



Школа _____ Класс _____

IT направление

10 – 11 классы

Прочтите текст и выполните письменно задания после текста

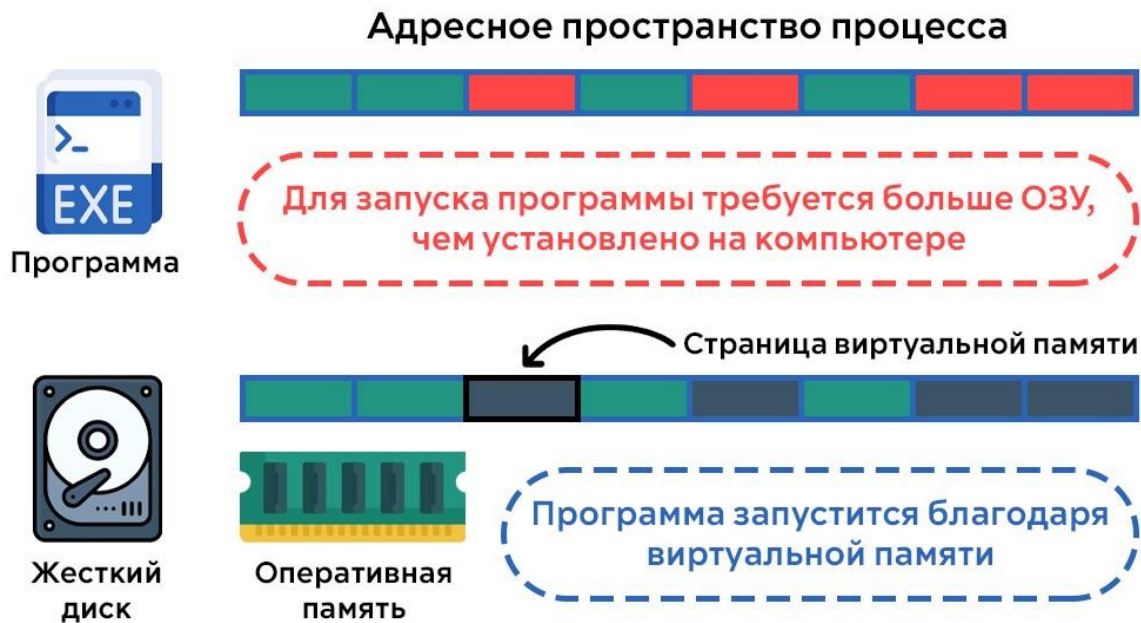
Как работает виртуальная память

Виртуальная память играет ключевую роль в распределении ограниченных ресурсов оперативной памяти между процессами. Она не только обеспечивает бесперебойную работу компьютера, но и создаёт для каждой программы иллюзию неограниченного адресного пространства. Это позволяет процессам работать независимо друг от друга, защищая их от несанкционированного доступа и предотвращая сбои системы. Благодаря механизму виртуальной памяти операционная система эффективно управляет физическими ресурсами, оптимизирует использование памяти и поддерживает стабильность работы даже при высокой нагрузке.

Основные концепции виртуальной памяти

Основные концепции виртуальной памяти заключаются в следующем. Оперативная или физическая память компьютера представляет собой реальное хранилище, в котором размещаются исполняемый код и данные активных процессов. Однако использование физической памяти имеет определённые ограничения: размер программы не должен превышать доступную физическую память, необходимо следить за тем, чтобы адресное пространство запускаемой программы не пересекалось с работающими процессами, и нужно изолировать каждый процесс от несанкционированного доступа других программ. Механизм виртуальной памяти решает эти проблемы, предоставляя каждому процессу виртуальное адресное пространство. При этом программа работает только с виртуальными адресами, будучи полностью изолированной от физического размещения данных.

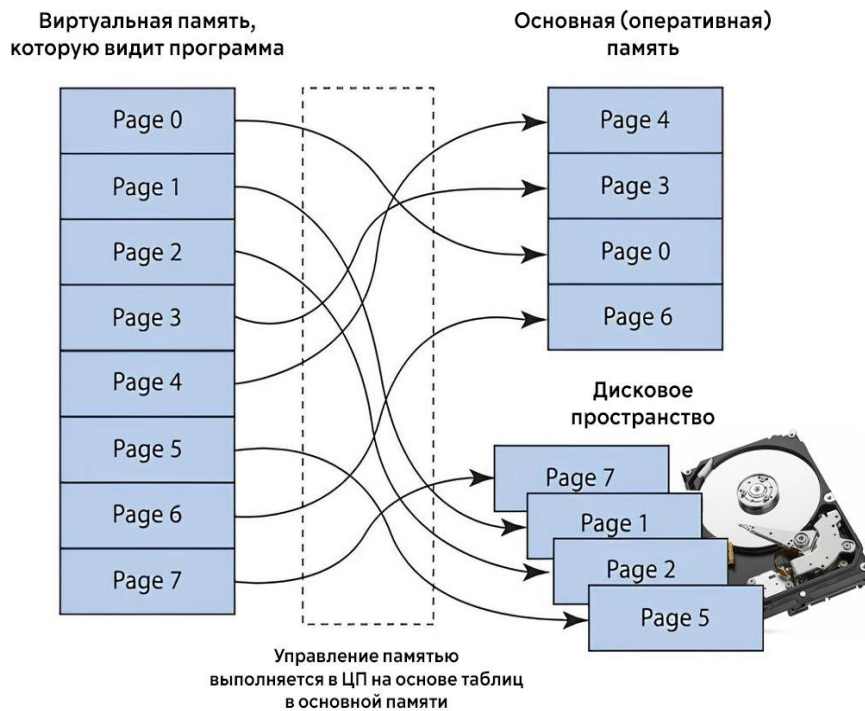
Такой подход обеспечивает три ключевых преимущества: процессы могут работать с объёмом данных, превышающим физическую память компьютера; процессы функционируют в отдельных виртуальных участках памяти, что предотвращает вмешательство одного процесса в память другого; операционная система может контролировать доступ к отдельным страницам памяти, предотвращая несанкционированный доступ. Каждый процесс обладает собственным виртуальным адресным пространством, которое разделено на фиксированные блоки данных — страницы. Размер этих страниц составляет 4 КБ или 8 КБ в зависимости от архитектуры системы. Физическая память также разделена на блоки того же размера, которые называются фреймами.



Когда процессу требуется доступ к данным, он обращается к виртуальному адресу. Операционная система преобразует этот виртуальный адрес в физический с помощью таблицы страниц. Если запрашиваемые данные отсутствуют в физической памяти, они загружаются с диска в оперативную память.

Страничный файл, который также называют файлом подкачки или swap-файлом, представляет собой область на жёстком диске, используемую для хранения страниц виртуальной памяти, когда они не помещаются в физическую память. С помощью файла подкачки операционная система освобождает физическую память, перемещая неактивные страницы на диск и загружая их обратно.

Механизм работы виртуальной памяти



Процесс преобразования виртуальных адресов в физические состоит из нескольких этапов. Сначала происходит разделение памяти: виртуальное адресное пространство разбивается на страницы, а физическая память — на фреймы. Для каждого процесса создаётся таблица страниц, которая содержит информацию о соответствии между виртуальными страницами и физическими фреймами. Когда процесс обращается к виртуальному адресу, операционная система использует таблицу страниц для определения соответствующего физического адреса. Если запрашиваемая страница отсутствует в физической памяти, возникает страничная ошибка, и ОС загружает требуемую страницу с диска в память. После загрузки страницы таблица страниц обновляется, чтобы отразить новое местоположение данных. Для оптимизации процесса преобразования адресов используется аппаратное устройство — буфер трансляции (TLB). TLB представляет собой кэш, в котором хранятся недавно использованные преобразования виртуальных адресов в физические, что значительно ускоряет доступ к памяти.

Когда физическая память заполнена и требуется загрузить новую страницу, операционная система должна решить, какую страницу выгрузить на диск. Для принятия решения используются алгоритмы замещения страниц. Алгоритм FIFO выгружает страницу, которая находится в памяти дольше всего. Этот алгоритм не учитывает частоту использования страниц. Алгоритм LRU выгружает страницу, которая не использовалась дольше всего. Он эффективнее FIFO, но требует больше ресурсов для отслеживания времени последнего использования каждой страницы. Алгоритм LFU выгружает страницу, которая использовалась реже всего, и полезен, когда данные часто используются и их желательно хранить в кэше. Алгоритм Clock похож на LRU и использует бит доступа для каждой страницы, циклически проверяя страницы, пока не найдёт страницу для выгрузки. Помимо алгоритмов замещения, для оптимизации использования виртуальной памяти применяются алгоритмы прогнозирования, которые основаны на том, что процессы, как правило, обращаются к ограниченному набору страниц в течение коротких промежутков времени. Это позволяет системе предугадывать, какие страницы могут потребоваться в ближайшем будущем, и загружать их заранее.

Когда процесс пытается получить доступ к странице, которой нет в физической памяти, возникает страничная ошибка. Операционная система должна обработать это исключение, загрузить нужную страницу с диска и возобновить выполнение процесса. Чтобы сократить количество

страничных ошибок, используются методы предварительной подкачки, группировки страниц и адаптивного управления рабочим набором. Предварительная подкачка позволяет ОС предугадывать, какие страницы понадобятся процессу в ближайшем будущем, и загружать их заранее. Группировка страниц предполагает загрузку нескольких соседних страниц при обращении к одной из них. Адаптивное управление рабочим набором заключается в отслеживании набора страниц, активно используемых процессом

Операционная система играет ключевую роль в управлении виртуальной памятью. Она отвечает за создание и поддержку таблиц страниц для каждого процесса, управление физической памятью и страничным файлом, обработку страничных ошибок и загрузку страниц с диска, реализацию алгоритмов замещения страниц, оптимизацию использования памяти и минимизацию фрагментации, а также изоляцию и защиту памяти между процессами. Реализация виртуальной памяти различается в разных операционных системах. Windows использует файл подкачки (pagefile.sys), размер которого может динамически изменяться, и поддерживает механизм приоритетного ввода-вывода для страниц виртуальной памяти. Linux использует выделенный раздел диска (swap-раздел) или файл подкачки (swap-файл) и предоставляет гибкие настройки для управления swap-пространством. macOS использует механизм сжатия, который сжимает неиспользуемые страницы в памяти перед их выгрузкой на диск, что уменьшает количество операций ввода-вывода. Операционная система также отвечает за очистку и освобождение хранилища, периодическое сканирование памяти для выявления и освобождения неиспользуемых страниц, дефрагментацию памяти для оптимизации физических ресурсов, управление кэшем файловой системы и балансировку между использованием

Виртуальная память имеет несколько преимуществ. Она позволяет одновременно выполнять больше программ, чем позволяет физическая память, обеспечивает эффективное использование памяти, так как только активно используемые данные хранятся в физической памяти, повышает безопасность системы благодаря изоляции процессов и контролю доступа к памяти, упрощает разработку программного обеспечения, позволяя разработчикам создавать программы без учёта физических ограничений памяти конкретного устройства, и поддерживает большие адресные пространства, что позволяет процессам работать с объёмами данных, превышающими физическую память. Однако у виртуальной памяти есть и недостатки. Чрезмерное использование страничного файла значительно снижает производительность системы из-за низкой скорости дисковых операций. Со временем может возникнуть фрагментация как физических, так и виртуальных участков памяти, что снижает эффективность использования ресурсов. Настройка виртуальной памяти является сложной задачей, особенно в системах с ограниченными ресурсами. Интенсивное использование виртуальной памяти может привести к ускоренному износу накопителя из-за большого количества операций записи.

Виртуальная память может стать узким местом в производительности системы в нескольких случаях. Если объем физической памяти недостаточен для хранения рабочих наборов активных процессов, система будет часто обращаться к страничному файлу, что замедлит работу. Приложения, работающие с большими объемами данных, могут вызывать частые ошибки страниц, если их рабочий набор не помещается в физическую память. Неправильно настроенный размер страничного файла или неэффективные алгоритмы управления памятью также снижают производительность. Сильная фрагментация памяти приводит к неэффективному использованию как физических, так и виртуальных ресурсов.

Задания

Блок задач 1

1. Используя текст, в двух-трех предложениях объясните принцип работы виртуальной памяти.
2. Приведите два аргумента, объясняющие, почему лучше использовать именно виртуальную память (ВП) и объясните, почему каждый из приведённых Вами аргументов является существенным.
3. В физической памяти Вашего устройства может поместиться всего 5 страниц. Поочередно с диска были запрошены следующие страницы: 2, 3, 2, 4, 5, 3, 6, 5, 3, 2. Какие страницы будут находиться в памяти в конце работы? (Используйте алгоритм замещения страниц по алгоритму FIFO).
4. Коротко объясните, почему компьютер может начать «тормозить», если на диске не хватает свободного места.
5. В тексте упоминается проблема фрагментации данных. Попробуйте предположить, в чём её суть. Кратко сформулируйте и запишите своё предположение.
6. Разработайте и опишите свой алгоритм замещения страниц. Поясните, почему он корректен.
7. Кроме описанного в тексте страничного метода существует еще и сегментный, когда фреймы имеют разный размер. Попробуйте сформулировать и записать проблемы, которые могут возникнуть при использовании такого метода.
8. В некоторой архитектуре размер страницы составляет 4 КБ, а общее количество страниц равно 1025. Какое минимальное количество байт потребуется для хранения таблицы адресов, содержащей информацию обо всех страницах?

Блок задач 2 (Программирование)

Задание 9.

Подсчитайте количество букв «а» в заданном с консоли слове.

Задание 10.

Посмотрев подсказку, напишите код для перемножения матриц, заданных с консоли

Подсказка для частного случая.

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{pmatrix} \cdot \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \end{pmatrix} =$$
$$= \begin{pmatrix} a_{11} \cdot b_{11} + a_{12} \cdot b_{21} & a_{11} \cdot b_{12} + a_{12} \cdot b_{22} & a_{11} \cdot b_{13} + a_{12} \cdot b_{23} \\ a_{21} \cdot b_{11} + a_{22} \cdot b_{21} & a_{21} \cdot b_{12} + a_{22} \cdot b_{22} & a_{21} \cdot b_{13} + a_{22} \cdot b_{23} \\ a_{31} \cdot b_{11} + a_{32} \cdot b_{21} & a_{31} \cdot b_{12} + a_{32} \cdot b_{22} & a_{31} \cdot b_{13} + a_{32} \cdot b_{23} \end{pmatrix}$$