Архитектура ЭВМ и систем

Лекция № 9

Управление памятью.

Виртуальная память.

План лекции

1. Динамическое распределение памяти
2. Сегментная организация памяти
3. Страничная организация памяти
4. Сегментно-страничная организация памяти

Динамическое распределение памяти

Непосредственно адресуемая процессором оперативная память (ОП) имеет сравнительно небольшой объём, что в большинстве случаев не позволяет её вместить все команды и данные исполняемых программ. Обычно в памяти содержатся только их фрагменты. Все программы и данные хранятся в относительно дешёвой внешней памяти – ВЗУ.

Поскольку процессор не имеет прямого доступа к ВЗУ, команды и данные исполняемой программы предварительно надо переписать в ОП. Ещё больше трудностей возникает в многозадачном режиме работы ВМ, когда в ОП одновременно должны храниться команды и данные нескольких задач.

На ОС возлагаются функции по распределению ресурса ОП между отдельными программами. Так как заранее неизвестно, какие программы, и в какой комбинации могут выполняться, распределение памяти между программами должно осуществляться динамически в ходе вычислений.

Процедура, при которой ОС активным частям программ выделяет определённые области ОП и осуществляет привязку адресов загружаемым программам к конкретным адресам физикой ОП, называется ***динамическим распределением памяти***.

Динамическое распределение памяти

В основе всех известных методов динамического распределения памяти лежат два положения:

1. каждому заданию (процессу) необходимо выделять непрерывную и перемещаемую область памяти;
2. должна обеспечиваться возможность попеременной загрузки заданий в ОП.

Для организации обменов между ВЗУ и ОП внешняя память представляется в виде набора частей (блоков) отдельных программ.

Если требуемый фрагмент программы отсутствует в ОП, процессор обращается к ОС, которая, используя специальные процедуры, считывает из внешней памяти в ОП соответствующий блок.

Если для требуемого фрагмента в ОП недостаточно места, то ОС предварительно освобождает такое место, пересылая неиспользуемые блоки из ОП в ВЗУ.

Динамическое распределение памяти

Перемещаемость программ при их размещении в ОП можно обеспечить, если для адресации операндов внутри каждого блока использовать метод базирования.

Этот метод предполагает, что все программы представлены в относительных адресах с началом в нулевой ячейке. Тогда адрес операнда определяется базовым адресом (сохраняется в одном из сегментных регистров процессора) и смещением относительно этого базового адреса (указывается или вычисляется в команде).

Путём изменения содержимого сегментных регистров программы можно перемещать в ОП, не нарушая их внутренней адресации.

Попеременная загрузка заданий (программ) в современных ВМ осуществляется путём ***свопинга*** (от англ. *swap* – обмен) между ОП и ВЗУ.

В мультипрограммных системах такой обмен осуществляется автоматически (без участия программиста), с помощью специальных программно-аппаратных средств под управлением ОС.

Динамическое распределение памяти

Динамическое распределение памяти тесно связано с понятием виртуальной памяти.

Под ***виртуализацией памяти*** понимается метод автоматического управления иерархической памятью, при котором программисту кажется, что он имеет дело с единой памятью большой ёмкости и высокого быстродействия. Эту память называют виртуальной (кажущейся). Впервые идея виртуализации памяти появилась в 1959 г.

Использование виртуальной памяти позволяет писать программы, размер которых превосходит имеющуюся ОП.

При этом с помощью виртуальных (логических) адресов обеспечивается адресация всего адресного пространства ВМ. Систему виртуальной памяти можно представить в виде одноуровневой логической и двухуровневой (ОП и ВЗУ) физической памяти.

Адреса, к которым программа может обратиться, образуют виртуальное адресное пространство системы, а реальные адреса – физическое адресное пространство, причём линейное, состоящее из *N* ячеек разрядностью *n*.

Программа пишется в виртуальных адресах, но для её выполнения требуется, чтобы обрабатываемые команды и данные находились в ОП. Для этого необходимо, чтобы каждому виртуальному адресу соответствовал физический адрес.

Динамическое распределение памяти

Все операции по управлению виртуальной памятью, динамическому распределению памяти и преобразованию адресов в ВМ выполняются автоматически.

В современных процессорах некоторые из указанных функций реализуются с помощью специального контроллера управления памятью *MMU (Memory Management Unit)*.

Среди моделей виртуальной памяти можно выделить:

1. сегментную,
2. страничную и
3. сегментно-страничную организацию виртуальной памяти.

Сегментная организация памяти

Сегментирование – это разделение памяти на логические блоки произвольной длины.

Логическое пространство задачи обычно представляется в виде нескольких сегментов. Каждый сегмент имеет имя, в соответствии с которым ОС при распределении памяти назначает базовый адрес. Количество сегментов определяет пользователь при подготовке программы.

Для раздельного хранения команд, данных и стековых данных выделяются специальные сегменты: сегмент кода, сегмент данных, сегмент стека.

Максимальное количество сегментов (*s*) определяется разрядностью поля команды, которое используется для задания номера сегмента. Например, в процессорах Pentium разрядность индекса селектора сегмента равна 13, что соответствует 8192 сегментам.

Предельный размер (длина) каждого сегмента определяется разрядностью внутрисегментного смещения. Например, для процессоров Pentium длина сегмента ограничивается величиной Гбайта. Длина сегмента может меняться во время выполнения программы. Сегмент может переполниться, но это случается редко, поскольку длина сегментов достаточно большая.

Сегментная организация памяти

Сегменты всех активных задач, определённых в виртуальном адресном пространстве, размещаются в ВЗУ. Поскольку объём ОП относительно невелик, то не все сегменты могут в неё поместиться.

Соответствие между виртуальной памятью и физической ОП устанавливается ОС с помощью специальной таблицы соответствия (дескрипторной таблицы), которую ОС формирует всякий раз, когда в распределении памяти происходят изменения (например, при загрузке новой задачи).

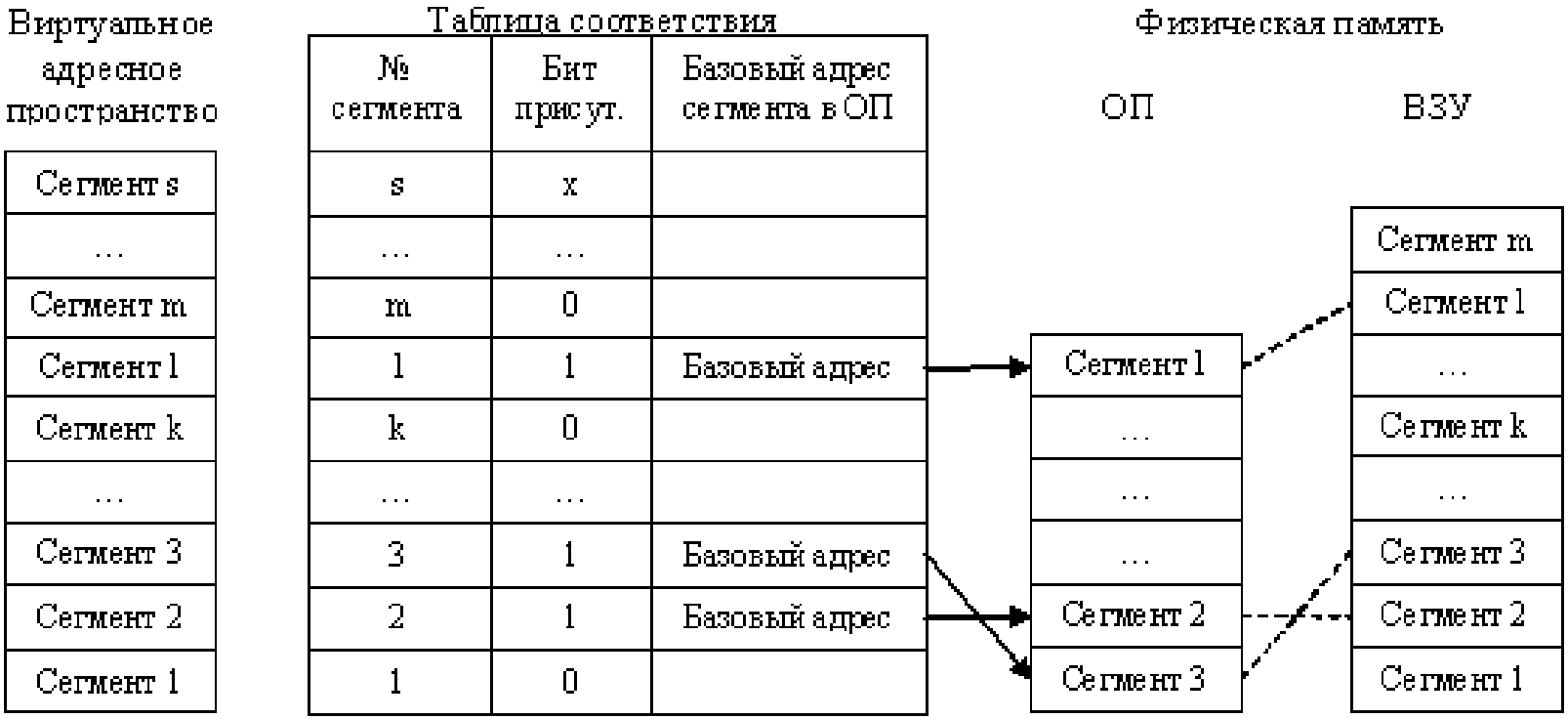
Размер таблицы определяется максимальным числом сегментов виртуального адресного пространства.

Каждая строка таблицы (дескриптор) содержит информацию о номере и базовом адресе сегмента в ОП.

Базовые адреса назначаются при загрузке сегмента и при любом изменении в распределении памяти.

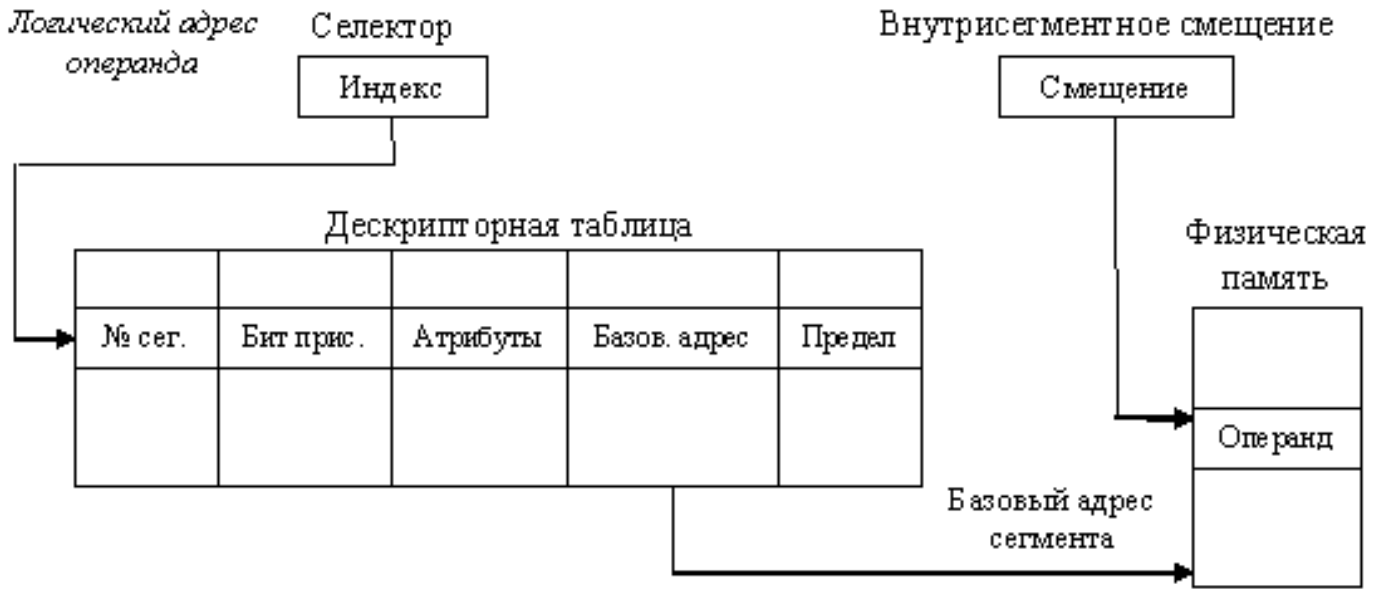
Доступ к элементам дескрипторной таблицы осуществляется с помощью указателей (селекторов), для размещения которых используются сегментные регистры.

Сегментная организация памяти



Сегментное распределение памяти

Сегментная организация памяти



Вычисление физического адреса в сегментной памяти

Сегментная организация памяти

При обращении к сегменту памяти, присутствующему в ОП, в выполнении команд нет никаких отличий по сравнению с обычной физической памятью.

По логическому адресу операнда, состоящему из двух частей (селектора – номера виртуального сегмента и смещения внутри сегмента) выбирается строка в дескрипторной таблице сегментов, где считывается информация о начальном адресе сегмента в ОП.

К этому адресу добавляется внутрисегментное смещение (вторая часть логического адреса), что и даёт требуемый физический адрес искомого операнда в ОП.

Сегментная организация памяти

Если требуемый сегмент отсутствует в ОП, то возникает прерывание, по которому процедура обработки этого прерывания переписывает в ОП нужный сегмент.

При наличии свободного места для загрузки сегмента никаких проблем не возникает, и после загрузки нового сегмента в ОП осуществляется повторный запуск команды обращения к памяти.

Если в ОП недостаточно места для записи отсутствующего сегмента, то соответствующая процедура ОС сначала освобождает требуемое место, переписывая на диск один или несколько неиспользуемых в данный момент сегментов. Для этого наиболее часто используется алгоритм LRU (Last Recently Used), обеспечивающий замену сегмента, к которому не было обращения самое продолжительное время.

ОС использует бит доступа *A* для определения приблизительного времени последнего использования сегмента. Используя механизм свопинга, ОС создаёт иллюзию, что все сегменты программы постоянно находятся в ОП.

Сегментная организация памяти

Способ реализации виртуальной памяти, при котором целые сегменты загружаются и удаляются из ОП только при необходимости, называют ***сегментацией с подкачкой сегментов*** или ***сегментацией с вызовом сегментов по требованию***.

Особенностью использования сегментированной памяти является то, что после многократных свопингов сегментов в ОП могут образовываться свободные участки памяти слишком малого размера и неудобные для использования. Это явление называется ***внешней фрагментацией*** (неиспользуемое пространство попадает не в сегменты, а в пустоты между ними).

Чтобы избежать подобной ситуации, ОС выполняет уплотнение сегментов.

При первом способе каждый раз при появлении пустого пространства следующие сегменты перемещаются ближе к адресу 0, удаляя таким образом это пустое пространство.

При втором способе уплотнение выполняется только тогда, когда на долю пустот приходится больше некоторого процента от общего объёма памяти.

Сегментная организация памяти

Использование сегментной модели в значительной степени упрощает изолирование программных блоков отдельных задач друг от друга в мультизадачной системе.

Для каждой задачи обычно выделяется собственная локальная память, и одновременно задача может разделять с другими задачами совместную память, которую называют глобальной.

Соответствие между логическими и физическими адресами в мультизадачной системе устанавливаются с помощью глобальной GDT (общей для всех задач) и локальных LDT (отдельных для каждой задачи) дескрипторных таблиц.

Общий объём адресуемой виртуальной памяти отдельной задачи определяется разрядностью адресных полей её логического адреса.

Например, в процессорах Pentium пространство виртуальных адресов задачи не может превышать 64 Тбайт.

Страничная организация памяти

При страничной организации памяти виртуальное и физическое адресные пространства разбиваются на блоки фиксированного размера – страницы. Размер страниц обычно выбирается равным 4 Кбайта (реже 4 Мбайта).

Блок ОП, соответствующий виртуальной странице, часто называют ***страничным кадром*** или ***фреймом***.

Страницам виртуальной и физической памяти присваивают номера.

Страничная организация памяти создаёт иллюзию линейной ОП такого же размера, как и адресное пространство программы.

В модели виртуальной памяти со страничной организацией соответствие между виртуальными и физическими страницами устанавливается в процессе распределения памяти при заполнении специальной таблицы страниц, каждая строка которой (страничный дескриптор) содержит базовый адрес страничного кадра в ОП, а также биты управления виртуальной памятью и биты защиты информации на странице.

Страничная организация памяти

При страничной организации памяти виртуальное и физическое адресные пространства разбиваются на блоки фиксированного размера – страницы. Размер страниц обычно выбирается равным 4 Кбайта (реже 4 Мбайта).

Блок ОП, соответствующий виртуальной странице, часто называют ***страничным кадром*** или ***фреймом***.

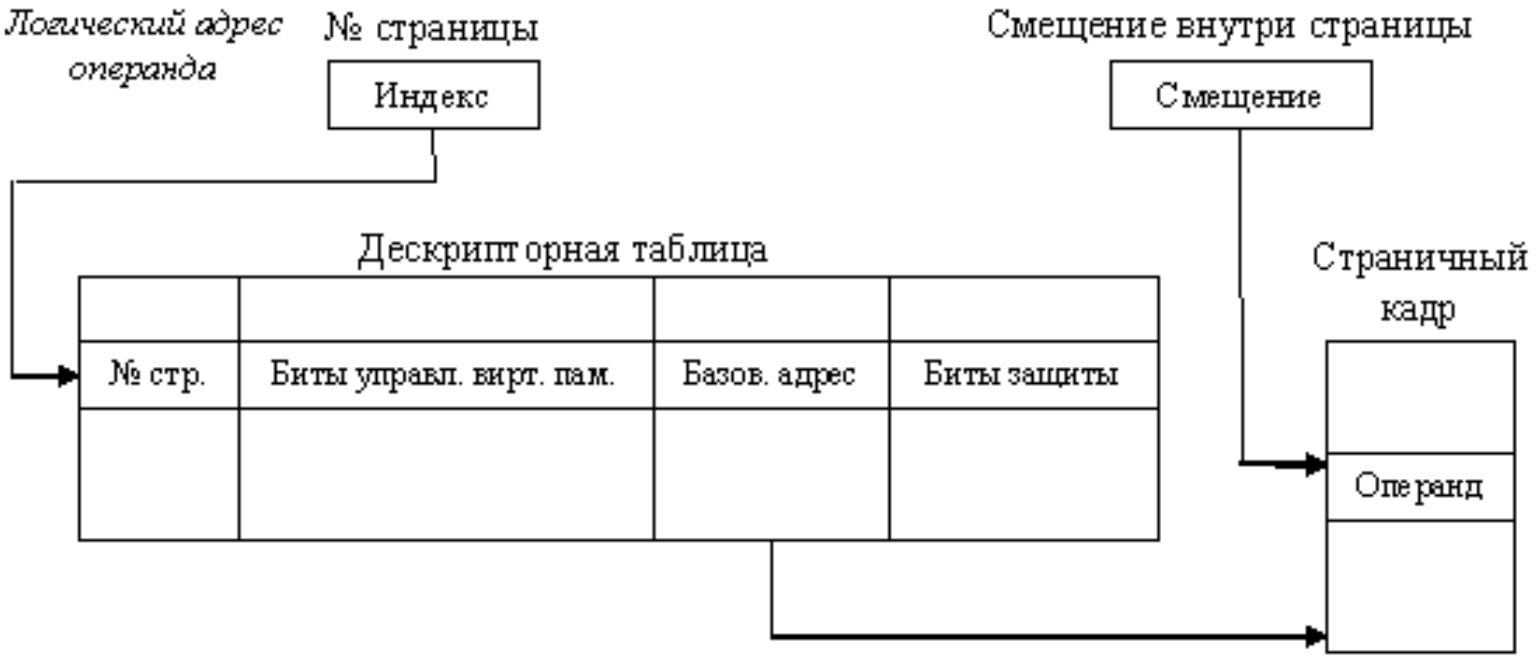
Страницам виртуальной и физической памяти присваивают номера.

Страничная организация памяти создаёт иллюзию линейной ОП такого же размера, как и адресное пространство программы.

В модели виртуальной памяти со страничной организацией соответствие между виртуальными и физическими страницами устанавливается в процессе распределения памяти при заполнении специальной таблицы страниц, каждая строка которой (страничный дескриптор) содержит базовый адрес страничного кадра в ОП, а также биты управления виртуальной памятью и биты защиты информации на странице.

Страничное распределение памяти аналогично сегментному распределению памяти (слайд 10), только вместо сегментов используются страницы.

Страничная организация памяти



Вычисление физического адреса в страничной памяти

Страничная организация памяти

При обращении к памяти номер виртуальной страницы извлекается из виртуального адреса и используется как индекс для поиска нужного дескриптора в таблице страниц.

Если страница присутствует в ОП, то из соответствующего дескриптора извлекается базовый адрес соответствующего страничного кадра, который совместно со смещением в виртуальном адресе, определяет физический адрес требуемого операнда.

Если требуемая страница отсутствует в ОП, то формируется прерывание, по которому ОС загружает из ВЗУ отсутствующую страницу. После этого управление возвращается процессору, который повторно выполняет команду обращения к памяти. При необходимости освобождения памяти некоторые страницы выгружаются из ОП в ВЗУ.

Для этого применяется алгоритм LRU (выгружается дольше всего неиспользовавшаяся страница) или алгоритм FIFO (выгружается страница, которая была загружена раньше всех, независимо от того, когда в последний раз производилось обращение к ней).

Страничная организация памяти

Метод работы с виртуальной памятью, при котором страницы переносятся в ОП только в случае необходимости, называется ***вызовом страниц по требованию***.

В отличие от сегментов, страницы не имеют прямой связи с логической структурой программы. Трансляция виртуального адреса в физический выполняется автоматически и быстрее, чем при сегментной организации виртуальной памяти.

Фиксированная длина страниц позволяет решить проблему внешней фрагментации и упрощает распределение памяти. При необходимости загрузки новой страницы в ОП её можно поместить либо в незанятый страничный кадр, либо вытеснить другую страницу, освободив таким образом требуемое место.

В любом случае, не требуется по-новому располагать страницы в ОП.

Страничная организация памяти

Если в сегментной памяти сегменты загружаются целиком, то страничная организация позволяет сократить объём передаваемой между ОП и ВЗУ информации за счёт того, что страницы программы могут не загружаться в ОП, пока они действительно не понадобятся.

Сначала в ОП загружается начальная страница программы, и ей передаётся управление. Если в процессе выполнения программы потребуется выборка операндов из другой страницы, ОС загрузит отсутствующую страницу.

Особенно заметны преимущества страничной организации памяти при реализации мультизадачных систем.

В таких системах при загрузке новой задачи её страницы могут быть направлены в любые свободные в данный момент страничные кадры независимо от того, расположены они подряд или нет.

Страничная организация памяти

Однако страничная память тоже подвержена фрагментации. Если пользовательская программа и данные занимают ровно целое число страниц, то при их загрузке в память свободного места там не остаётся.

Но если они не занимают ровно целое число страниц, на последней странице останется неиспользованное пространство, и в ОП будут появляться свободные участки, бесполезно занимая место.

Эта проблема получила название ***внутренней фрагментации***, так как неиспользуемое пространство является внутренним по отношению к странице.

Чтобы свести к минимуму объём бесполезного пространства, страницы должны быть небольшими, однако для их хранения потребуется большая таблица страниц.

Сегментно-страничная организация памяти

При сегментно-страничной организации память разбивается на сегменты, а сегменты, в свою очередь, разделены на страницы фиксированной длины.

При этом размер сегмента выбирается не произвольно, а задаётся кратным размеру страницы.

Сегмент может содержать произвольное, но обязательно целое число страниц, даже если одна из страниц заполнена частично.

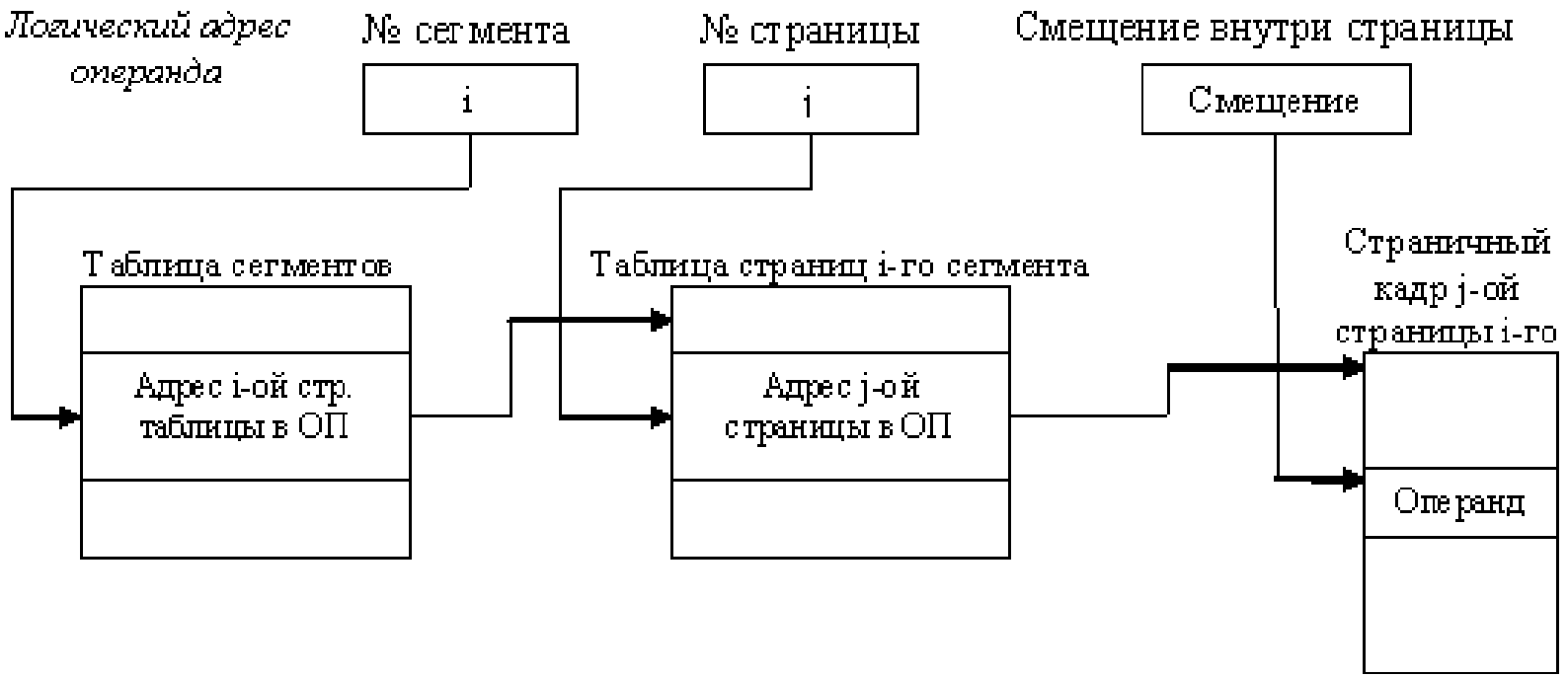
Для преобразования логических адресов в физические используются две таблицы соответствия.

Каждый сегмент имеет отдельную таблицу страниц. В дескрипторной таблице сегментов перечисляются все сегменты с указанием начальных адресов страничных таблиц, относящихся к каждому сегменту.

Каждая таблица страниц, в свою очередь, определяет положение каждой из страниц сегмента в памяти.

Страницы сегмента могут располагаться не подряд – часть их может находиться в ОП, остальные – в ВЗУ.

Сегментно-страничная организация памяти



Вычисление физического адреса при сегментно-страничной организации памяти

Сегментно-страничная организация памяти

Виртуальный адрес операнда состоит из 3-х составляющих: номера сегмента, номера страницы в этом сегменте и смещения внутри данной страницы.

Для получения физического адреса операнда сначала по номеру сегмента в таблице сегментов определяется начальный адрес соответствующей ему страничной таблицы.

Затем по номеру страницы в таблице страниц определяется начальный адрес конкретной страницы в ОП.

К базовому адресу страницы в ОП добавляется смещение внутри страницы.

Сам процессор только предоставляет аппаратные средства поддержки виртуальной памяти, а их реальное использование зависит от корректного построения ОС.