

УВАЖАЕМЫЕ СТУДЕНТЫ! Изучите приведенную лекцию, законспектируйте основные понятия, дайте ответы на контрольные вопросы.

Ответы на вопросы, фотоотчет, предоставить преподавателю на e-mail: r.bigangel@gmail.com **до 29.05.2023.**

При возникновении вопросов по приведенному материалу обращаться по следующему номеру телефона: (072)111-37-59, (Viber, WhatsApp), vk.com: <https://vk.com/daykini>

ВНИМАНИЕ!!! При отправке работы, не забывайте указывать ФИО студента, наименование дисциплины, дата проведения занятия (по расписанию).

Лекция Организация виртуальной оперативной памяти.

Необходимым условием для того, чтобы программа могла выполняться, является ее нахождение в оперативной памяти. Только в этом случае процессор может извлекать команды из памяти и интерпретировать их, выполняя заданные действия. Объем оперативной памяти, который имеется в компьютере, существенно сказывается на характере протекания вычислительного процесса. Он ограничивает число одновременно выполняющихся программ и размеры их виртуальных адресных пространств. В некоторых случаях, когда все задачи мультипрограммной смеси являются вычислительными (то есть выполняют относительно мало операций ввода-вывода, разгружающих центральный процессор), для хорошей загрузки процессора может оказаться достаточным всего 3-5 задач. Однако если вычислительная система загружена выполнением интерактивных задач, то для эффективного использования процессора может потребоваться уже несколько десятков, а то и сотен задач. Эти рассуждения хорошо иллюстрирует рис.6, на котором показан график зависимости коэффициента загрузки процессора от числа одновременно выполняемых процессов и доли времени, проводимого этими процессами в состоянии ожидания ввода-вывода.

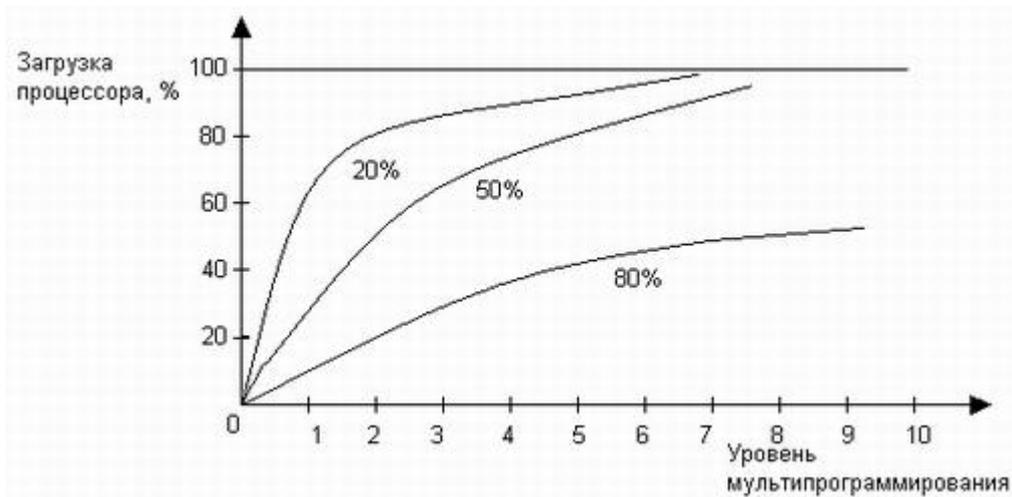


Рис 1. Зависимость загрузки процессора от числа задач и интенсивности ввода-вывода

Большое количество задач, необходимое для высокой загрузки процессора, требует большого объема оперативной памяти. В условиях, когда для обеспечения приемлемого уровня мультипрограммирования имеющейся оперативной памяти недостаточно, был предложен метод организации вычислительного процесса, при котором образы некоторых процессов целиком или частично временно выгружаются на диск.

В мультипрограммном режиме помимо активного процесса, то есть процесса, коды которого в настоящий момент интерпретируются процессором, имеются приостановленные процессы, находящиеся в ожидании завершения ввода-вывода или освобождения ресурсов, а также процессы в состоянии готовности, стоящие в очереди к процессору. Образы таких неактивных процессов могут быть временно, до следующего цикла активности, выгружены на диск. Несмотря на то, что коды и данные процесса отсутствуют в оперативной памяти, ОС «знает» о его существовании и в полной мере учитывает это при распределении процессорного времени и других системных ресурсов. К моменту, когда подходит очередь выполнения выгруженного процесса, его образ возвращается с диска в оперативную память. Если при этом обнаруживается, что свободного места в оперативной памяти не хватает, то на диск выгружается другой процесс.

Такая подмена (виртуализация) оперативной памяти дисковой памятью позволяет повысить уровень мультипрограммирования — объем оперативной памяти компьютера теперь не столь жестко ограничивает количество одновременно выполняемых процессов, поскольку суммарный объем памяти, занимаемой образами этих процессов, может существенно превосходить имеющийся объем оперативной памяти. Виртуальным называется ресурс, который пользователю или пользовательской программе представляется обладающим свойствами, которыми он в действительности не обладает. В данном случае в распоряжение прикладного программиста предоставляется виртуальная оперативная память, размер которой намного превосходит всю имеющуюся в системе реальную оперативную память. Пользователь пишет программу, а транслятор, используя виртуальные адреса, переводит ее в машинные коды так, как будто в распоряжении программы имеется однородная оперативная память большого объема. В действительности же все коды и данные, используемые программой, хранятся на дисках и только при необходимости загружаются в реальную оперативную память. Понятно, однако, что работа такой «оперативной памяти» происходит значительно медленнее.

Виртуализация оперативной памяти осуществляется совокупностью программных модулей ОС и аппаратных схем процессора и включает решение следующих задач:

- 1) размещение данных в запоминающих устройствах разного типа, например, часть кодов программы — в оперативной памяти, а часть — на диске; выбор образов процессов или их частей для перемещения из оперативной памяти на диск и обратно;
 - 2) перемещение по мере необходимости данных между памятью и диском;
- Q преобразование виртуальных адресов в физические.

Очень важно то, что все действия по организации совместного использования диска и оперативной памяти, выделение места для перемещаемых фрагментов, настройка адресов, выбор кандидатов на загрузку и выгрузку, осуществляются операционной системой и аппаратурой процессора

автоматически, без участия программиста, и никак не сказываются на логике работы приложений.

Отметим, что уже достаточно давно пользователи столкнулись с проблемой размещения в памяти программы, размер которой превышает имеющуюся в наличии свободную память. Одним из первых решений было разбиение программы на части, называемые оверлеями. Когда первый оверлей заканчивал свое выполнение, он вызывал другой оверлей. Все оверлеи хранились на диске и перемещались между памятью и диском средствами операционной системы на основании явных директив программиста, содержащихся в программе. Этот способ, несмотря на внешнее сходство, имеет принципиальное отличие от виртуальной памяти, заключающееся в том, что разбиение программы на части и планирование их загрузки в оперативную память должны были выполняться заранее программистом во время написания программы.

Виртуализация памяти может быть осуществлена на основе двух различных подходов:

- 1) свопинг (swapping) — образы процессов выгружаются на диск и возвращаются в оперативную память целиком;
- 2) виртуальная память (virtual memory) — между оперативной памятью и диском перемещаются части (сегменты, страницы и т. п.) образов процессов.

Свопинг представляет собой частный случай виртуальной памяти и, следовательно, более простой в реализации способ совместного использования оперативной памяти и диска. Однако подкачке свойственна избыточность: когда ОС решает активизировать процесс, для его выполнения, как правило, не требуется загружать в оперативную память все его сегменты полностью — достаточно загрузить небольшую часть кодового сегмента с подлежащей выполнению инструкцией и частью сегментов данных, с которыми работает эта инструкция, а также отвести место под сегмент стека.

Аналогично при освобождении памяти для загрузки нового процесса очень часто вовсе не требуется выгружать другой процесс на диск целиком, достаточно вытеснить на диск только часть его образа. Перемещение избыточной

информации замедляет работу системы, а также приводит к неэффективному использованию памяти. Кроме того, системы, поддерживающие свопинг, имеют еще один очень существенный недостаток: они не способны загрузить для выполнения процесс, виртуальное адресное пространство которого превышает имеющуюся в наличии свободную память.

В некоторых современных ОС, например, версиях UNIX, основанных на коде SVR4, механизм свопинга используется как дополнительный к виртуальной памяти, включающийся только при серьезных перегрузках системы.

Именно из-за указанных недостатков свопинг как основной механизм управления памятью почти не используется в современных ОС. На смену ему пришел более совершенный механизм виртуальной памяти, который, как уже было сказано, заключается в том, что при нехватке места в оперативной памяти на диск выгружаются только части образов процессов.

Ключевой проблемой виртуальной памяти, возникающей в результате многократного изменения местоположения в оперативной памяти образов процессов или их частей, является преобразование виртуальных адресов в физические. Решение этой проблемы, в свою очередь, зависит от того, какой способ структуризации виртуального адресного пространства принят в данной системе управления памятью. В настоящее время все множество реализаций виртуальной памяти может быть представлено тремя классами.

1. Страничная виртуальная память организует перемещение данных между памятью и диском страницами — частями виртуального адресного пространства, фиксированного и сравнительно небольшого размера.

2. Сегментная виртуальная память предусматривает перемещение данных сегментами — частями виртуального адресного пространства произвольного размера, полученными с учетом смыслового значения данных,

3. Сегментно-страничная виртуальная память использует двухуровневое деление: виртуальное адресное пространство делится на сегменты, а затем сегменты делятся на страницы. Единицей перемещения данных здесь является страница. Этот способ управления памятью объединяет в себе элементы обоих предыдущих подходов.

Для временного хранения сегментов и страниц на диске отводится либо специальная область, либо специальный файл, которые во многих ОС по традиции продолжают называть областью или файлом свопинга, хотя перемещение информации между оперативной памятью и диском осуществляется уже не в форме полного замещения одного процесса другим, а частями. Другое популярное название этой области — страничный файл (page file, или paging file).

Текущий размер страничного файла является важным параметром, оказывающим влияние на возможности операционной системы: чем больше страничный файл, тем больше приложений может одновременно выполнять ОС (при фиксированном размере оперативной памяти). Однако необходимо понимать, что увеличение числа одновременно работающих приложений за счет увеличения размера страничного файла замедляет их работу, так как значительная часть времени при этом тратится на перекачку кодов и данных из оперативной памяти на диск и обратно. Размер страничного файла в современных ОС является настраиваемым параметром, который выбирается администратором системы для достижения компромисса между уровнем мультипрограммирования и быстродействием системы.