиты **powershell** информации об физической оперативной памяти компьютера, поясните эту информацию.

Get-WmiObject: Команда в PowerShell для получения информации из WMI (Windows Management Instrumentation).

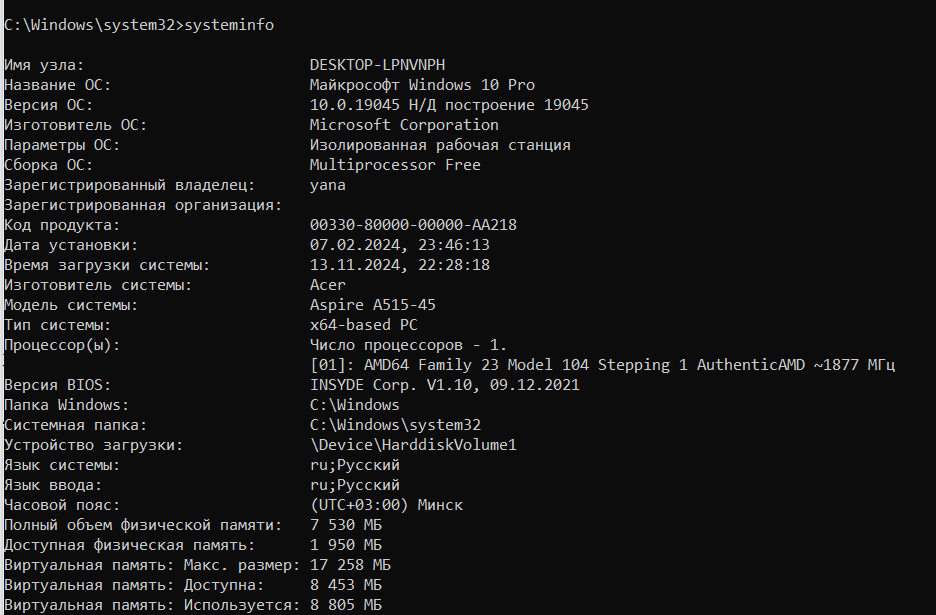
Class: Параметр, указывающий, из какого класса WMI нужно извлекать информацию.

"Win32\_PhysicalMemory": Имя класса, представляющего физическую память (оперативную память) в системе.

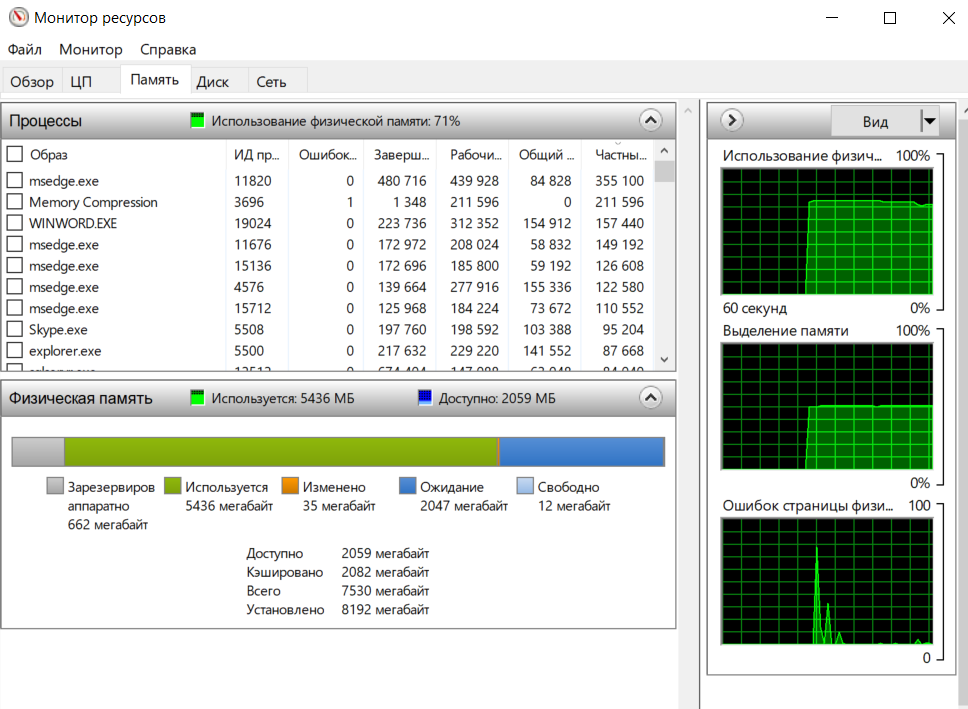


1. Получите с помощью утилиты **systeminfo** информации об оперативной памяти компьютера, поясните эту информацию.

systeminfo показывает, что полный объем физической памяти составляет 7530 МБ (около 7,5 ГБ), что связано с резервированием части памяти.

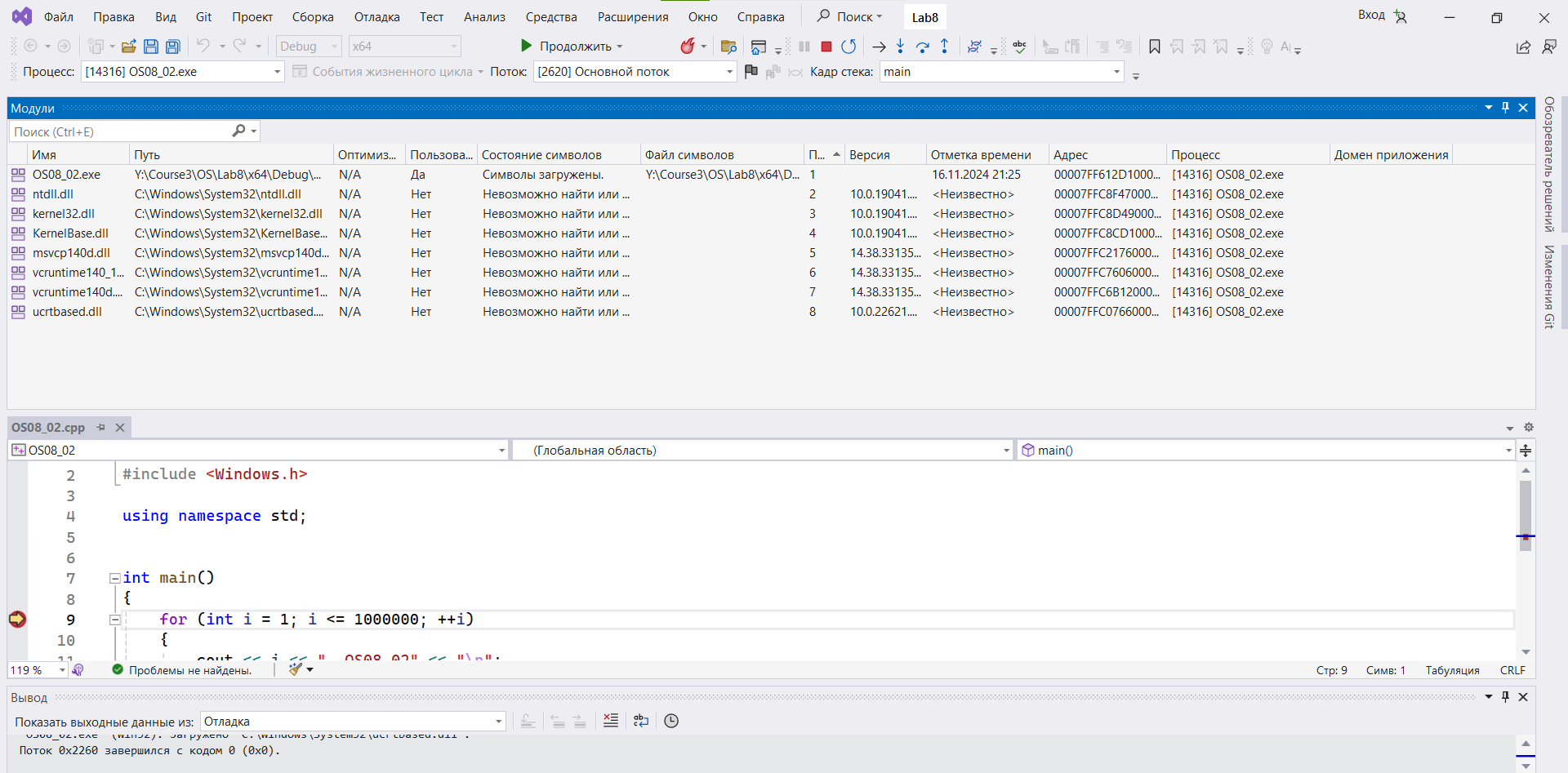


1. Получите с помощью утилиты **performance monitor** информации об оперативной памяти компьютера, поясните эту информацию.

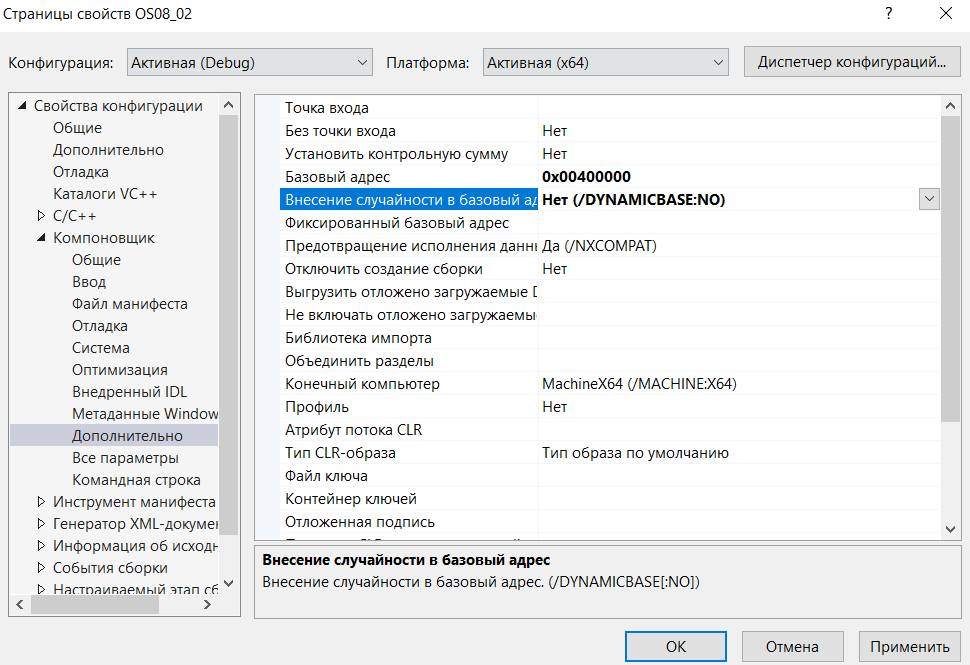
****

**Задание 02.Windows**

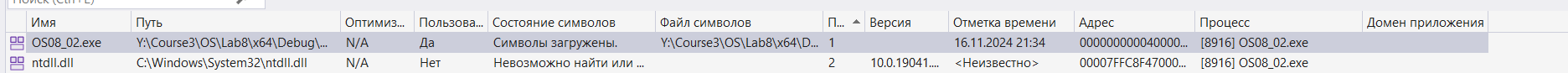
1. Разработайте консольное приложение **OS08\_02**, выполняющее длинный цикл.
2. Продемонстрируйте с помощью отладчика адреса расположения модулей приложения **OS08\_02**.



1. Установите для приложения **OS08\_02** стандартный адрес загрузки в память.



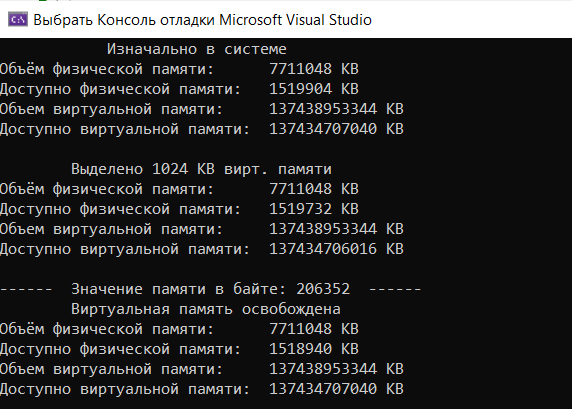
1. Продемонстрируйте с помощью отладчика стандартный адрес расположения модулей приложения **OS08\_02**

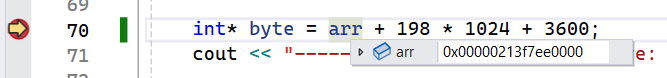


Установка стандартного адреса загрузки в память позволяет контролировать где в адресном пространстве будет загружено приложение, что может быть полезно для оптимизации производительности и предотвращения конфликтов с другими модулями.

**Задание 03.Windows**

1. Разработайте консольное приложение **OS08\_03**, выполняющее получение 256 страниц оперативной памяти.
2. Разместите в этой памяти массив типа **int,** полностью занимающее все 256 страниц.
3. Заполните этот массив нарастающей последовательностью чисел с шагом 1.
4. Запишите 3 первых буквы своей фамилии в 16-ричными числами в кодировке Windows-1251.
5. Найдите в полученной области памяти с помощью отладчика значение в байте, имеющем адрес вычисленный по следующему принципу: номер страницы = число в нулевом байте, смещение в странице = число 12 бит в 1ом и втором байтах.

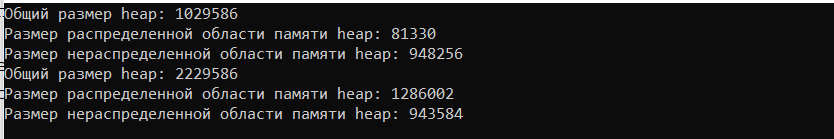






**Задание 04.Windows**

1. Разработайте консольное приложение **OS08\_04,** которое включает функцию **sh**, принимающую 1 параметр: дескриптор (HANDLE) heap.
2. Функция **sh** выводит на консоль, общий размер heap, размеры распределенной и нераспределенных областей памяти heap.
3. Приложение **OS08\_04** размещает в стандартной heap процесса int-массив размерности 300000.
4. Выведите с помощью функции **sh** информацию до размещения массива и после.



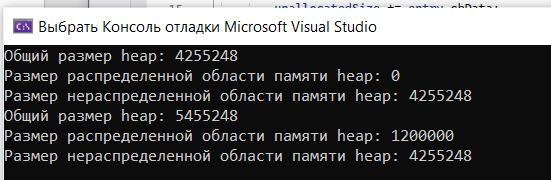
1. Объясните результат.

распределенная память возрастает до 1286002 байт, поскольку массив занял большое пространство.

Объем нераспределенной памяти остаётся почти неизменным (943584 байта), что говорит о том, что свободные блоки памяти не были затронуты выделением нового массива.

**Задание 05.Windows**

1. Разработайте консольное приложение **OS08\_05** аналогичное приложению **OS08\_04,** но использующее пользовательскую heap, которая имеет первоначальный размер 4MB.

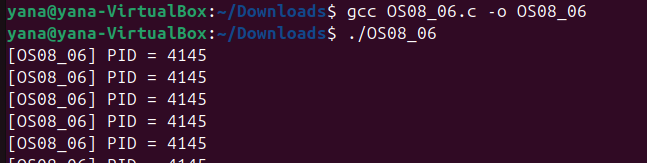


В первом выводе размер распределенной области памяти равен 0, что указывает на то, что в кучу еще не было выделено ни одного блока данных, и вся память считается нераспределенной.

После выделения массива размер heap увеличивается, а размер распределенной области памяти составляет 1200000 байт, что означает, что массив занимает часть памяти и теперь есть выделенные блоки данных.

**Задание 06.Linux**

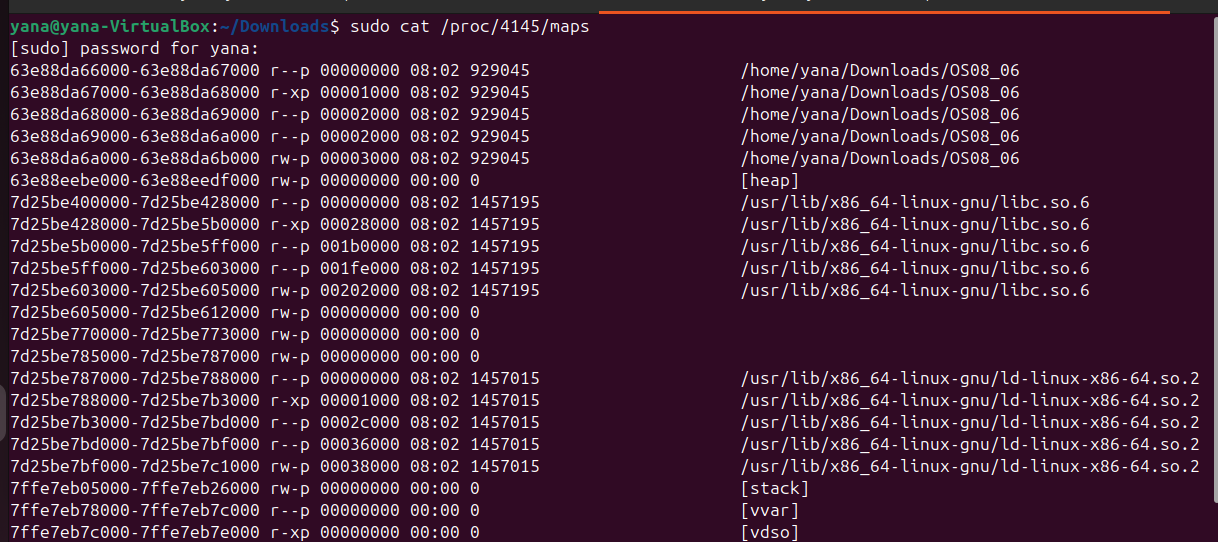
1. Разработайте консольное приложение **OS08\_06**, выполняющее длинный цикл.



1. Продемонстрируйте с помощью файловой системы **/proc** структуру адресного пространства.

sudo: Команда, позволяющая выполнять команды с правами суперпользователя (администратора).

cat: Команда для отображения содержимого файлов в терминале выводит необработанные данные о виртуальных адресах, но не включает информацию о размере и использовании памяти.

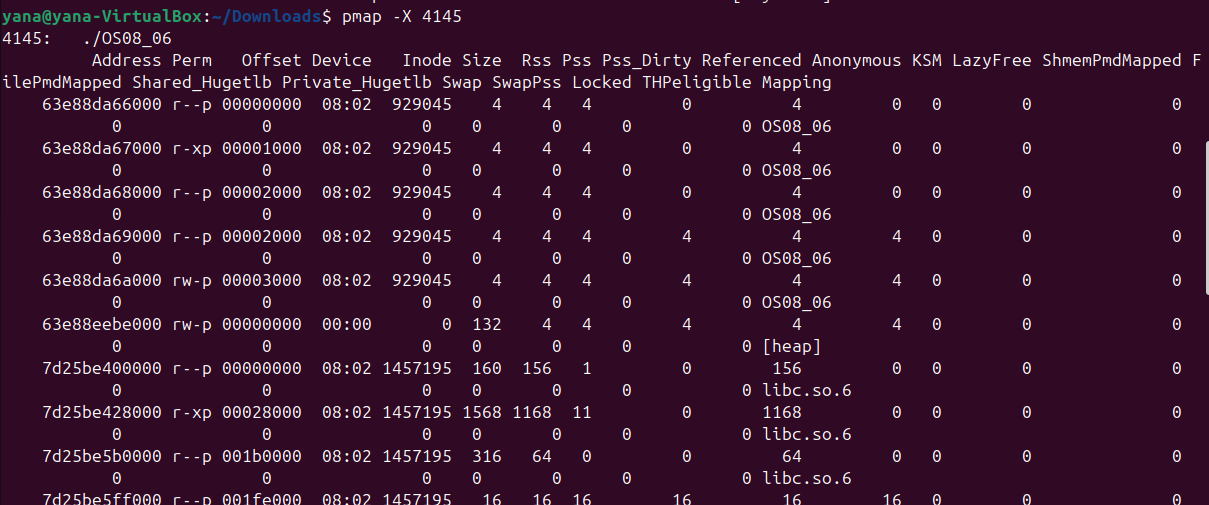


1. Продемонстрируйте с помощью **pmap**  структуру адресного пространства.

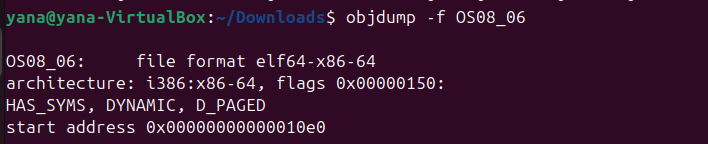
pmap: Утилита в Unix-подобных системах для отображения карты памяти процесса.

-x: Параметр, который выводит детальную информацию о памяти, включая размер, резидентную память, использование и другие параметры.

4145: Идентификатор процесса (PID).



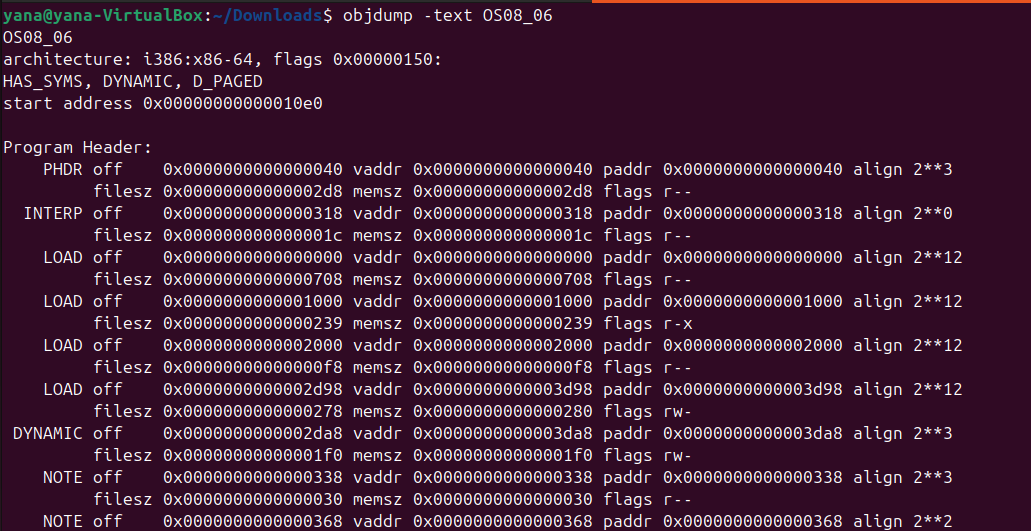
1. Определите с помощью утилиты objdump адрес загрузки main-модуля, секций с кодом, данными, неинициализированными глобальными переменными.



objdump: Утилита для отображения информации о объектах и исполняемых файлах, поддерживающая разные форматы.

-f: Параметр, который указывает на необходимость вывести информацию о заголовке файла, включая архитектуру и тип файла.

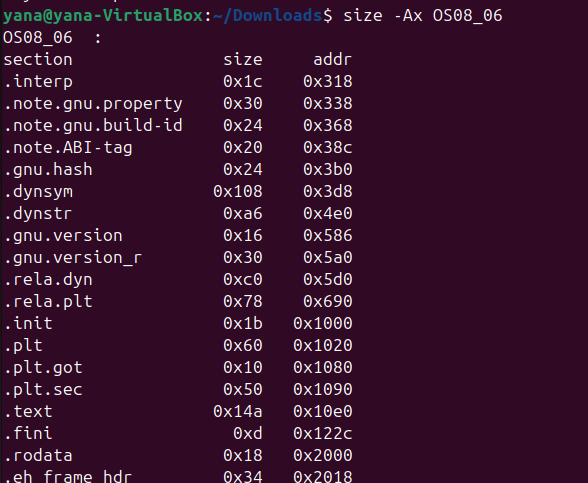
OS08\_06: Имя файла.



-text: Параметр, который указывает на необходимость вывести ассемблерный код (дизассемблирование) из файла.



-data: Параметр, который указывает на необходимость вывести содержимое секций данных файла в двоичном или шестнадцатеричном формате.



size: размер секций в объектных и исполняемых файлах.

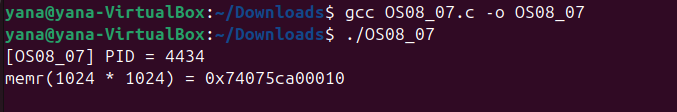
-A: для вывод информации *для всех файлов*.

-x: выводит размеры всех секций в шестнадцатеричном формате.

OS08\_06: Имя файла.

**Задание 07.Linux**

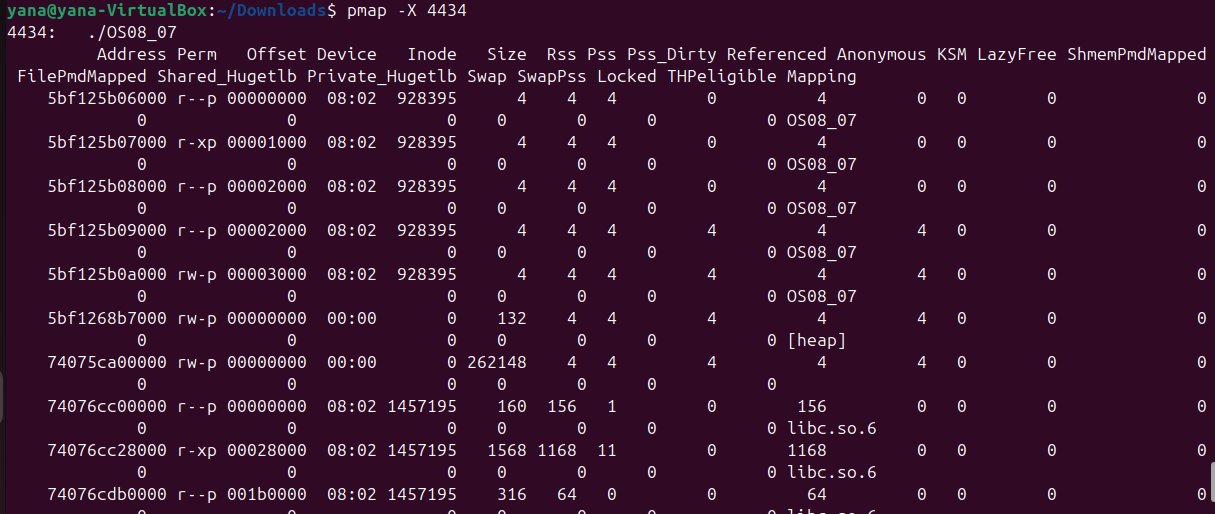
1. Разработайте консольное приложение **OS08\_07**, которое динамически выделяет 256 МБ памяти.
2. В выделенной памяти разместите int-массив максимальной размерности. Проинициализируйте массив последовательными значениями с шагом 1.
3. Выведите на консоль адрес выделенной памяти.



1. После инициализации приложение должно приостановить свое выполнение на длительный интервал времени.
2. С помощью утилиты pmap определите область памяти в которой выделена память.

pmap: отображение карты памяти процесса.

-X: выводит информацию о памяти в формате, включая дополнительные сведения о сегментах и их использовании.



**Задание 08** Ответьте на следующие вопросы

1. Поясните понятие «виртуальная память».
2. Поясните понятие «свопинг».
3. Поясните понятие «страничная память».
4. Поясните понятие MMU(Memory Management Unit).
5. Поясните понятие TLB(Translation Lookaside Buffer).
6. Какая информация содержится в строке таблицы страниц
7. Поясните принцип применения хэш-таблиц.
8. Поясните применение «инвертированной таблицы физических» страничной памяти.
9. Поясните понятие «рабочий набор страниц».
10. Поясните принцип работы алгоритма LRU(Least Recently Used).
11. Windows: поясните назначение сервиса SysMain.
12. Windows: поясните назначение файла hiberfil.sys.abbb
13. Windows: поясните назначение файла pagefile.sys.
14. Windows: поясните назначение файла swapfile.sys.
15. Windows: перечислите области адресного пространства (от младших к старшим адресам) и поясните их назначения.
16. Windows: какой стандартный начальный размер области heap?
17. Windows: каким образом можно изменить начальный размер области памяти heap приложения?
18. Windows: какой стандартный размер области памяти stack?
19. Windows: каким образом можно изменить размер области памяти stack приложения?
20. Windows: поясните назначение функции Windows API: GlobalMemoryStatus.
21. Windows: поясните назначение функции Windows API: VirtualQuery; перечислите значения атрибутов Protect, State и Type.
22. Windows: что такое «рабочее множество»? поясните принцип управления рабочим множеством с помощью OS API.
23. Windows: что означает «страница заблокирована»? с помощью каких функций OS API можно установить блокировку страниц и снять блокировку? Какое максимальное количество страниц можно заблокировать?
24. Windows: что такое «heap»? Что такое «heap процесса»? Что такое «пользовательская heap»? Поясните принцип устройства heap.
25. Linux: перечислите области адресного пространства (от младших к старшим адресам) и поясните их назначения.
26. Linux: в какой части адресного пространства выделяется памяти с помощью функций malloc, calloc?