**Вопросы к лабораторной работе №8**

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.

2. 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.

3. Поясните понятие «страничная память».

Страничная память – реализация 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.

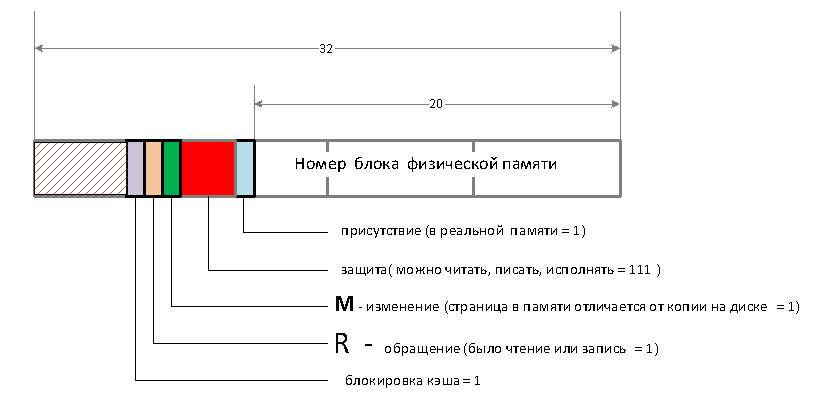
4. Поясните понятие MMU.

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

5. 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.

6. Какая информация содержится в строке таблицы страниц



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

2. Защита, обращение и изменение - используется в алгоритмах вытеснения.

7. Поясните принцип применения хэш-таблиц.

Решает проблему инвертированной таблицы, которая усложняла преобразование виртуальных адресов в физические. Одним из приемлемых способов осуществления этого поиска является ведение 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).

8. Поясните применение «инвертированной таблицы физических» страничной памяти.

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).

9. Поясните понятие «рабочий набор страниц».

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

Проще говоря, рабочий набор – это набор страниц памяти, которые в настоящее время принадлежат вашему процессу и не выгружены (то есть в ОЗУ).

10. 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.

11. Windows: поясните назначение сервиса SysMain.

SysMain – это сервис, который работает в фоновом режиме, постоянно анализируя модели использования ОЗУ и выясняя, какие приложения вы запускаете чаще всего. Со временем SysMain отмечает эти приложения как «часто используемые» и предварительно загружает их в оперативную память. Идея в том, что, когда вы хотите запустить приложение, оно запустится намного быстрее, потому что оно уже загружено в память.

12. Windows: поясните назначение файла hiberfil.sys.

hiberfile.sys – файл для сохранения памяти в режиме «сон» (гибернация).

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.

13. Windows: поясните назначение файла pagefile.sys.

pagefile.sys - файл подкачки.

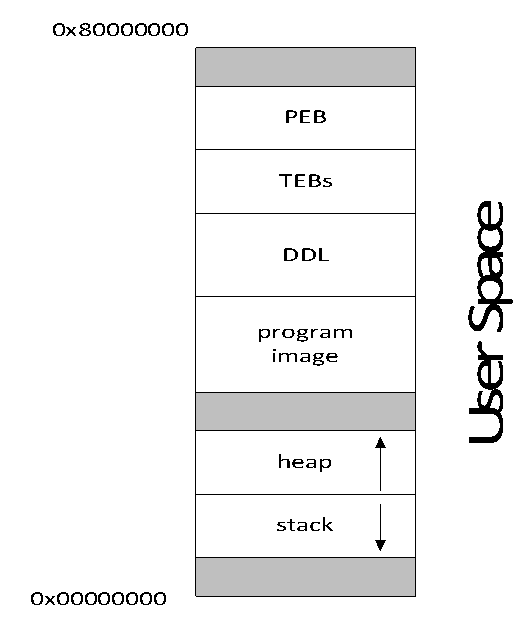
Зачем он нужен? Дело в том, что какое бы количество оперативной памяти не было установлено на вашем компьютере, не всем программам для работы будет ее достаточно. Современные игры, видео и графические редакторы и многое другое программное обеспечение с легкостью заполнит Ваши 8 Гб RAM и попросит еще. В этом случае и используется файл подкачки.

14. Windows: поясните назначение файла swapfile.sys.

swapfile.sys – файл подкачки отдельных (предварительно скаченных из магазина приложений UWP) для быстрого применения (в случае надобности).

Файл swapfile.sys ещё один файл подкачки (в дополнение к pagefile.sys), но служащий исключительно для приложений из магазина приложений (UWP).

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



PEB- структура процесса

TEB -потоки

16. Windows: какой стандартный начальный размер области heap?

По умолчанию – 1MB, из них 4K сразу забирает процесс.

17. 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:”2097152”

18. Windows: какой стандартный размер области памяти stack?

1MB

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

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

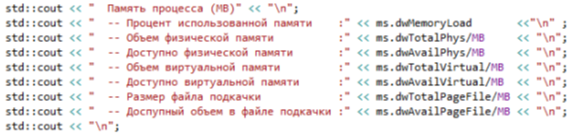
Если компилировать через консоль: /STACK:”2097152”

20. Windows: поясните назначение функции Windows API: GlobalMemoryStatus.

GlobalMemoryStatus получает информацию о текущем состоянии памяти компьютера.

MEMORYSTATUS ms; // ствуктура в которую записывается инфа

GlobalMemoryStatus(&ms);



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

Извлекает информацию о диапазоне страниц в виртуальном адресном пространстве вызывающего процесса.

Protect: Атрибуты защиты участка виртуальной памяти.



State: Состояние участка виртуальной памяти



Type: Тип участка виртуальной памяти



22. Windows: что такое «рабочее множество»? поясните принцип управления рабочим множеством с помощью OS API.

Рабочее множество - количество памяти, требующееся [процессу](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81_(%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)) в заданный интервал времени.

*Рабочий набор процесса – это набор страниц в виртуальном адресном пространстве процесса, которые в настоящее время находятся в физической памяти.*

Операционные системы, включая Windows, предоставляют API для управления рабочим множеством процесса. Два основных API, связанных с управлением рабочим множеством в Windows, - это GetProcessWorkingSetSize и SetProcessWorkingSetSize.

GetProcessWorkingSetSize:

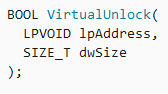
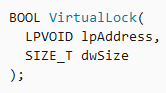
Эта функция используется для получения информации о текущем размере и максимальном размере рабочего множества процесса.

SetProcessWorkingSetSize:

Эта функция используется для установки минимального и максимального размеров рабочего множества процесса.

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?

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.



Блокирует 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.

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

Heap – область памяти адресного пространства, предназначенного для использования программой фрагментов динамически выделяемой памяти (malloc, new).

При инициализации процесса система создает в его адресном пространстве кучу. (Ее размер по умолчанию — 1 Мб). Но система позволяет увеличивать этот размер, для чего надо указать компоновщику при сборке программы ключ /HEAP.

Пользовательская heap – куча, создаваемая пользователем.

При запуске [процесса](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81_(%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)) [ОС](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) выделяет [память](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) для размещения кучи. В дальнейшем [память](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) для кучи (под кучу) может выделяться динамически.

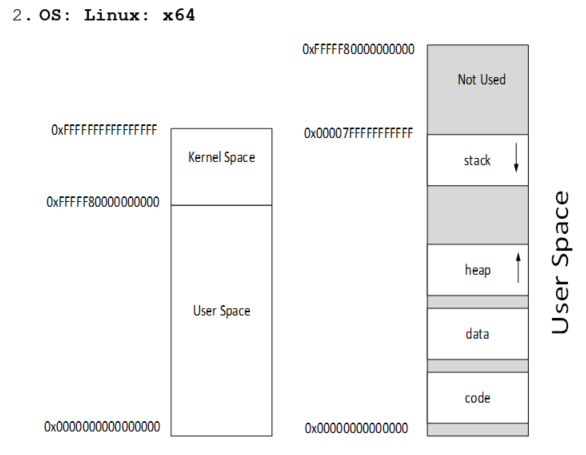
Программа пользователя, используя [функции](https://ru.wikipedia.org/wiki/%D0%A4%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D1%8F_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)), подобные malloc(), может получать [указатели](https://ru.wikipedia.org/wiki/%D0%A3%D0%BA%D0%B0%D0%B7%D0%B0%D1%82%D0%B5%D0%BB%D1%8C_(%D1%82%D0%B8%D0%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().

[Память](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) кучи можно разделить на занятую (выделенную программе 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 malloc()) и свободную (ещё не занятую или уже освобождённую 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()).

Для хранения данных о том, какая область кучи является занятой, а какая — свободной, обычно используется дополнительная область памяти.

Перед началом работы программы выполняется инициализация кучи, в ходе которой [память](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), выделенная под кучу, отмечается как свободная.

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?

HEAP