1. **Поясните понятие «виртуальная память».**

**Виртуальная память** – метод управления памятью процессора, предназначенный для выполнения программ, которым выделяется адресное пространство превышающее доступный физический объем памяти компьютера.

1. **Поясните понятие «свопинг».**

**Свопинг** – механизм OS обмена (вытеснения и загрузки) содержимым блоков оперативной физической памяти компьютера с устройством хранения данных с целью расширения адресуемого объема оперативной памяти компьютера. Механизм является аппаратно-программным.

1. **Поясните понятие «страничная память».**

**Страничная память** – реализации виртуальной памяти, при которой физическая память и адресное пространство разбивается на блоки (страницы), а также осуществляется страничный свопинг. Размеры страниц для X86-64: 4K, 2MB, 1GB.

В основе виртуальной памяти лежит идея, что у каждой программы имеется собственное адресное пространство, которое разбивается на участки, называемые страницами. Каждая страница представляет собой непрерывный диапазон адресов. Эти страницы отображаются на физическую память, но для запуска программы одновременное присутствие в памяти всех страниц необязательно.

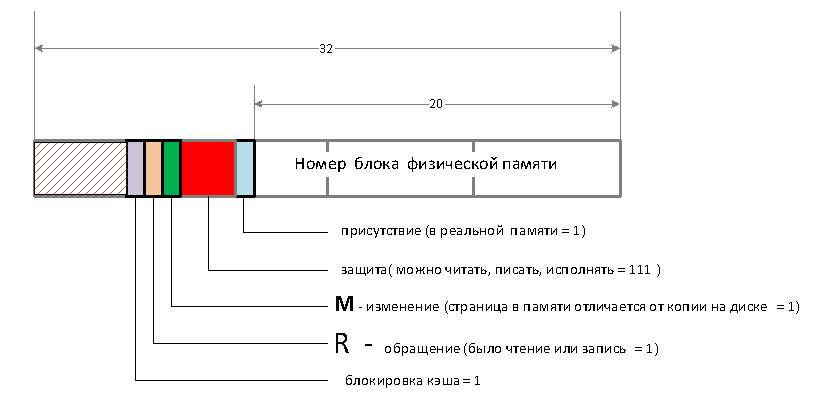
1. **Поясните понятие MMU.**

**MMU** – Memory Management Unit – диспетчер памяти – аппаратное (программируемое) устройство, входящее в состав процессора и предназначенное для трансляции виртуальных адресов оперативной памяти в реальные.

1. **Поясните понятие TLB.**

**TLB** (Translation Lookaside Buffer) – буфер быстрого преобразования адреса; компонент MMU, предназначенный для вычисления реальных адресов, хранит 64 строки таблицы страниц, полностью таблица хранится во вторичной (диск) памяти без свопинга.

1. **Какая информация содержится в строке таблицы страниц?**



Блокировка кэша – блокируется кэширование реальных страниц, которые содержат регистры портов ввода/вывода.

Биты защиты,обращения и изменения используются в алгоритмах вытеснения.

1. **Поясните принцип применения хэш-таблиц.**

Инвертированные таблицы - таблица для физических страниц. Инвертированные таблицы, для ускорения поиска номера реальной таблицы, обычно реализованы в виде хэш-таблицы, созданной на основе виртуальных адресов. Все находящиеся на данный момент в памяти виртуальные страницы, имеющие одинаковые хэш-значения, связываются в одну цепочку. Как только будет найден номер страничного блока, в TLB будет введена новая пара значений (виртуального, физического).

1. **Поясните применение «инвертированной таблицы физических» страничной памяти.**

**Инвертированная таблица страниц** – таблица для физических страниц. В данной конструкции имеется одна запись для каждого страничного блока в реальной памяти, а не одна запись на каждую страницу в виртуальном адресном пространстве. В каждой записи отслеживается, что именно находится в страничном блоке (процесс, виртуальная страница).

1. **Поясните понятие «рабочий набор страниц».**

Набор страниц, который процесс использует в данный момент, известен как **рабочий набор**.

1. **Поясните принцип работы алгоритма LRU.**

**LRU (least recently used)** — это алгоритм, при котором вытесняются значения, которые дольше всего не запрашивались. Соответственно, необходимо хранить время последнего запроса к значению. И как только число закэшированных значений превосходит N необходимо вытеснить из кеша значение, которое дольше всего не запрашивалось.

1. **Windows: поясните назначение сервиса SysMain.**

**SysMain** – утилита, занимающаяся предварительной подкачкой наиболее часто использующихся страниц. При необходимости замены страницы, высока вероятность, что страница в сжатом виде уже будет в памяти.

1. **Windows: поясните назначение файла hiberfil.sys.**

**Файл hiberfil.sys** — это файл гибернации, используемый в Windows для хранения данных и их последующей быстрой загрузки в оперативную память при включении компьютера или ноутбука.

1. **Windows: поясните назначение файла pagefile.sys.**

**Файл pagefile.sys** — это файл подкачки операционной системы Windows. При нехватке оперативной памяти Windows резервирует определенное место на жестком диске и использует его для увеличения своих возможностей. Иными словами, выгружает часть данных из оперативной памяти в файл pagefile.sys.

1. **Windows: поясните назначение файла swapfile.sys.**

**Файл swapfile.sys -** является аналогом файла подкачки **pagefile.sys,** но используется в качестве виртуальной памяти для временного хранения данных современных приложений (UWP).

1. **Windows: перечислите области адресного пространства (от младших к старшим адресам) и поясните их назначения.**

Code->Static->Heap->Data->Stack.

**Сегмент heap** – структура данных, с помощью которой реализована динамически распределяемая память приложения.

**Сегмент кода** – содержит машинные команды.

**Сегмент данных** – содержит данные, то есть константы и рабочие области, необходимые программе.

**Сегмент стека** – содержит адреса возврата в точку вызова подпрограмм.

1. **Windows: какой стандартный начальный размер области heap?**

**Heap** – фрагмент памяти адресного пространства (по умолчанию 1MB), предназначенный для динамического использования

1. **Windows: каким образом можно изменить начальный размер области памяти heap приложения?**

Можно установить стартовое значение величины HEAP в параметрах Linker.

В Visual Studio : Проект -> Свойства -> Свойства конфигурации -> Компоновщик -> Все параметры -> Резервный размер кучи.

Если компилировать через консоль: /HEAP:”значение”

1. **Windows: какой стандартный размер области памяти stack?**

По умолчанию размер stack равен 1MB.

1. **Windows: каким образом можно изменить размер области памяти stack приложения?**

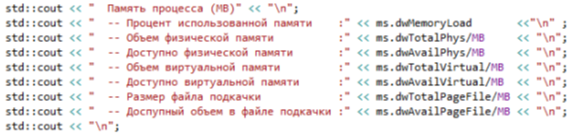
В Visual Studio : Проект -> Свойства -> Свойства конфигурации -> Компоновщик -> Все параметры -> Резервный размер стека.

Если компилировать через консоль: /STACK:”значение”.

1. **Windows: поясните назначение функции Windows API: GlobalMemoryStatus.**

**GlobalMemoryStatus** получает информацию о текущем использовании системой как физической, так и виртуальной памяти.

MEMORYSTATUS – структура, в которую записывается информация, получаемая при помощи данной функции.



1. **Windows: поясните назначение функции Windows API: VirtualQuery; перечислите значения атрибутов Protect, State и Type.**

**VirtualQuery** извлекает информацию о диапазоне страниц в виртуальном адресном пространстве вызывающего процесса.

**Protect**:

1. PAGE\_READONLY – только для чтения;
2. PAGE\_READWRITE – чтение/запись;
3. PAGE\_EXECUTE – исполнять;
4. PAGE\_EXECUTE\_READWRITE – читать/писать/исполнять;
5. PAGE\_NOACCESS – нет доступа;
6. PAGE\_NOCACHE – не кэшировать.

**State**:

1. MEM\_COMMIT – распределена по программе;
2. MEM\_RESERVE – зарезервировано;
3. MEM\_RESET – временно не используется.

**Type**:

1. MEM\_PRIVATE – принадлежит процессу;
2. MEM\_MAPPERD – файл, проецируемый в память;
3. MEM\_IMAGE – исполняемый код.
4. **Windows: что такое «рабочее множество»? Поясните принцип управления рабочим множеством с помощью OS API.**

**Рабочее множество** - количество памяти, требующееся процессу в заданный интервал времени.

Процесс уменьшает или опустошает рабочий набор, вызывая функцию **SetProcessWorkingSetSize** (устанавливает минимальный и максимальный размеры рабочего набора для указанного процесса.)

BOOL ***SetProcessWorkingSetSize***(

HANDLE hProcess, //дескриптор процесса

SIZE\_T dwMinimumWorkingSetSize, //минимальный размер рабочего множества

SIZE\_T dwMaximumWorkingSetSize //максимальный размер рабочего множества

);

BOOL **EmptyWorkingSet(**HANDLE hProcess) **-** Удаляет как можно больше страниц из рабочего набора указанного процесса.

Согласно модели рабочего множества, процесс может находиться в ОЗУ тогда и только тогда, когда множество всех его страниц, используемых в настоящее время (или множество последних по времени использования страниц, которое часто используется как его приближение) могут находиться в ОЗУ.

1. **Windows: что означает «страница заблокирована»? с помощью каких функций OS API можно установить блокировку страниц и снять блокировку? Какое максимальное количество страниц можно заблокировать?**

Страницы, заблокированные процессом, остаются в физической памяти до тех пор, пока процесс не разблокирует их или не завершит работу. Эти страницы гарантированно не будут записаны в файл подкачки, пока они заблокированы.

BOOL VirtualLock(LPVOID lpAddress ,SIZE\_T dwSize) / BOOL VirtualUnlock(LPVOID lpAddress ,SIZE\_T dwSize) – блокирует/(снимает блокировку) указанную область виртуального адресного пространства процесса в физической памяти, гарантируя, что последующий доступ к области не приведет к ошибке страницы.

Максимальное количество страниц, которые может заблокировать процесс, равно количеству страниц в его минимальном рабочем наборе за вычетом небольших накладных расходов.

1. **Windows: что такое «heap»? Что такое «heap процесса»? Что такое «пользовательская heap»? Поясните принцип устройства heap.**

**Heap** – область памяти адресного пространства, предназначенного для использования программной фрагментов динамически выделяемой памяти.

При инициализации процесса система создает в его адресном пространстве кучу. (Ее размер по умолчанию — 1 Мб). Пользовательская heap - куча создаваемая пользователем.

При запуске процесса ОС выделяет память для размещения кучи. В дальнейшем память для кучи (под кучу) может выделяться динамически.

Программа пользователя, используя [функции](https://ru.wikipedia.org/wiki/%D0%A4%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D1%8F_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)), подобные malloc(), может получать [указатели](https://ru.wikipedia.org/wiki/%D0%A3%D0%BA%D0%B0%D0%B7%D0%B0%D1%82%D0%B5%D0%BB%D1%8C_(%D1%82%D0%B8%D0%BF_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85)) на области [памяти](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D0%B8%D0%B2%D0%BD%D0%B0%D1%8F_%D0%BF%D0%B0%D0%BC%D1%8F%D1%82%D1%8C), принадлежащие куче. Программы используют кучу для размещения динамически создаваемых [структур данных](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85). Программа может освободить память с помощью [функций](https://ru.wikipedia.org/wiki/%D0%A4%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D1%8F_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)), подобных free().

1. **Linux: перечислите области адресного пространства (от младших к старшим адресам) и поясните их назначения.**

Code->Data->Heap->Stack.

**Сегмент heap** – структура данных, с помощью которой реализована динамически распределяемая память приложения.

**Сегмент кода** – содержит машинные команды.

**Сегмент данных** – содержит данные, то есть константы и рабочие области, необходимые программе.

**Сегмент стека** – содержит адреса возврата в точку вызова подпрограмм.

1. **Linux: в какой части адресного пространства выделяется памяти с помощью функций malloc, calloc?**

Heap segment.