Управление памятью

Управление памятью

- **Оперативная память** – важнейший ресурс вычислительной системы, требующий управления со стороны ОС. Причина – процессы и потоки хранятся и обрабатываются в оперативной памяти.
- Память распределяется между приложениями и модулями самой операционной системы.
- **Функции ОС по управлению оперативной памятью**:
  - Отслеживание наличия свободной и занятой памяти;
  - Контроль доступа к адресным пространствам процессов;
  - Вытеснение кодов и данных из оперативной памяти на диск, когда размеров памяти недостаточно для размещения всех процессов, и возвращение их обратно;
  - Настройка адресов программы на конкретную область физической памяти;
  - Защита выделенных областей памяти процессов от взаимного вмешательства.
- Часть ОС, которая отвечает за управление памятью, называется **менеджером памяти**.

Физическая организация памяти

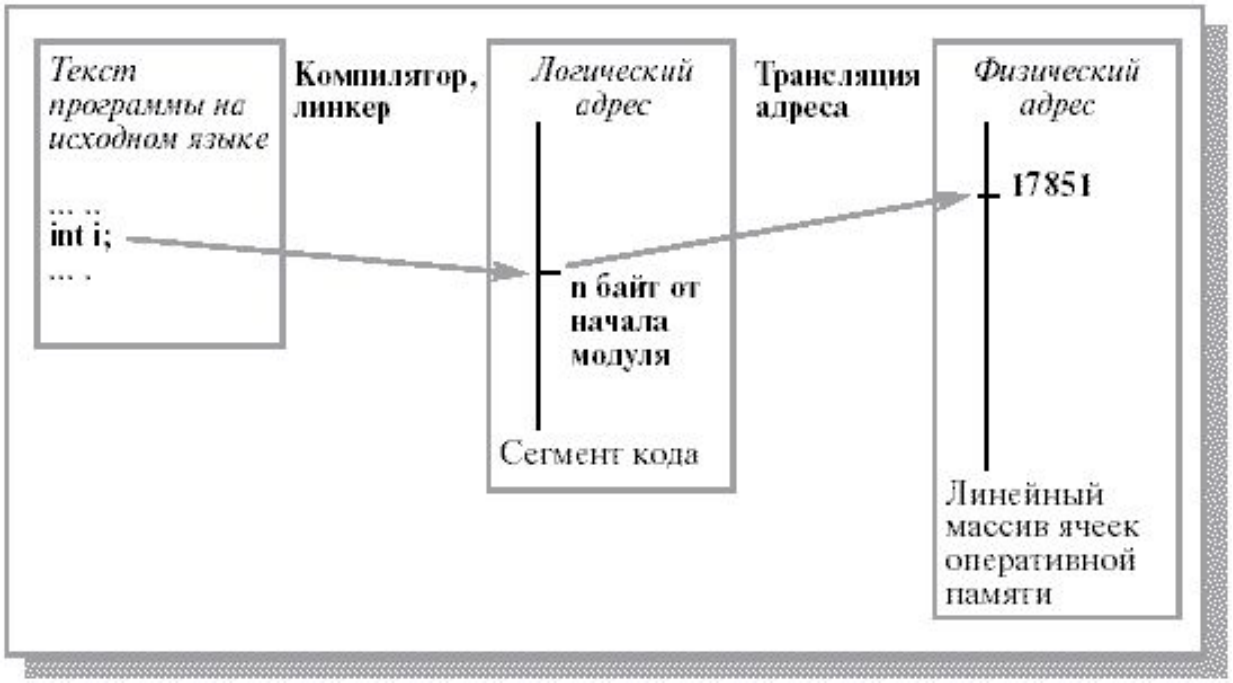
- Запоминающие устройства компьютера разделяют, как минимум, на два уровня: *основную* (главную, *оперативную*, *физическую*) и вторичную (внешнюю) память.
- *Основная память* представляет собой упорядоченный массив однобайтовых ячеек, каждая из которых имеет свой уникальный адрес (номер). Процессор извлекает команду из *основной памяти*, декодирует и выполняет ее. Для выполнения команды могут потребоваться обращения еще к нескольким ячейкам *основной памяти*.
- Вторичную память (это главным образом диски) также можно рассматривать как одномерное линейное *адресное пространство*, состоящее из последовательности байтов. В отличие от *оперативной памяти*, она является энергонезависимой, имеет существенно большую емкость и используется в качестве расширения *основной памяти*.

Иерархия памяти

Представление потоков в оперативной памяти

- Для идентификации переменных и команд программы используются разные типы адресов:
  - **Символьные** (имена переменных, функций и т.п.);
  - **Виртуальные** – условные числовые значения, вырабатываемые компиляторами;
  - **Физические** – адреса фактического размещения в оперативной памяти.

Связывание адресов



Виртуальное пространство

- Совокупность виртуальных адресов называется *виртуальным адресным пространством*. Диапазон возможных адресов виртуального пространства у всех процессов одинаков.
- Совпадение виртуальных адресов различных процессов не должно приводить к конфликтам и операционная система отображает виртуальные адреса различных процессов на разные физические адреса.
- Разные ОС по разному организуют виртуальное адресное пространство:
  - ***Линейная организация*** – пространство представляется непрерывной линейной последовательностью адресов (по другому плоская структура адресного пространства).
  - ***Сегментная организация*** – пространство разделяется на отдельные части. В этом случае, помимо линейного адреса, может быть использован виртуальный адрес (сегмент, смещение).

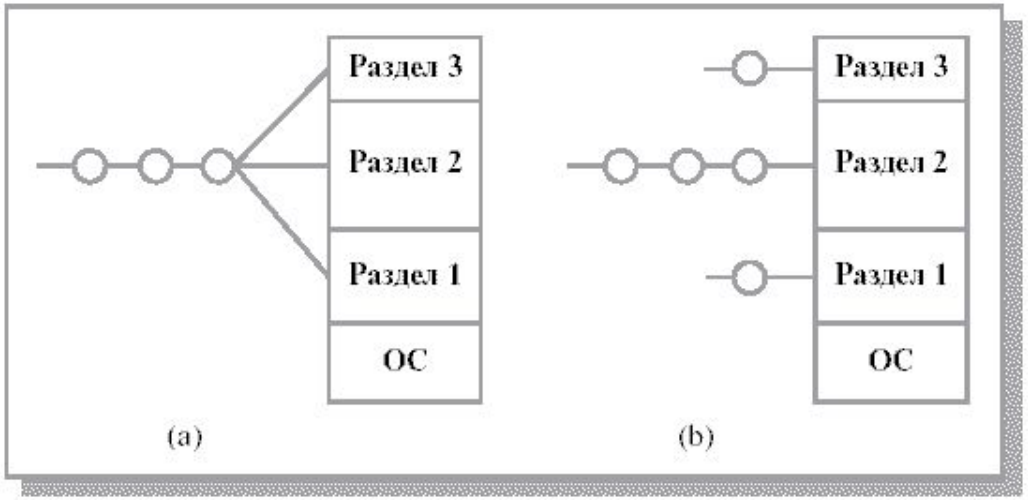
Виртуальное адресное пространство

- В виртуальном адресном пространстве выделяют две непрерывные части:
  - Системная – для размещения модулей общих для всей системы (размещаются коды и данные ядра ОС, другие служебные модули);
  - Пользовательская – для размещения кода и данных пользовательских программ.
- Системная область включает в себя область, подвергаемую страничному вытеснению, и область, на которую страничное вытеснение не распространяется. В последней располагаются системные процессы, требующие быстрой реакции или постоянного присутствия в памяти. Остальные сегменты подвергаются вытеснению, как и пользовательские приложения.

Алгоритмы распределения памяти

Схема с фиксированными разделами

- Схема основана на предварительном разбиении общего адресного пространства на несколько разделов фиксированной величины.
- Процессы помещаются в тот или иной раздел.
- Связывание физических и логических адресов процесса происходит на этапе его загрузки.

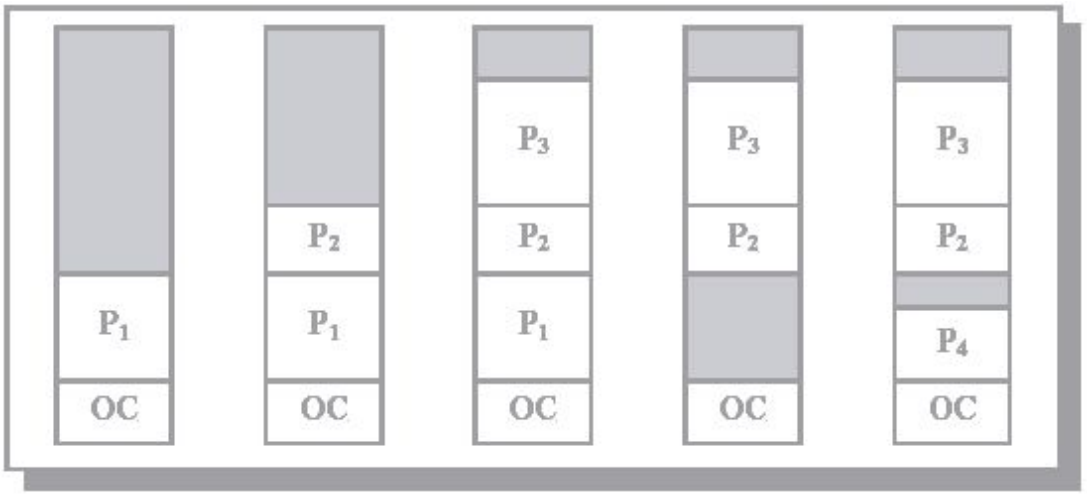


Динамическое распределение. Свопинг.

- В системах с разделением времени возможна ситуация, когда память не в состоянии содержать все пользовательские процессы.
- В таких случаях используется свопинг (swapping) – перемещению процессов из главной памяти на диск и обратно целиком. Частичная выгрузка процессов на диск осуществляется в системах со страничной организацией (paging).
- Выгруженный процесс может быть возвращен в то же самое *адресное пространство* или в другое. Это ограничение диктуется методом *связывания*. Для схемы *связывания* на этапе выполнения можно загрузить процесс в другое место памяти.

Схема с переменными разделами

- Типовой цикл работы менеджера памяти состоит в анализе запроса на выделение свободного участка (раздела), выборе его среди имеющихся в соответствии с одной из стратегий (первого подходящего, наиболее подходящего и наименее подходящего), загрузке процесса в выбранный раздел и последующих изменениях таблиц свободных и занятых областей.
- Аналогичная корректировка необходима и после завершения процесса. *Связывание адресов* может осуществляться на этапах загрузки и выполнения.



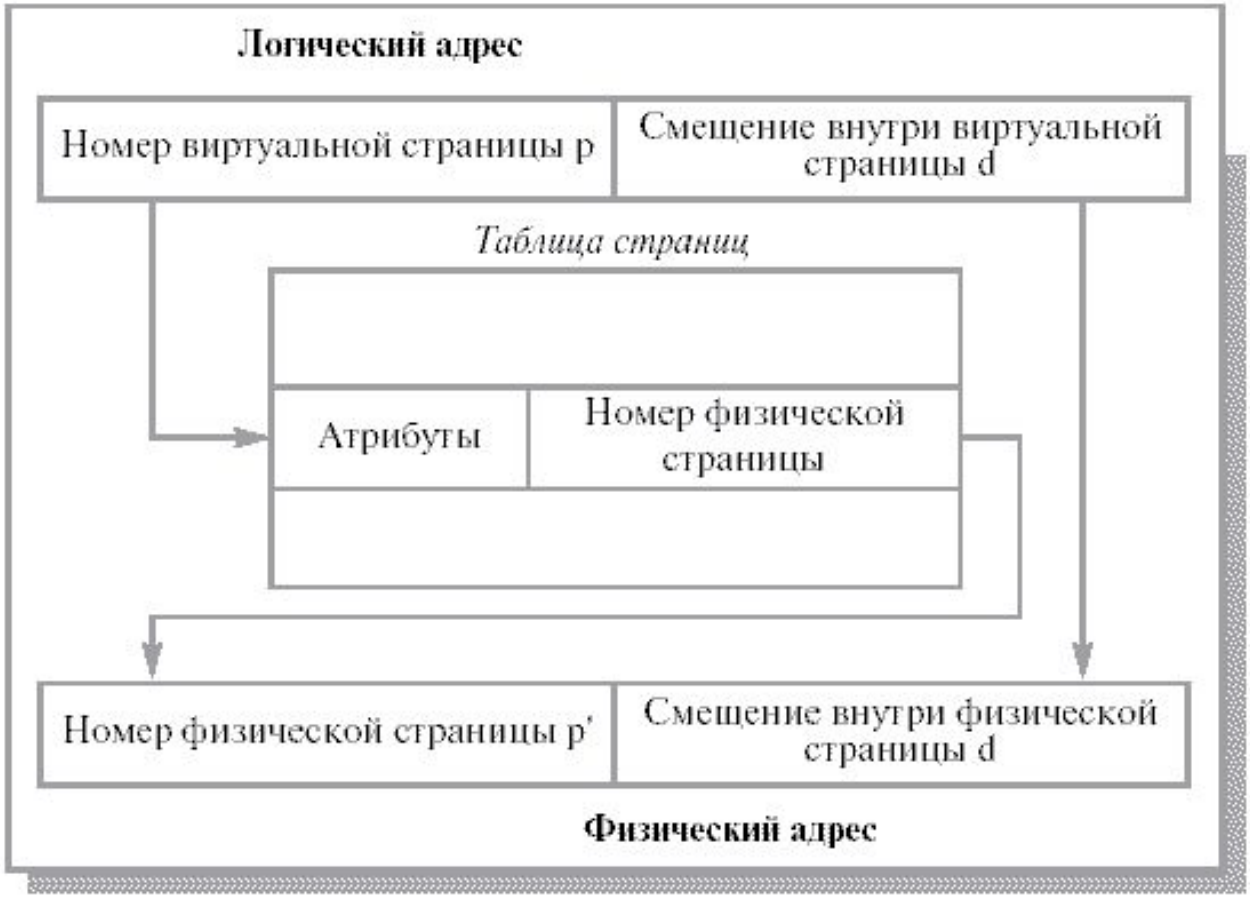
Страничная организация

- В случае страничной организации памяти (или paging) как логическое *адресное пространство*, так и физическое представляются состоящими из наборов блоков или *страниц* одинакового размера.
- При этом образуются логические *страницы* (page), а соответствующие единицы в *физической памяти* называют страничными кадрами (page frames). *Страницы* (и страничные кадры) имеют фиксированную длину, обычно являющуюся степенью числа 2, и не могут перекрываться.
- Каждый кадр содержит одну *страницу* данных. При такой организации *внешняя фрагментация* отсутствует, а потери из-за *внутренней фрагментации*, поскольку процесс занимает целое число *страниц*, ограничены частью последней *страницы* процесса.

Связь логического и физического адресов

- Логический адрес в страничной системе – упорядоченная пара (p,d), где p – номер *страницы* в виртуальной памяти, а d – смещение в рамках *страницы* p, на которой размещается адресуемый элемент.
- Разбиение *адресного пространства* на *страницы* осуществляется вычислительной системой незаметно для программиста.
- Адрес является двумерным лишь с точки зрения операционной системы, а с точки зрения программиста *адресное пространство* процесса остается линейным.

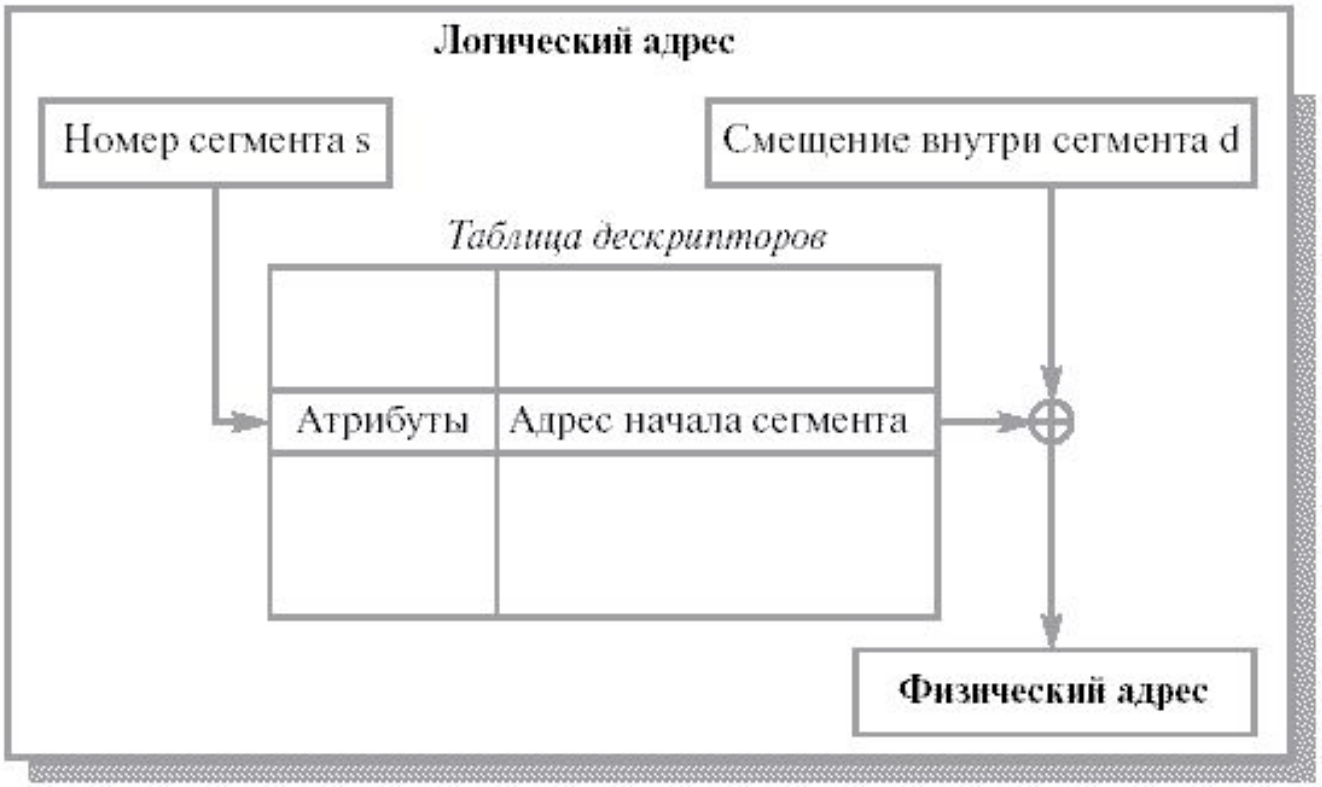
Схема адресации при страничной организации



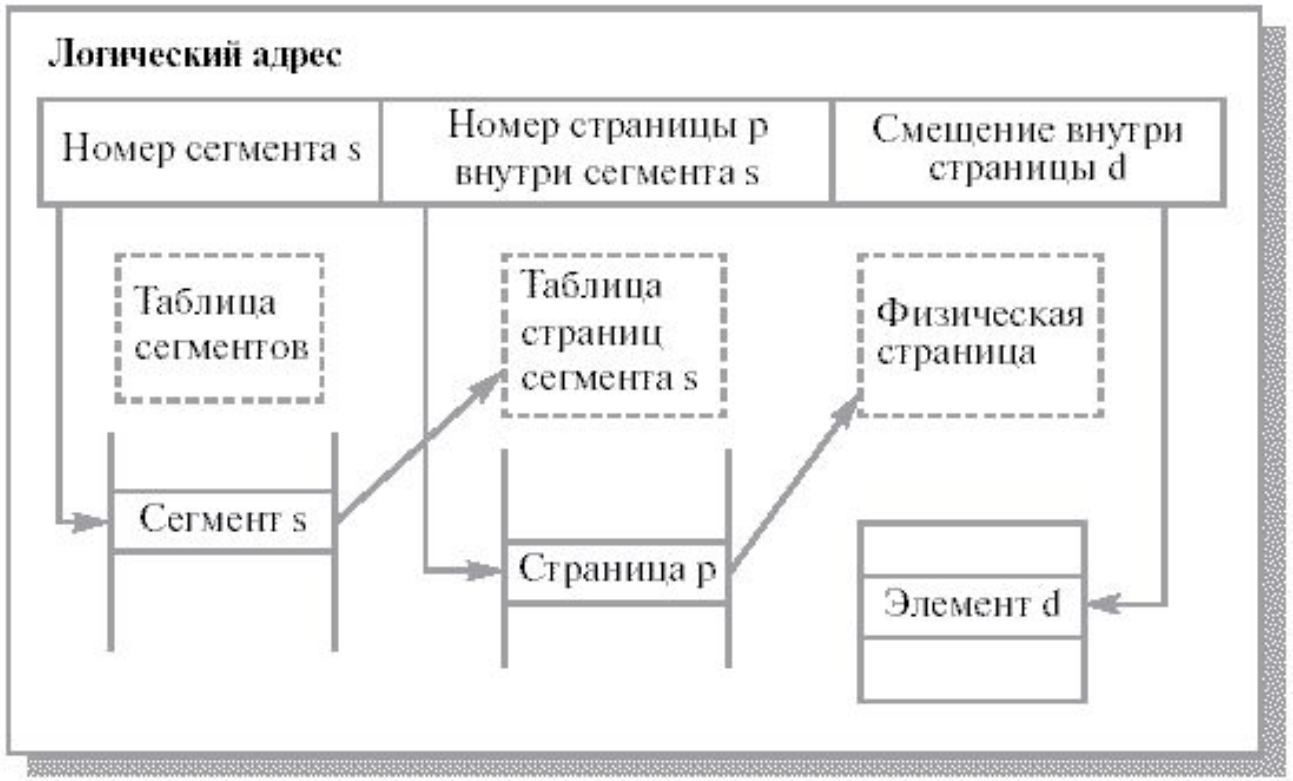
Сегментная и сегментно-страничная организация памяти

- *Сегменты*, в отличие от *страниц*, могут иметь переменный размер.
- Каждый *сегмент* – линейная последовательность адресов, начинающаяся с 0. Максимальный размер *сегмента* определяется разрядностью процессора (при 32-разрядной адресации это 232 байт или 4 Гбайт).
- Размер *сегмента* может меняться динамически (например, *сегмент* стека). В элементе таблицы *сегментов* помимо физического адреса начала *сегмента* обычно содержится и длина *сегмента*.
- Логический адрес – упорядоченная пара v=(s,d), номер *сегмента* и смещение внутри *сегмента*.

Преобразование логического адреса при сегментной организации



Формирование адреса при странично-сегментной организации памяти



Виртуальная память

- Разработчикам программного обеспечения часто приходится решать проблему размещения в памяти больших программ, размер которых превышает объем доступной оперативной памяти.
- Развитие архитектуры компьютеров и расширение возможностей операционной системы по управлению памятью позволило переложить решение этой задачи на компьютер. Одним из подходов стало появление *виртуальной памяти* (virtual memory).

Концепция работы с виртуальной памятью

- Информация, с которой работает активный процесс, должна располагаться в оперативной памяти.
- В схемах *виртуальной памяти* у процесса создается иллюзия того, что вся необходимая ему информация имеется в основной памяти.
  - во-первых, занимаемая процессом память разбивается на несколько частей, например страниц;
  - во-вторых, логический адрес (логическая страница), к которому обращается процесс, динамически транслируется в физический адрес (физическую страницу);
  - и наконец, в тех случаях, когда страница, к которой обращается процесс, не находится в физической памяти, нужно организовать ее подкачку с диска.
- Для контроля наличия страницы в памяти вводится специальный *бит присутствия*, входящий в состав атрибутов страницы в *таблице страниц*.

Кэширование данных

- Для ускорения доступа к данным используется принцип кэширования. В вычислительных системах существует иерархия запоминающих устройств:
  - нижний уровень занимает емкая, но относительно медленная дисковая память;
  - оперативная память;
  - верхний уровень – сверхоперативная память процессорного кэша.
- Каждый уровень играет роль кэша по отношению к нижележащему.

Кэширование данных

- Каждая запись в кэш-памяти об элементе данных включает в себя:
  - Значение элемента данных;
  - Адрес, который этот элемент данных имеет в основной памяти;
  - Дополнительную информацию, которая используется для реализации алгоритма замещения данных в кэше и включает признак модификации и актуальности данных.