Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Операционные системы

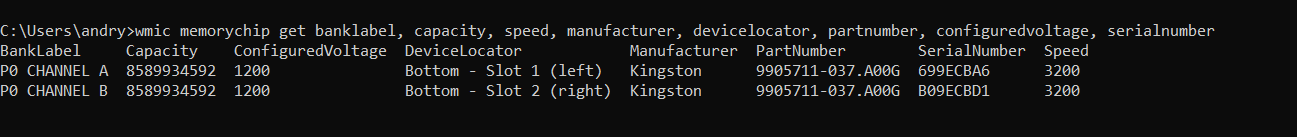
Студент: Коренчук А.В.

ФИТ 3 курс 2 группа

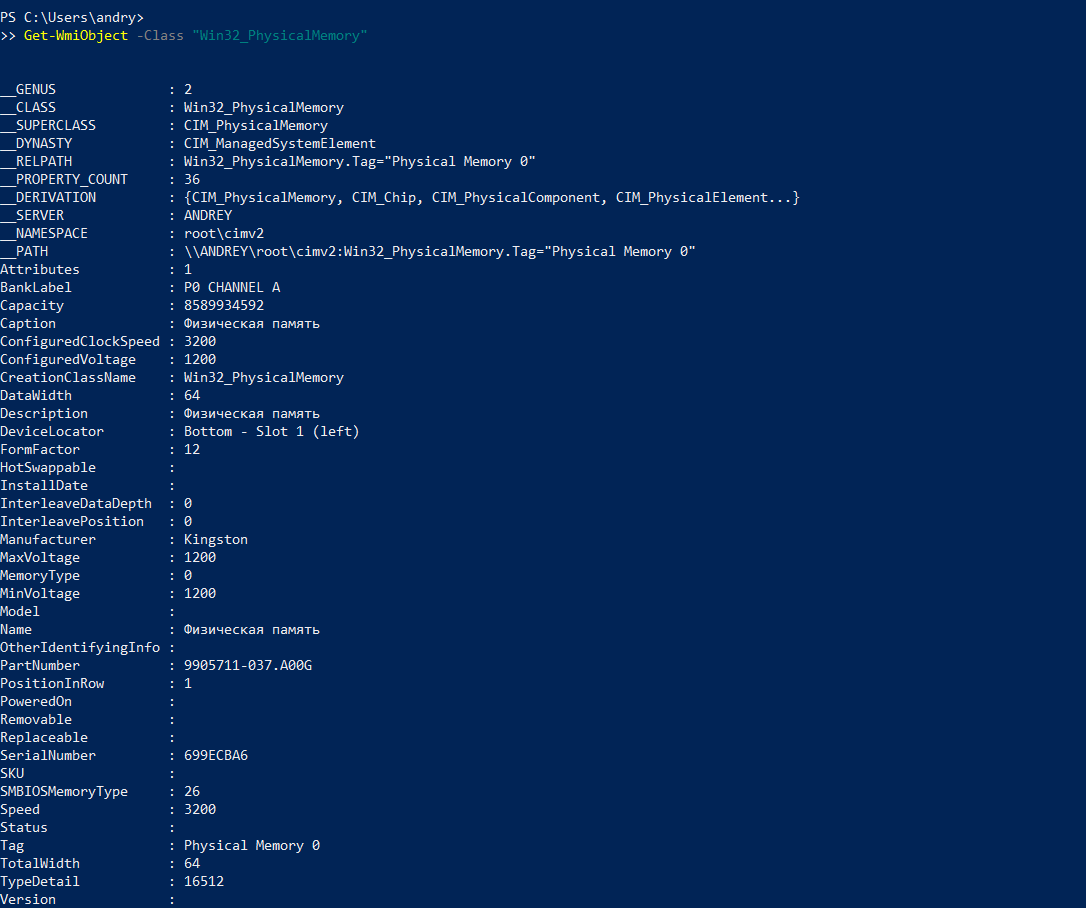
Преподаватель: Савельева М.Г.

**Задание 01. Windows**

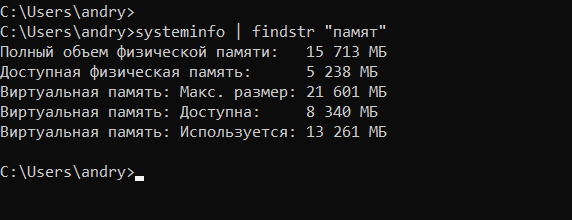
1. Получите с помощью утилиты **wmic** информации об физической оперативной памяти компьютера, поясните эту информацию.



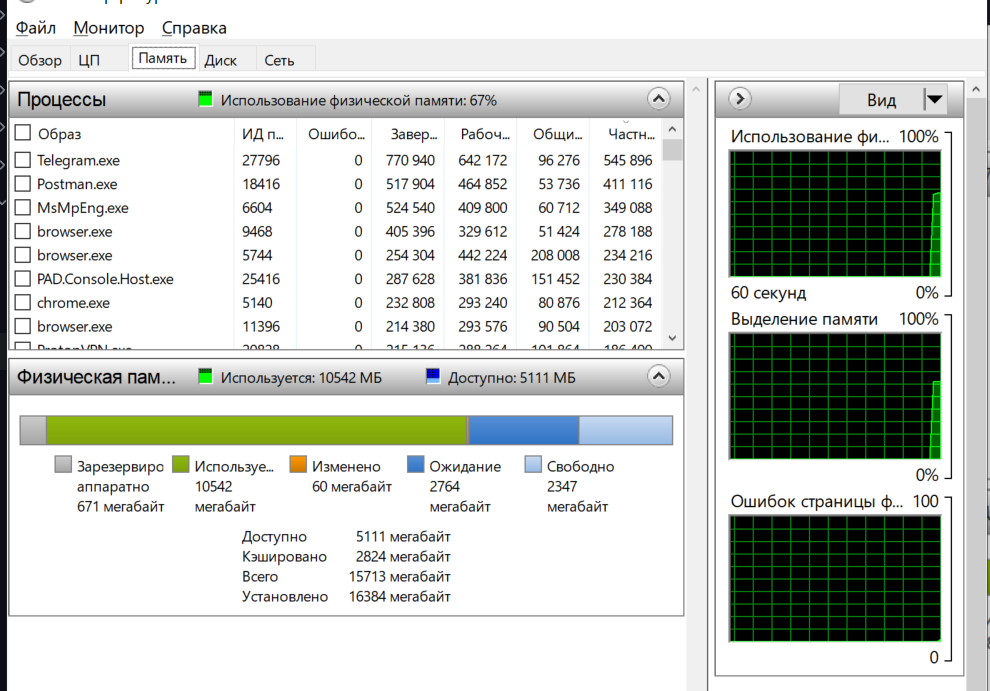
1. Получите с помощью утилиты **powershell** информации об физической оперативной памяти компьютера, поясните эту информацию.



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

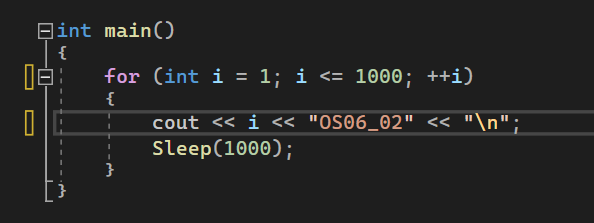


1. Получите с помощью утилиты **performance monitor** информации об оперативной памяти компьютера, поясните эту информацию.

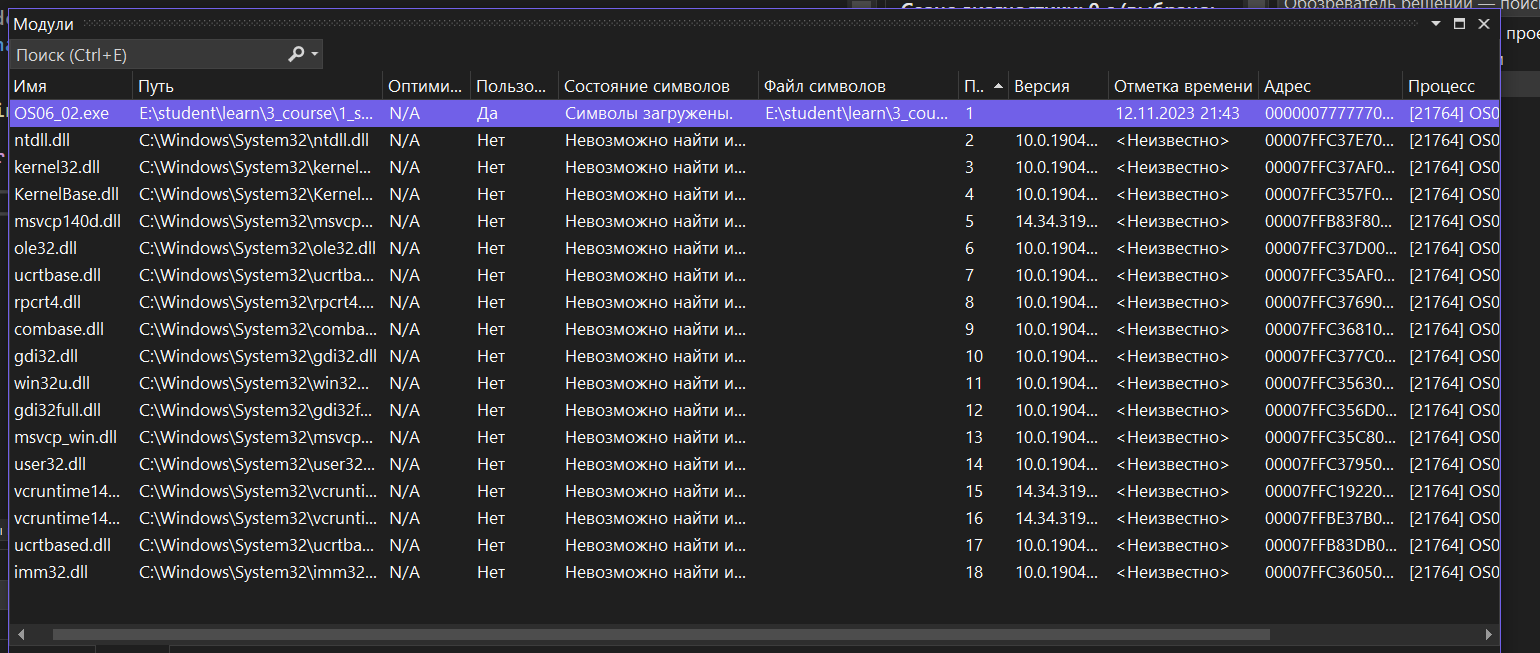


**Задание 02.Windows**

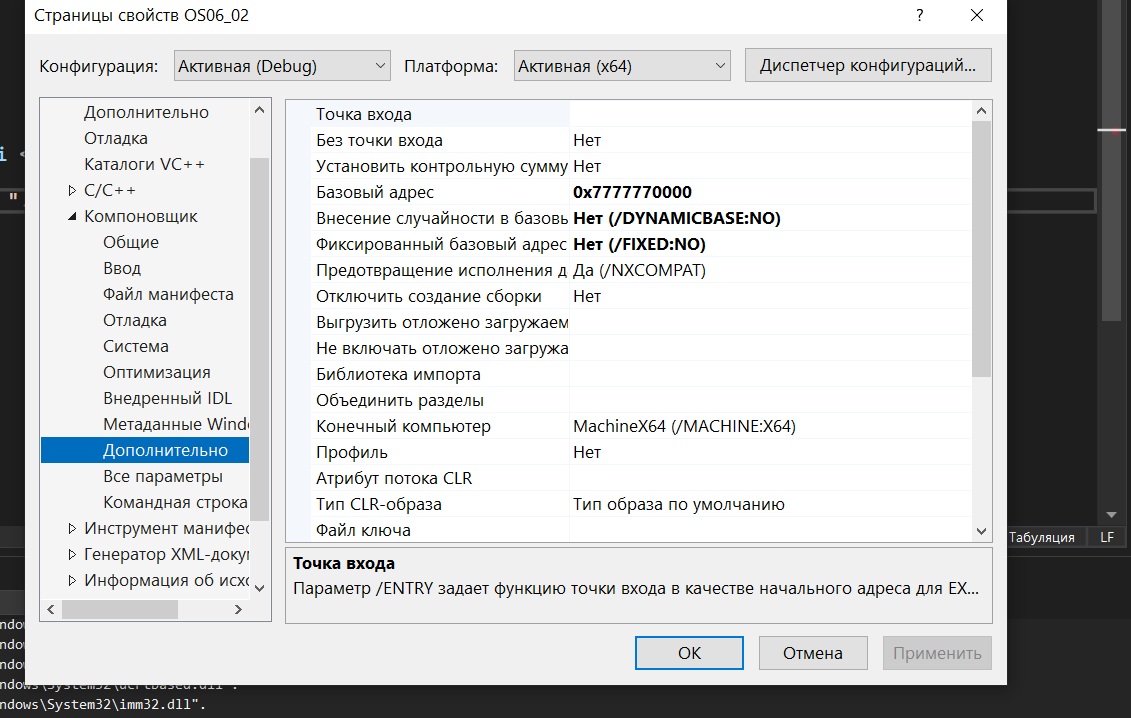
1. Разработайте консольное приложение **OS06\_02**, выполняющее длинный цикл.



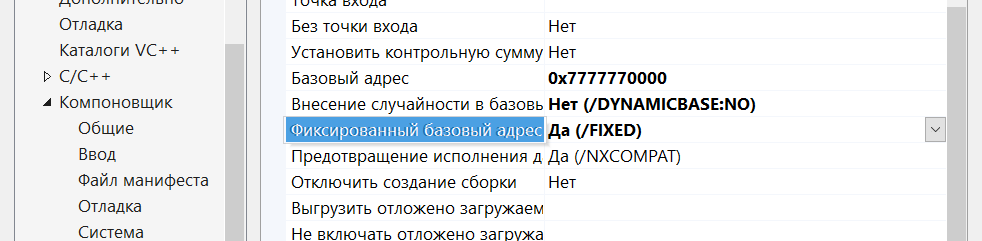
1. Продемонстрируйте с помощью отладчика адреса расположения модулей приложения **OS06\_02**.

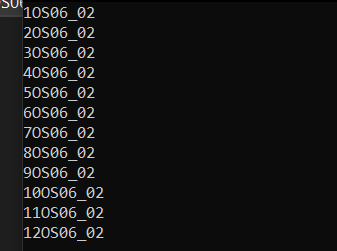


1. Установите для приложения **OS06\_02** стандартный адрес загрузки в память.



1. Продемонстрируйте с помощью отладчика стандартный адрес расположения модулей приложения **OS06\_02**



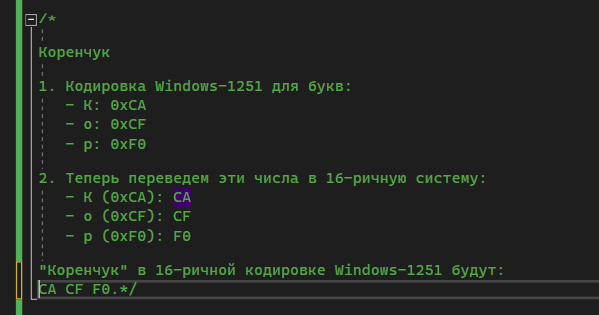


**Задание 03.Windows**

1. Разработайте консольное приложение **OS06\_03**, выполняющее получение 256 страниц оперативной памяти.



1. Разместите в этой памяти массив типа **int,** полностью занимающее все 256 страниц.
2. Заполните этот массив нарастающей последовательностью чисел с шагом 1.
3. Запишите 3 первых буквы своей фамилии в 16-ричными числами в кодировке Windows-1251.



1. 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.

*Пример: Иванов*

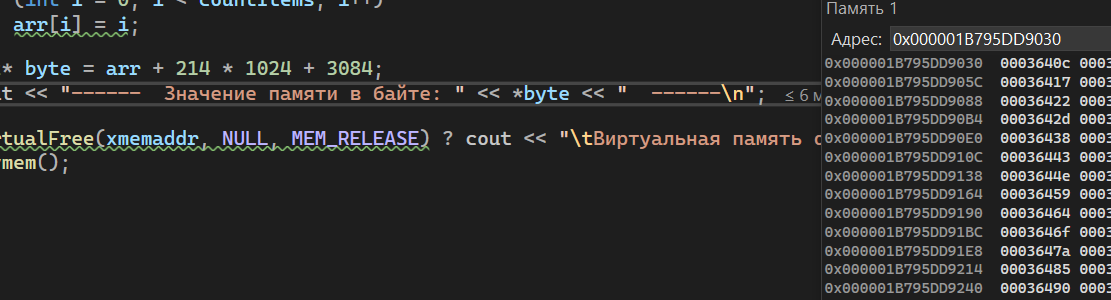
*И = C8*

*в = E2*

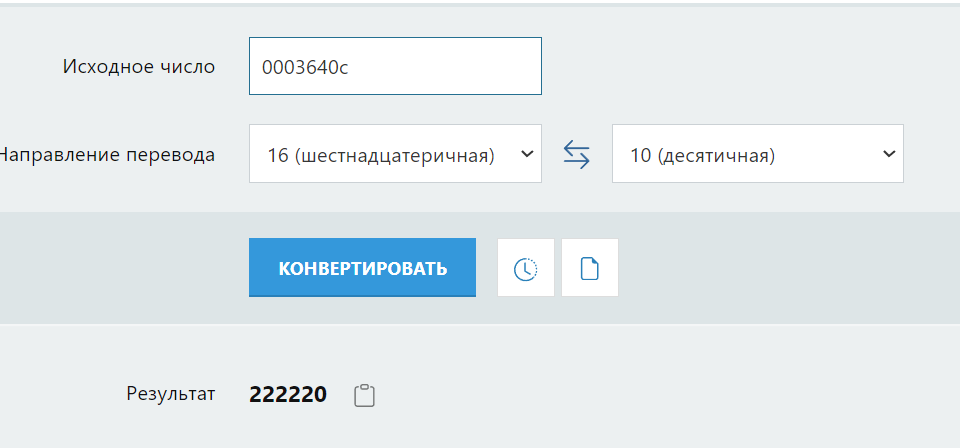
*а = E0*

*Страница C8 = 200, смещение E2E = 3630*

**

**

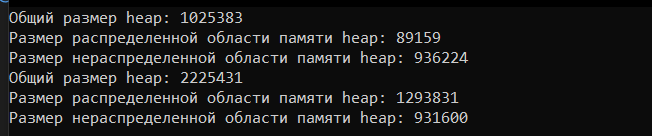




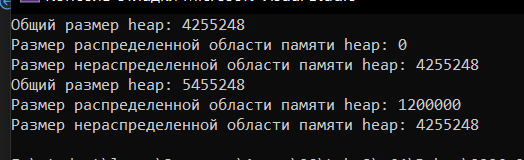


**Задание 04.Windows**

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.

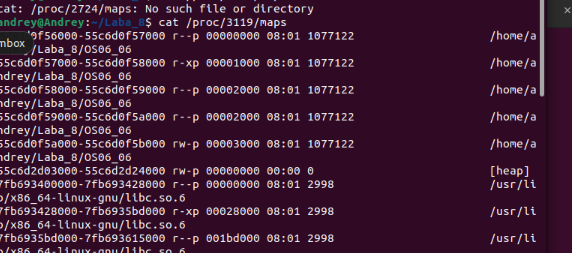


**Задание 05.Windows**

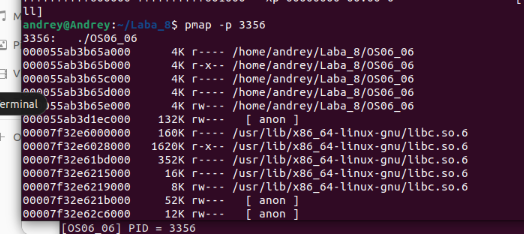
1. Разработайте консольное приложение **OS06\_05** аналогичное приложению **OS06\_05,** но использующее пользовательскую heap, которая имеет первоначальный размер 4MB.
2. Объясните результат.
3. 

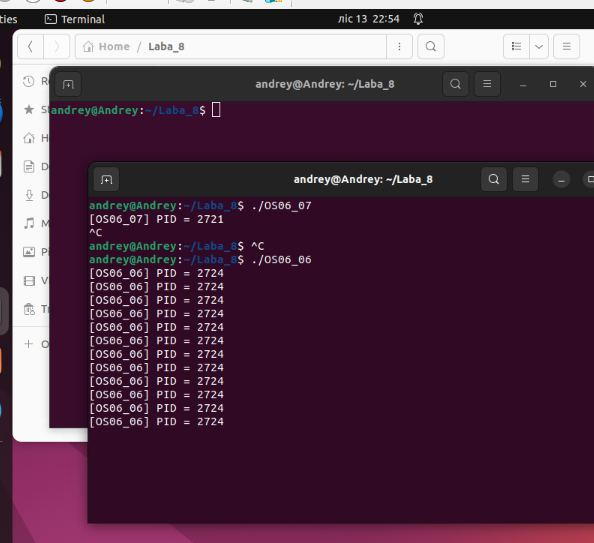
**Задание 06.Linux**

1. Разработайте консольное приложение **OS06\_06**, выполняющее длинный цикл.
2. Продемонстрируйте с помощью файловой системы **/proc** структуру адресного пространства.

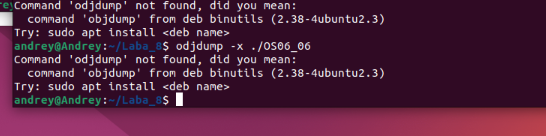


1. Продемонстрируйте с помощью **pmap**  структуру адресного пространства.



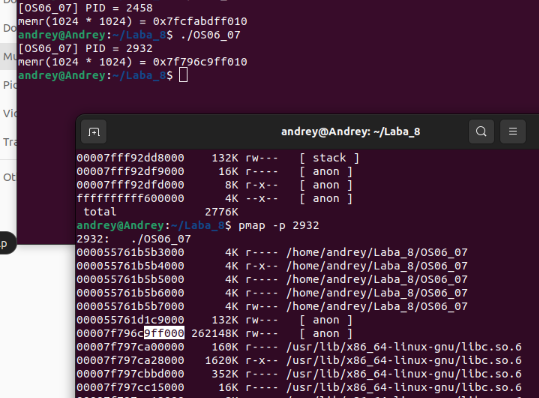


1. Определите с помощью утилиты objdump адрес загрузки main-модуля, секций с кодом, данными, неинициализированными глобальными переменными.



**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?

Виртуальная память - механизм ОС, предоставляющий программам больше памяти, чем физически доступно, через создание виртуальных адресов.

Свопинг - процесс перемещения данных из ОЗУ на диск, когда ОЗУ исчерпана, для освобождения места.

Страничная память - механизм, разделяющий физическую и виртуальную память на небольшие блоки (страницы).

MMU (Memory Management Unit) - устройство, преобразующее виртуальные адреса в физические и управляющее памятью.

TLB (Translation Lookaside Buffer) - буфер, хранящий часто используемые преобразования виртуальных адресов в физические.

Строка таблицы страниц содержит информацию о соответствии виртуальных и физических адресов и флагах, таких как доступ.

Принцип хэш-таблицы - эффективный поиск данных с использованием хэш-функции.

Инвертированная таблица физических страниц - структура данных, представляющая соответствие физических и виртуальных адресов.

Рабочий набор страниц - активные страницы в оперативной памяти, используемые программой.

Принцип LRU - алгоритм замещения страниц, выбирающий наиболее давно неиспользуемую страницу.

SysMain в Windows - служба управления памятью для улучшения производительности.

hiberfil.sys в Windows - файл гибернации, хранящий данные для режима гибернации.

pagefile.sys в Windows - файл подкачки, временно хранящий данные при нехватке ОЗУ.

swapfile.sys в Windows (начиная с Windows 10) - файл подкачки, заменяющий pagefile.sys.

Области адресного пространства в Windows: User Space, Kernel Space, Page Table Area, Paged Pool, Non-paged Pool.

Стандартный начальный размер heap в Windows зависит от версии и архитектуры.

Изменить начальный размер heap приложения в Windows можно с помощью функции HeapCreate и параметра dwInitialSize.

Стандартный размер stack в Windows зависит от версии ОС и архитектуры.

Изменить размер stack приложения в Windows можно с помощью функции SetThreadStackGuarantee.

GlobalMemoryStatus в Windows API предоставляет информацию о памяти.

VirtualQuery в Windows API используется для получения информации о виртуальной памяти.

Рабочее множество в Windows - набор активных страниц в памяти; управление с помощью OS API.

Страница заблокирована в Windows - фиксация страницы в физической памяти; управление с помощью VirtualLock и VirtualUnlock; максимум - зависит от системы.

Heap в Windows - область памяти для динамического выделения; heap процесса и пользовательская heap предоставляют пространство для malloc и calloc.

Области адресного пространства в Linux: Text, Data, Heap, Stack.

Функции malloc и calloc выделяют память в Heap в Linux.