Лабораторная работа 06

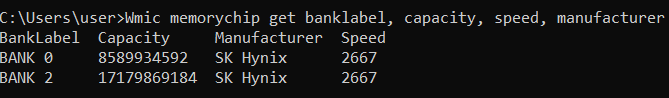
Управление памятью

OC, ПОИТ-3

**Задание 01. Windows**

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

Wmic memorychip get banklabel, capacity, speed, manufacturer



1.BankLabel: Это метка банка памяти. Обычно оперативная память разбита на банки, и каждый банк имеет свою метку. Например, "BANK 0" и "BANK 2" указывают на два банка памяти на канале A и канале B соответственно.

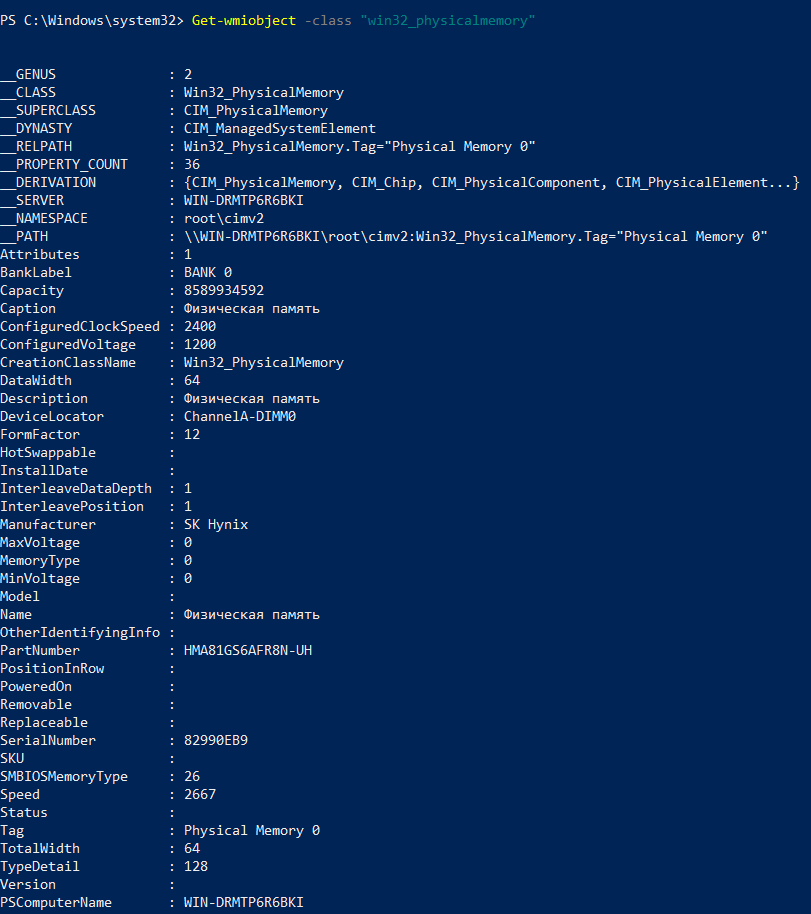
2.Capacity: Это емкость оперативной памяти в байтах. В вашем случае значение "8589934592" байт, что соответствует примерно 8 гигабайтам (1 байт = 8 бит).

3.Manufacturer: Производитель оперативной памяти

4.Speed: Это частота работы оперативной памяти, измеренная в мегагерцах (МГц). Значение "3200" указывает на скорость работы 3200 МГц.

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

Get-wmiobject -class “win32\_physicalmemory”



Genus: Классификация объекта в системе управления.

Class: Тип класса объекта (физическая память).

Superclass: Родительский класс объекта (управляемый элемент CIM).

Dynasty: Иерархия классов, которой принадлежит объект (управляемый элемент CIM).

RelPath: Уникальный путь к объекту (физическая память 0).

Property Count: Количество свойств объекта (36).

Derivation: Происхождение объекта от других классов (физическая память, чип и т. д.).

Server: Имя сервера, на котором расположен объект (WIN-DRMTP6R6BKI).

Namespace: Пространство имен объекта (root\cimv2).

Path: Полный путь к объекту в пространстве имен (Win32\_PhysicalMemory.Tag="Physical Memory 0").

Attributes: Количество атрибутов объекта (1).

BankLabel: Метка банка памяти (BANK 0).

Capacity: Объем памяти в байтах (8589934592).

Caption: Заголовок объекта (физическая память).

ConfiguredClockSpeed: Настроенная тактовая частота (2400).

ConfiguredVoltage: Настроенное напряжение (1200).

CreationClassName: Имя класса создания объекта (Win32\_PhysicalMemory).

DataWidth: Ширина данных (64 бита).

Description: Описание объекта (физическая память).

DeviceLocator: Местоположение устройства (ChannelA-DIMM0).

FormFactor: Форм-фактор (12).

HotSwappable: Возможность горячей замены (пусто).

InstallDate: Дата установки (пусто).

InterleaveDataDepth: Глубина интерливинга данных (1).

InterleavePosition: Позиция интерливинга (1).

Manufacturer: Производитель (SK Hynix).

MaxVoltage: Максимальное напряжение (0).

MemoryType: Тип памяти (0).

MinVoltage: Минимальное напряжение (0).

Model: Модель (пусто).

Name: Имя объекта (физическая память).

OtherIdentifyingInfo: Дополнительная идентификационная информация (пусто).

PartNumber: Номер детали (HMA81GS6AFR8N-UH).

PositionInRow: Позиция в строке (пусто).

PoweredOn: Включено (пусто).

Removable: Возможность удаления (пусто).

Replaceable: Возможность замены (пусто).

SerialNumber: Серийный номер (82990EB9).

SKU: Номер продукта (пусто).

SMBIOSMemoryType: Тип памяти по стандарту SMBIOS (26).

Speed: Скорость памяти (2667).

Status: Состояние объекта (пусто).

Tag: Метка объекта (Physical Memory 0).

TotalWidth: Общая ширина данных (64 бита).

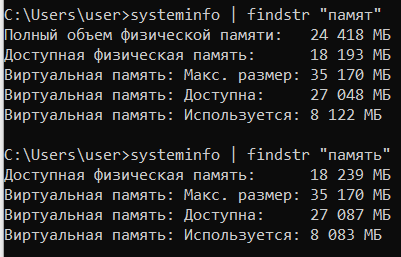
TypeDetail: Дополнительная информация о типе (128).

Version: Версия (пусто).

PSComputerName: Имя компьютера PowerShell (WIN-DRMTP6R6BKI).

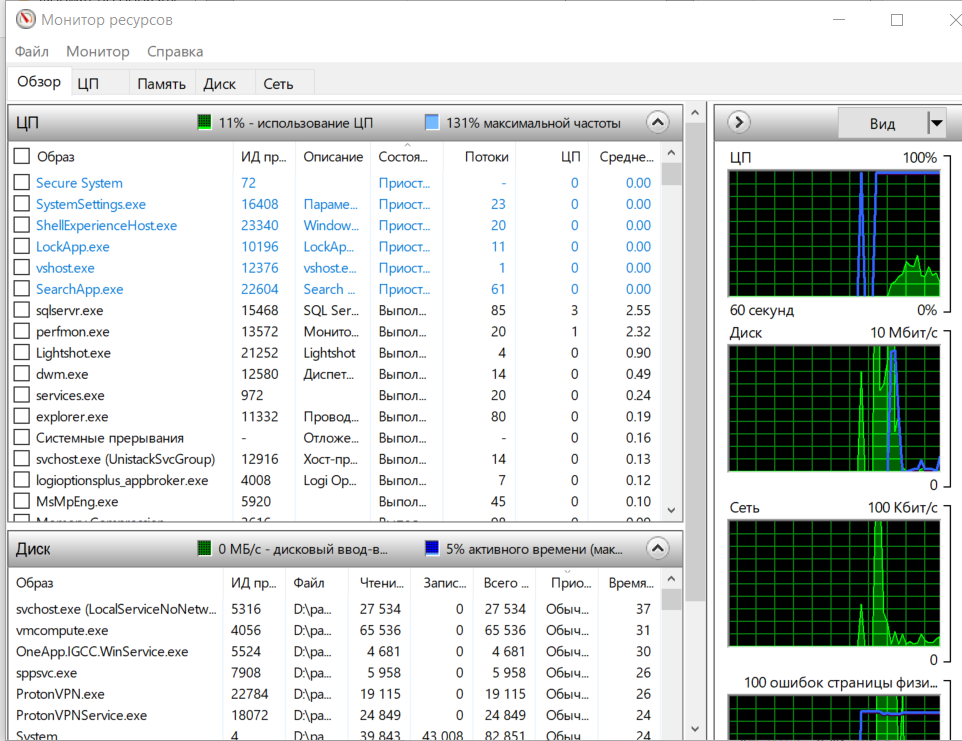
Получите с помощью утилиты **systeminfo** информации об оперативной памяти компьютера, поясните эту информацию.

systeminfo | findstr "память"



Физическая память - это физически установленные модули оперативной памяти, доступные прямо процессору.

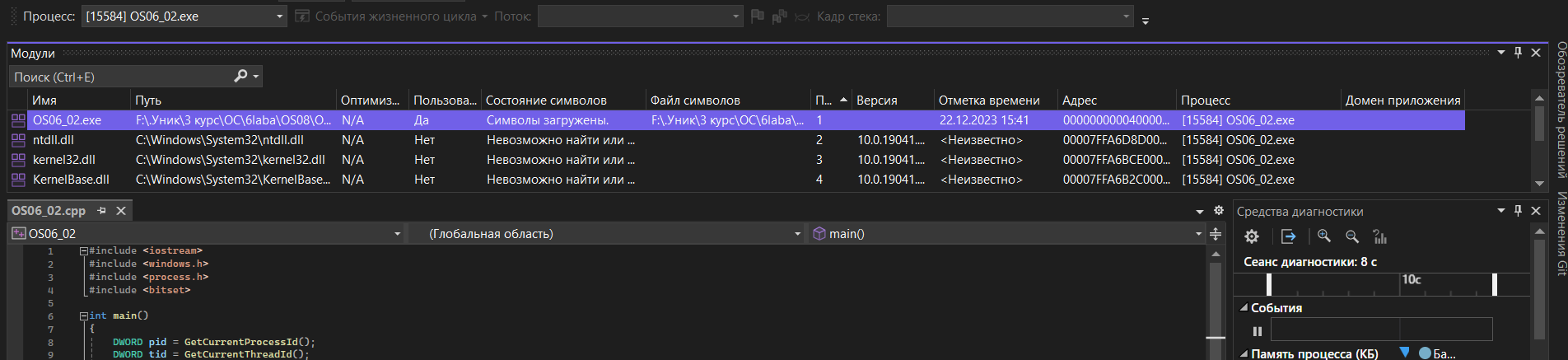
1. Получите с помощью утилиты **performance monitor** информации об оперативной памяти компьютера, поясните эту информацию.



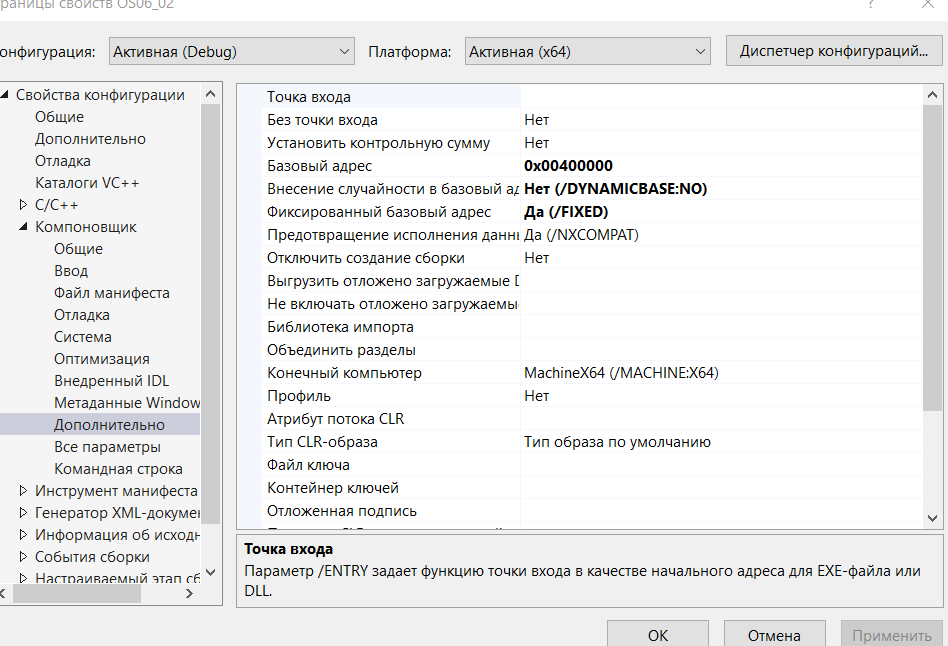
**Задание 02.Windows**

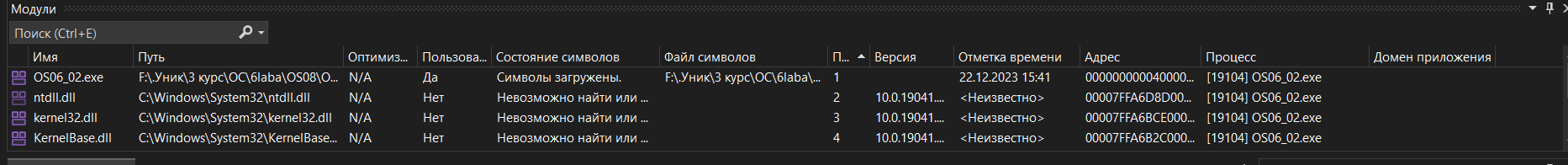
1. Разработайте консольное приложение **OS06\_02**, выполняющее длинный цикл.
2. Продемонстрируйте с помощью отладчика адреса расположения модулей приложения **OS06\_02**.

F5-Отладка-окна-модули



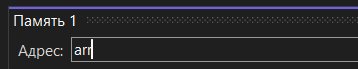
1. Установите для приложения **OS06\_02** стандартный адрес загрузки в память.
2. Продемонстрируйте с помощью отладчика стандартный адрес расположения модулей приложения **OS06\_02**

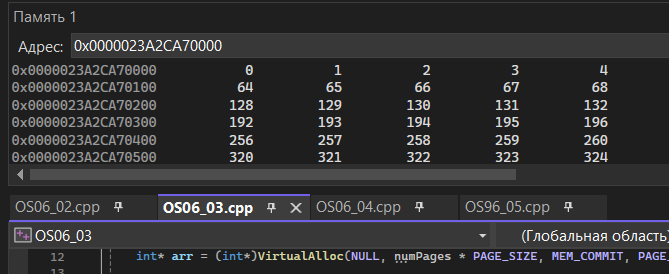


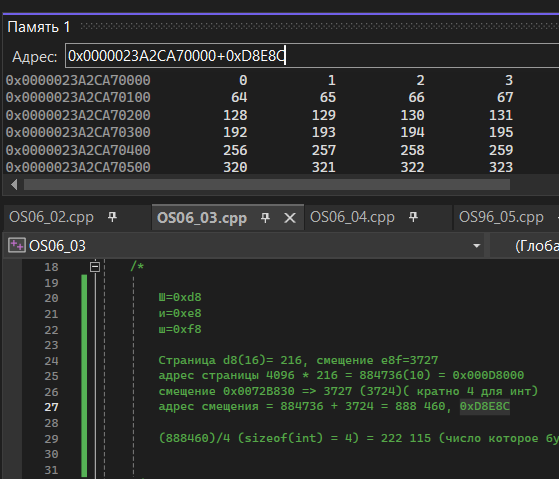


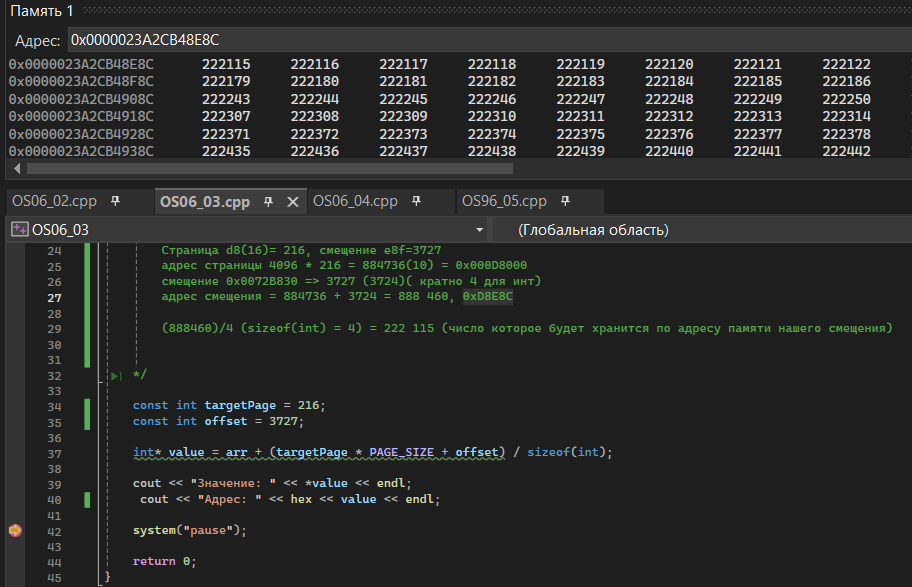
**Задание 03.Windows**

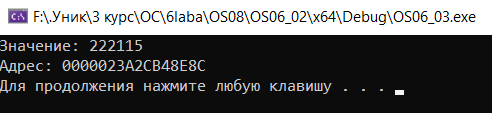
1. Разработайте консольное приложение **OS06\_03**, выполняющее получение 256 страниц оперативной памяти.
2. Разместите в этой памяти массив типа **int,** полностью занимающее все 256 страниц.
3. Заполните этот массив нарастающей последовательностью чисел с шагом 1.
4. Запишите 3 первых буквы своей фамилии 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.





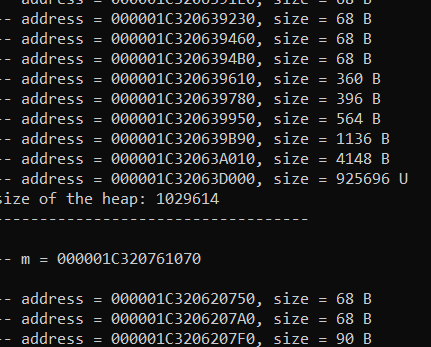


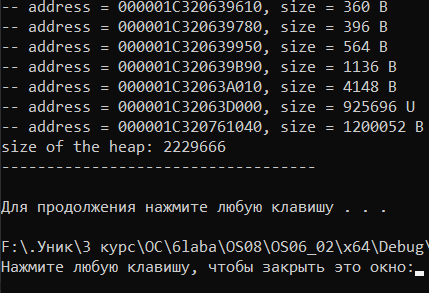




**Задание 04.Windows**

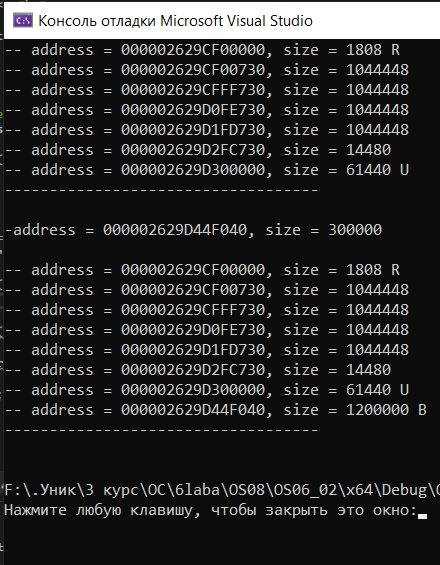
1. Разработайте консольное приложение **OS06\_04,** которое включает функцию **sh**, принимающую 1 параметр: дескриптор (HANDLE) heap.
2. Функция **sh** выводит на консоль, общий размер heap, размеры распределенной и нераспределенных областей памяти heap.
3. Приложение **OS06\_04** размещает в стандартной heap процесса int-массив размерности 300000.
4. Выведите с помощью функции **sh** информацию до размещения массива и после.
5. Объясните результат.





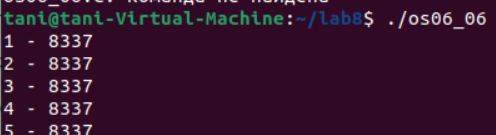
**Задание 05.Windows**

1. Разработайте консольное приложение **OS06\_05** аналогичное приложению **OS06\_05,** но использующее пользовательскую heap, которая имеет первоначальный размер 4MB.
2. Объясните результат.



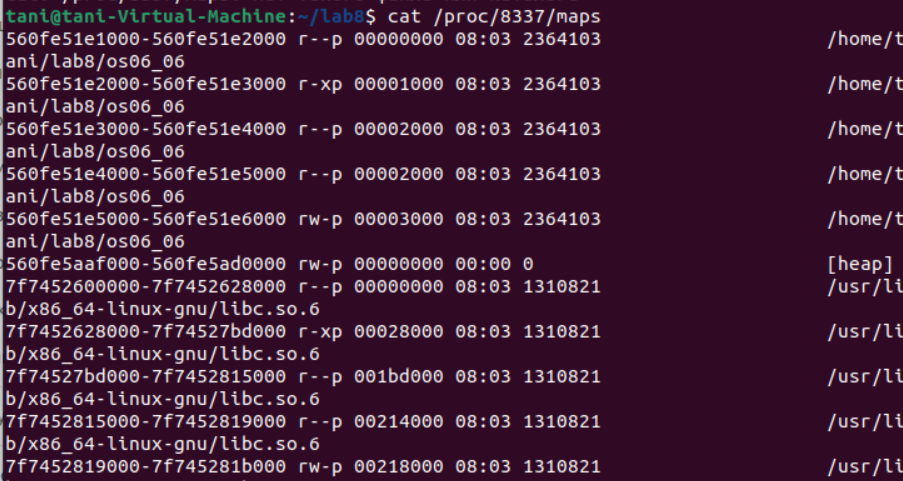
**Задание 06.Linux**

1. Разработайте консольное приложение **OS06\_06**, выполняющее длинный цикл.
2. Продемонстрируйте с помощью файловой системы **/proc** структуру адресного пространства.
3. Продемонстрируйте с помощью **pmap**  структуру адресного пространства.
4. Определите с помощью утилиты **objdump** адрес загрузки main-модуля, секций с кодом, данными, неинициализированными глобальными переменными.



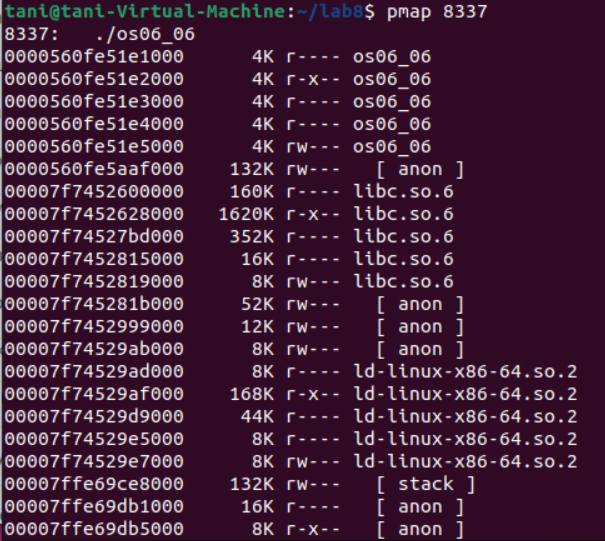
cat /proc/8337/maps

Показывает отображение виртуальной памяти процесса, включая диапазоны адресов, права доступа, смещения файлов (если применимо) и другую информацию.



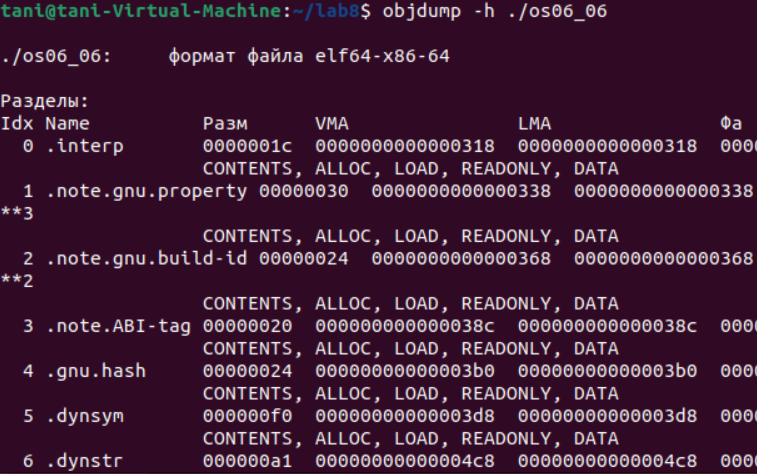
Pmap

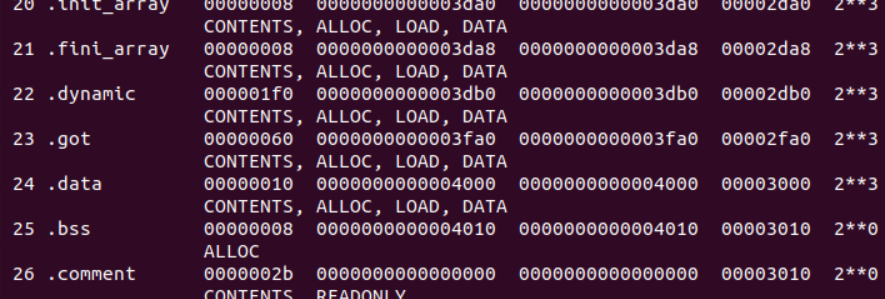
Показывает распределение использования памяти внутри процесса, включая адреса, разрешенные права доступа и размеры секций памяти.



objdump -h ./os06\_06

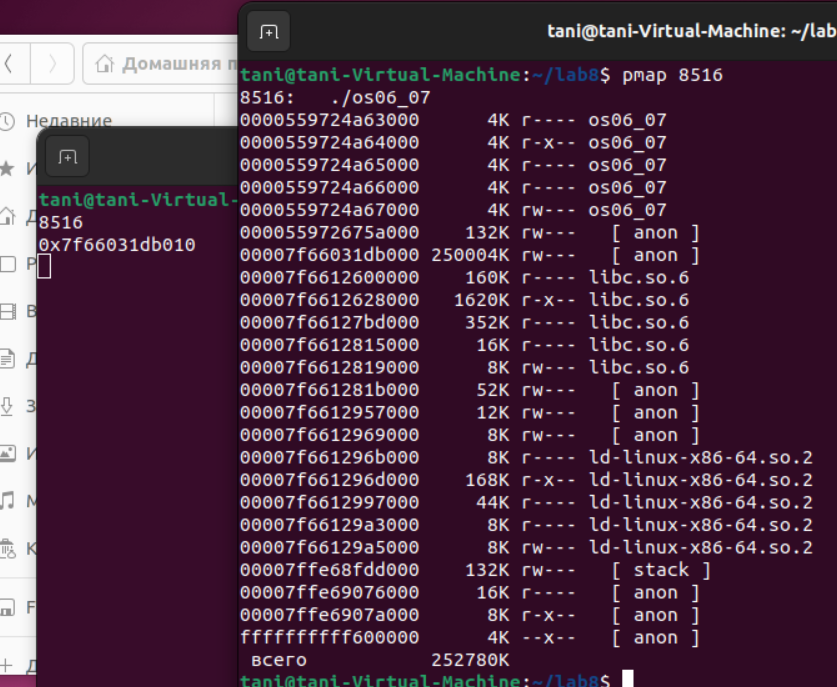
Показывает различную информацию о формате и структуре исполняемого файла, такую как архитектура, флаги, размер секций и другие атрибуты.





**Задание 07.Linux**

1. Разработайте консольное приложение **OS06\_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.
4. После инициализации приложение должно приостановить свое выполнение на длительный интервал времени.
5. С помощью утилиты **pmap** определите область памяти в которой выделена память.



**Задание 08** Ответьте на следующие вопросы

1. Поясните понятие «виртуальная память».
2. Поясните понятие «свопинг».
3. Поясните понятие «страничная память».
4. Поясните понятие MMU.
5. Поясните понятие TLB.
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.
11. Windows: поясните назначение сервиса SysMain.
12. Windows: поясните назначение файла hiberfil.sys.
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?

ghp\_UiP1dFUGPRJE81DceswyhGAoat0OD618kUI2 – github token to download a folder from gt