**Организация виртуальной памяти**

**Виртуа́льная па́мять** — технология управления памятью ЭВМ, разработанная для многозадачных ОС, которая позволяет выполнять процесс, только частично располагаемый в основной памяти. Таким образом, виртуальная память позволяет выполнять программы, размеры которых превышают размеры физического адресного пространства

**Виртуа́льная па́мять** — схема адресации памяти компьютера, при которой память представляется программному обеспечению непрерывной и однородной, в то время как в реальности для фактического хранения данных используются отдельные (разрывные) области различных видов памяти, включая кратковременную (**оперативную**) и долговременную (**жёсткие диски**, …).

Все методы управления памятью имеют одну и ту же цель - хранить в памяти **мультипрограммную смесь**, необходимую для мультипрограммирования

**Применение механизма виртуальной памяти позволяет:**

* **упростить адресацию памяти клиентским программным обеспечением;**
* **рационально управлять оперативной памятью компьютера (хранить в ней только активно используемые области памяти);**
* **изолировать процессы друг от друга (процесс полагает, что монопольно владеет всей памятью)**.

Виртуальная память (Virtual Memory) представляет собой программно-аппаратное средство расширения пространства памяти, предоставляемой программе в качестве оперативной. Эта память физически реализуется в **оперативной и дисковой памяти** под управлением соответствующей операционной системы. Требуется так организовать работу процессора с дисковой памяти, чтобы возникала иллюзия работы только с оперативной памятью.

Виртуальное пространство памяти разбито на страницы фиксированного размера, а в физической оперативной памяти в каждый момент времени присутствует только часть из них. Остальные страницы хранятся на диске, откуда операционная система может "подкачать" их в физическую память, предварительно выгрузив на диск часть неиспользуемых в данный момент модифицированных страниц.

Обращение процессора к ячейке виртуальной памяти, присутствующей в физической памяти, происходит обычным способом.

Если же затребованная область в данный момент не отображена в физической памяти, процессор вырабатывает исключение (внутреннее прерывание), по которому операционная система программно организует **замещение страниц**, называемое свопингом (**Swapping**).

Виртуальную память поддерживают процессоры, работающие в защищенном режиме, начиная с 80286, но реально ее широко стали использовать только в операционных системах и оболочках для 32-разрядных процессоров (80386 и далее).

Максимальный объем  виртуальной памяти определяется размером файла подкачки (***Swap File***), выделяемом на жестком диске для нужд виртуальной памяти.

*Немного истории*

*Общепринятая в настоящее время концепция виртуальной памяти появилась в те далекие времена, когда вычислительные системы, научившись выполнять одну программу стали переходить на новый режим работы - работу в многозадачном (мультипрограммном) режиме. Концепция виртуальной памяти позволила решить целый ряд актуальных вопросов организации вычислений.   
Встал вопрос о том, как разделить между разными задачами небольшой объем оперативной памяти, чтобы каждая задача выполнялась так же, как она выполнялась бы в однопрограммном режиме и как управлять при этом распределением памяти.*

*К этому времени у программистов уже накопился опыт выполнения задач, программный код которых не помещался в слишком маленькую оперативную память.*

*Программистам пришлось  делить такие программы на части и затем  выполнять части программы независимо, организуя оверлейные структуры, которые загружались в основную память и выгружались из нее под управлением программы пользователя.*

*Система  организации виртуальной памяти освободила программистов от этой работы. Она автоматически управляет обменом между двумя уровнями иерархии памяти: основной памятью и внешней (дисковой) памятью.*

В многопрограммном режиме работы компьютер выполняет множество задач, каждой из которых отводится свое адресное пространство в оперативной памяти. Поэтому необходим механизм разделения небольшой физической памяти между различными задачами. Виртуальная память делит физическую память на блоки и распределяет их между различными задачами. При этом обеспечивается иллюзия того, что каждая задача и в самом деле исполняется на машине с огромным количеством реальной оперативной памяти.

**Системы виртуальной памяти можно разделить на два класса:** системы с **фиксированным размером блоков, называемых страницами**, и системы с **переменным размером блоков, называемых сегментами.**

Очевидно, что для реализации возможности автоматического выделения памяти каждой задаче в процессе ее выполнения (динамически), необходимо обеспечить эту возможность аппаратно. Для того, чтобы программа могла действительно воспользоваться виртуальной памятью (иллюзией очень большого количества  реальной оперативной памяти) требуется еще и поддержка программного обеспечения.

Задача управления виртуальной памятью возлагается на ОС. Различные ОС по-разному решают вопросы управления памятью.

Механизмы управления памятью решают две главные задачи:

* Трансляцию виртуального адресного пространства процесса на физическую память. Это позволяет ссылаться на конкретные адреса физической памяти потокам процесса, работающим в  виртуальном адресном пространстве.
* Подкачку части содержимого памяти с диска, когда потоки пытаются задействовать больший объем физической памяти, чем тот, который имеется в наличии, и выгрузку страниц обратно по мере необходимости.

Виртуальная память может использовать страничную и сегментную адресации.

При использовании ***страничной адресации*** в специальные сегментные регистры процессора загружаются селекторы, базовые адреса которых равны нулю, а размер сегмента составляет 4 Гбайта. И хотя для микропроцессора  механизм сегментной адресации продолжает работать, основным механизмом формирования адреса становится страничная адресация.

Такая модель памяти и называется плоской (**FLAT**). Логическая адресация в такой модели определяется только  ***смещением***. Для программиста область памяти, адресуемая смещением, находится в полном распоряжении. Это и есть логический адрес, который преобразуется ОС в физический.

**Страничный механизм является составной частью системы виртуальной памяти.** Страничное преобразование позволяет сохранить линейность адресов для прикладных программ, несмотря на то, что  их работающие страницы располагаются в оперативной памяти в произвольном порядке. Программы  работают со своим адресным пространством, так словно у каждой из них есть своя собственная линейно адресуемая оперативная память.

Применительно к виртуальной памяти различают три адресных пространства: ***логическое, линейное и физическое***.

***Логический адрес*** формируется при обращении к виртуальному адресному пространству.

Блок сегментации транслирует логический адрес в линейный.

***Физический  адрес*** образуется после преобразования линейного адреса блоком страничной переадресации.

И так, выполняемая программа загружается в оперативную память не целиком, а отдельными страницами (рис1.). Страница, с которой работает процессор называется активной.

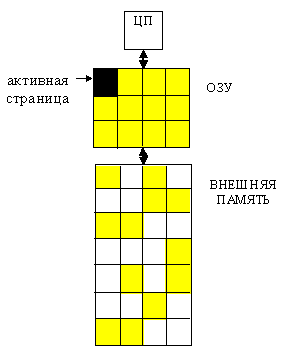


Рис.1.  Отражение страниц дискового пространства  
 на пространство оперативной памяти.

В случае динамического распределения памяти (по запросу программы), адрес содержит номер виртуальной страницы и номер слова в странице. Для защиты памяти от записи и чтения другими процессами необходима соответствующая схема трансляции адреса, которая разрешает доступ к данной области памяти только определенной задаче.

Соответствие между физическим положением данных в памяти и адресом существует только для основной (оперативной) памяти, виртуальная память использует всю иерархическую структуру памяти как непрерывно адресуемую.

Для переадресации используется глобальная (GDT) и локальная (LDТ)  таблицы переадресации. 

**Динамическая трансляция виртуального адреса в реальный**

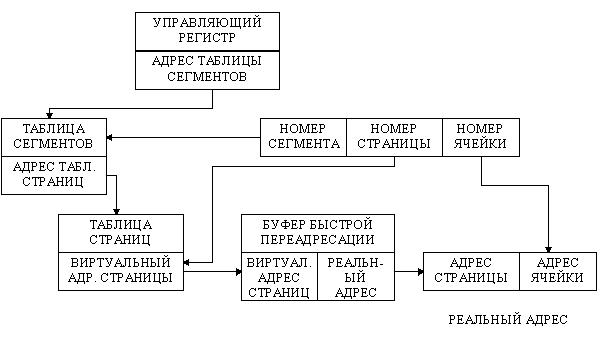


Рис.2. Преобразование виртуального адреса

Рассмотрим систему переадресации с сегментацией памяти (рис.2.).

В глобальной таблице компьютера хранятся сведения о нахождении таблицы сегментов каждой задачи. Адрес таблицы сегментов для выполняемой задачи загружается в управляющий регистр. По адресу таблицы и номеру сегмента виртуального адреса определяется адрес таблицы страниц. По адресу таблицы страниц и номеру страницы виртуального адреса определяется адрес страницы в таблице страниц (виртуальный адрес, который преобразуется в реальный в буфере быстрой переадресации). Из буфера быстрой переадресации считывается реальный адрес страницы. Полный реальный адрес равен реальному адресу страницы плюс номер ячейки памяти (смещение в странице).

Если разбиение на сегменты не используется, то для трансляции логического адреса используется двухуровневая таблица страниц. Виртуальный адрес интерпретируется как совокупность 3 элементов: указателя каталога страниц, указателя таблицы страниц и указателя слова на странице.

У каждого процесса (каждой задачи) есть один каталог страниц, который содержит адреса всех таблиц страниц для данного процесса. Указатель каталога страниц применяется для поиска каталога, из каталога определяется указатель таблицы страниц для данного процесса, по указателю таблицы страниц находится нужная таблица, из которой читается адрес, содержащий указатель для требуемой страницы (ее физический адрес). Остается определить реальный адрес данных, расположенных на этой странице.  Указатель слова  позволяет найти конкретный адрес на физической странице.

Но что делать, если количество  страниц во всех линейных адресных пространствах выполняемых процессов превышает объем оперативной  памяти и поэтому невозможно поставить в соответствие каждой из них свою страницу реальной памяти? Для решения этой проблемы дополнительно ведется учет присутствия страницы в физической памяти ‑ вводится дескриптор страницы, где предусмотрен флаг (*бит) реального присутствия* страницы в оперативной памяти. Наличие такого бита позволяет отметить часть страниц, непомещающихся в физической памяти, как временно отсутствующие.

Если происходит обращение к такой странице, то процессор формирует прерывание. Получив прерывание по отсутствию страницы, операционная система может записать одну из присутствующих в памяти страниц на диск и отметить ее как временно отсутствующую во всех адресных пространствах, в которых она была видна. На освободившееся место с диска считывается та страница, из-за которой произошло прерывание. Заполняются все таблицы, по которым ведется поиск данной страницы и в дескрипторе страницы устанавливается флажок ее реального присутствия в оперативной памяти.

После возврата из программы - обработчика прерывания процессор повторит попытку доступа к памяти, но теперь нужная страница уже отмечена как присутствующая и прерывания не произойдет. Если  вновь потребуется та страница, которая была записана на диск, то снова произойдет прерывание, ведь эту страницу пометили как временно отсутствующую в памяти. В ответ на это прерывание на диск будет перемещена еще какая-нибудь страница, а на ее место с диска будет загружена требуемая.

Этот процесс называется страничным обменом или свопингом (***paging*** или ***swapping***). Страницы, помещаемые на диск хранятся в специальном файле, файле подкачки (***swap-файле***). Взаимодействие с файлом подкачки происходит значительно медленнее, чем с оперативной памятью компьютера, поэтому оптимизация работы с этим файлом зачастую дает возможность увеличить скорость работы системы памяти. ***Теоретически, наиболее оптимальным считается объем файла подкачки, превышающий объем установленной на компьютере оперативной памяти примерно в два раза.***

Если страница не изменялась с момента последнего считывания с диска, то при ее замене другой страницей нет необходимости повторно записывать ее на диск. Можно просто пометить ее как отсутствующую. Для слежения за изменением страницы современные процессоры поддерживают еще один флаг в дескрипторе страницы. Другой важный флаг, который поддерживается аппаратурой процессора - это флаг обращения. Он устанавливается всегда, когда происходит обращение к странице на чтение или запись. Операционная система может время от времени просматривать страничные таблицы, проверять и сбрасывать этот бит, выявляя те страницы, которые часто используются программами.

**стратегия страничного обмен**а

Для того, чтобы уменьшить количество дисковых операций, операционная система использует ту или иную стратегию страничного обмена. Оптимальная стратегия  состоит в том, чтобы возвращать на диск ту страницу, которая дольше всего не понадобиться программам.

Одной из лучших стратегий страничного обмена является **LRU** **(Least Recently Used)**. Алгоритм **LRU** предполагает, что на диск надо переносить ту страницу, которая дольше всего не использовалась. Существуют и более сложные алгоритмы, но обычно они базируются на LRU.

**page prefetch**

В некоторых случаях операционная система заранее знает, что определенные страницы, которые сейчас находятся на диске, понадобятся ей (или прикладным программам) в ближайшее время. Тогда она может заранее перенести их в оперативную память, не дожидаясь прерываний по отсутствию страницы. Такая операция называется страничной предвыборкой (**page prefetch**). Аналогичный прием может быть использован и для записи на диск пассивных страниц раньше, чем будет затребовано место, занимаемое этой страницей.