

Распределение памяти



Методы распределения памяти с использованием дискового пространства

- ▶ • **Страничный** - перемещение данных между памятью и диском происходит частями фиксированного размера.
- ▶ • **Сегментная** - перемещение данных между памятью и диском происходит частями виртуального адресного пространства произвольного размера, полученными с учетом смыслового значения данных.
- ▶ • **Сегментно-страничная** - способ управления памятью объединяет в себе элементы обоих предыдущих подходов.



Страничное распределение

- ▶ Виртуальное адресное пространство и вся оперативная память делится на части одинакового, фиксированного для данной системы размера, которые называются **виртуальными** и **физическими страницами** соответственно.
- ▶ При загрузке каждого процесса операционная система создаёт **таблицу страниц** – специальную информационную структуру, элемент которой называется **дескриптор страницы**.

•



Страничное распределение

С помощью дескрипторов устанавливается соответствие между номерами виртуальных и физических страниц, если они загружены в оперативную память, или делается отметка о том, что виртуальная страница выгружена на диск.



Страничное распределение

В таблице страниц содержится управляющая информация, такая как:

- **признак модификации страницы**, признак невыгружаемости (выгрузка некоторых страниц может быть запрещена),
- **признак обращения к странице** (используется для подсчета числа обращений за определенный период времени)
- и другие данные, формируемые и используемые механизмом виртуальной памяти.



Страничное распределение

► При каждом обращении к памяти происходит чтение из таблицы страниц информации о виртуальной странице, к которой произошло обращение.



- Если данная виртуальная страница находится в оперативной памяти, то немедленно выполняется преобразование виртуального адреса в физический.



Страничное распределение

- ▶ Если нужная виртуальная страница в данный момент выгружена на диск, то происходит так называемое **страничное прерывание**.
- ▶ При этом выполняющийся процесс переводится в состояние ожидания, и активизируется другой процесс из очереди готовых.
- ▶ Параллельно программа обработки страничного прерывания находит на диске требуемую виртуальную страницу и пытается загрузить ее в оперативную память.



Страничное распределение

Если в памяти имеется свободная физическая страница, то загрузка выполняется немедленно, если же свободных страниц нет, то решается вопрос, какую страницу следует выгрузить из оперативной памяти.

Критерии выбора:

- 1) дольше всего не использовавшаяся страница;
- 2) первая попавшаяся страница;
- 3) страница, к которой в последнее время было меньше всего обращений.



Страничное распределение

- ▶ После того, как выбрана страница, которая должна покинуть оперативную память, анализируется ее признак модификации (из таблицы страниц).
- ▶ Если выталкиваемая страница с момента загрузки была модифицирована, то ее новая версия переписывается на диск иначе физическая страница объявляется свободной.



Страничное распределение

▶ На производительность системы со страничной организацией памяти влияют временные затраты, связанные с обработкой страничных прерываний и преобразованием виртуального адреса в физический.



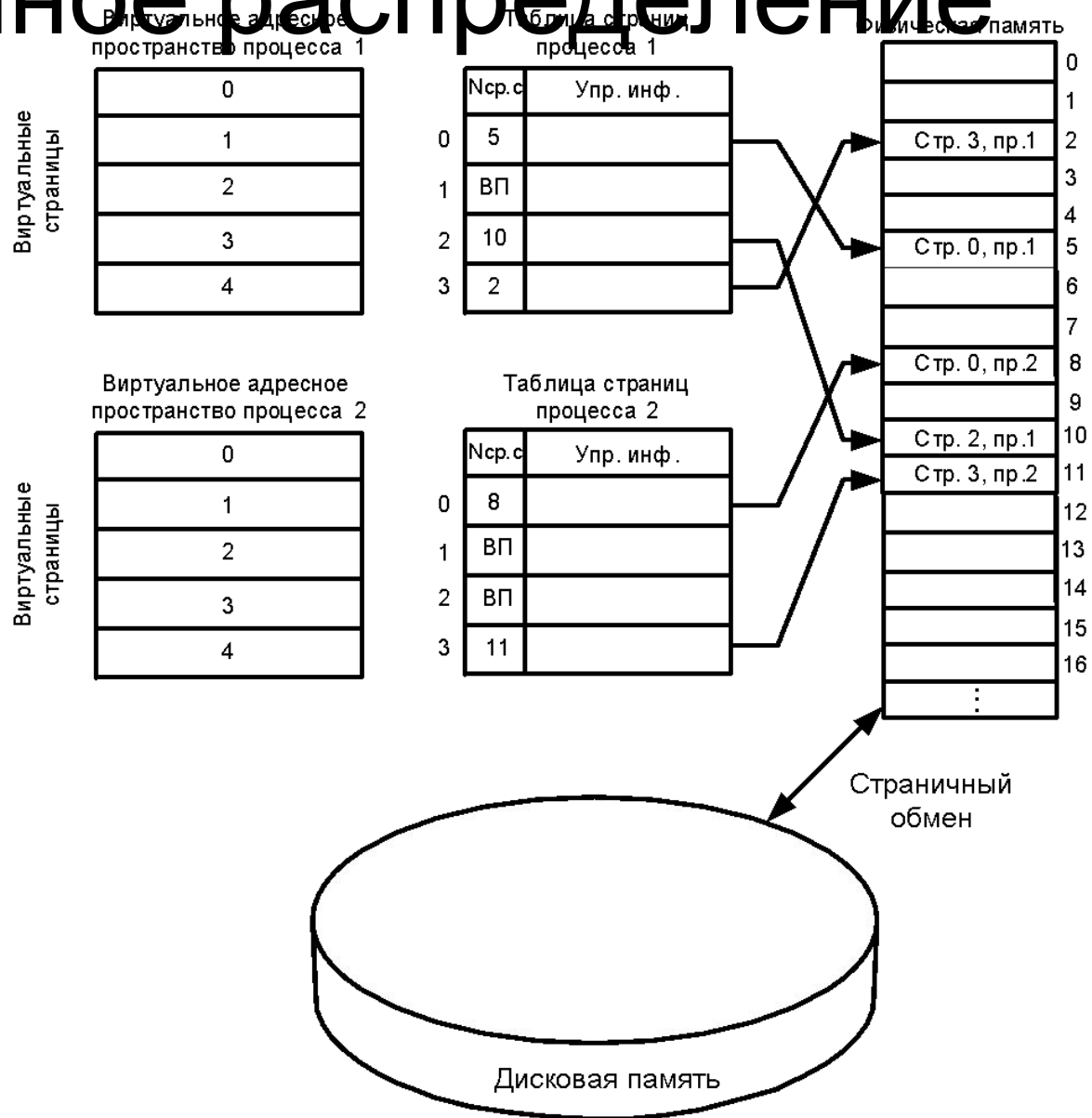
способы повышения

производительности

- ▶ 1) **увеличение размера страницы** – приводит к уменьшению частоты страничных прерываний и размера таблицы страниц, т.е. уменьшается затраты времени и оперативной памяти. Однако, при увеличении размера страницы увеличивается и *фиктивная область* в последней виртуальной странице каждой программы.
- ▶ 2) **размещение таблицы страниц в "быстрых" запоминающих устройствах**, т.е. уменьшается время доступа к таблице страниц.



Страничное распределение





Двоичное представление адресов

00 0 000 000 000	начало 0-й страницы
00 0 000 000 001	
00 0 000 000 010	
00 1 111 111 111	конец 0-й страницы
01 0 000 000 000	начало 1-й страницы
01 0 000 000 001	
01 1 111 111 111	конец 1-й страницы
10 0 000 000 000	начало 2-й страницы
10 0 000 000 001	
10 1 000 111 001	исходный виртуальный
10 1 111 111 111	конец 2-й страницы
11 0 000 000 000	начало 3-й страницы



Схема преобразования виртуального адреса в физический

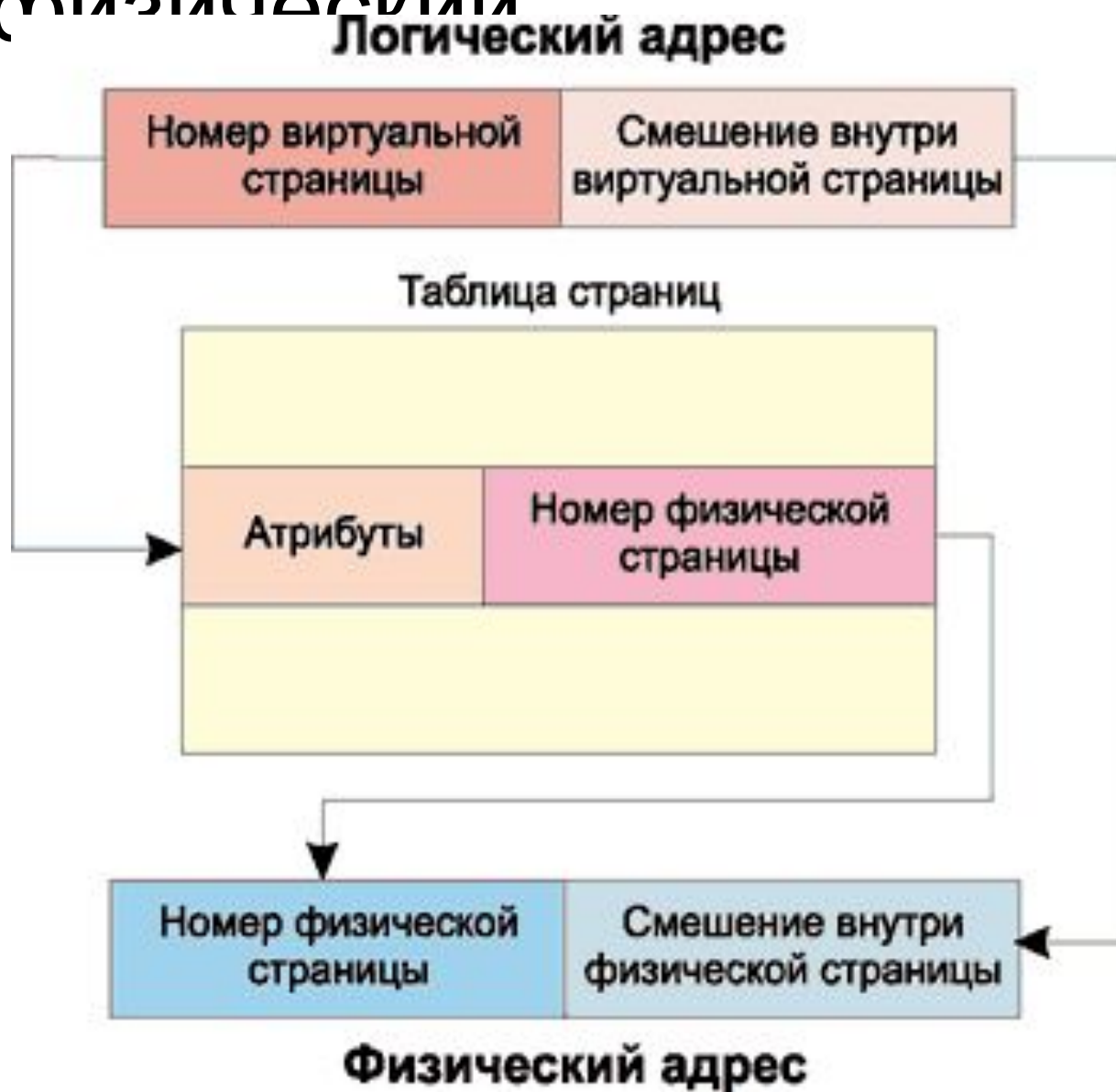
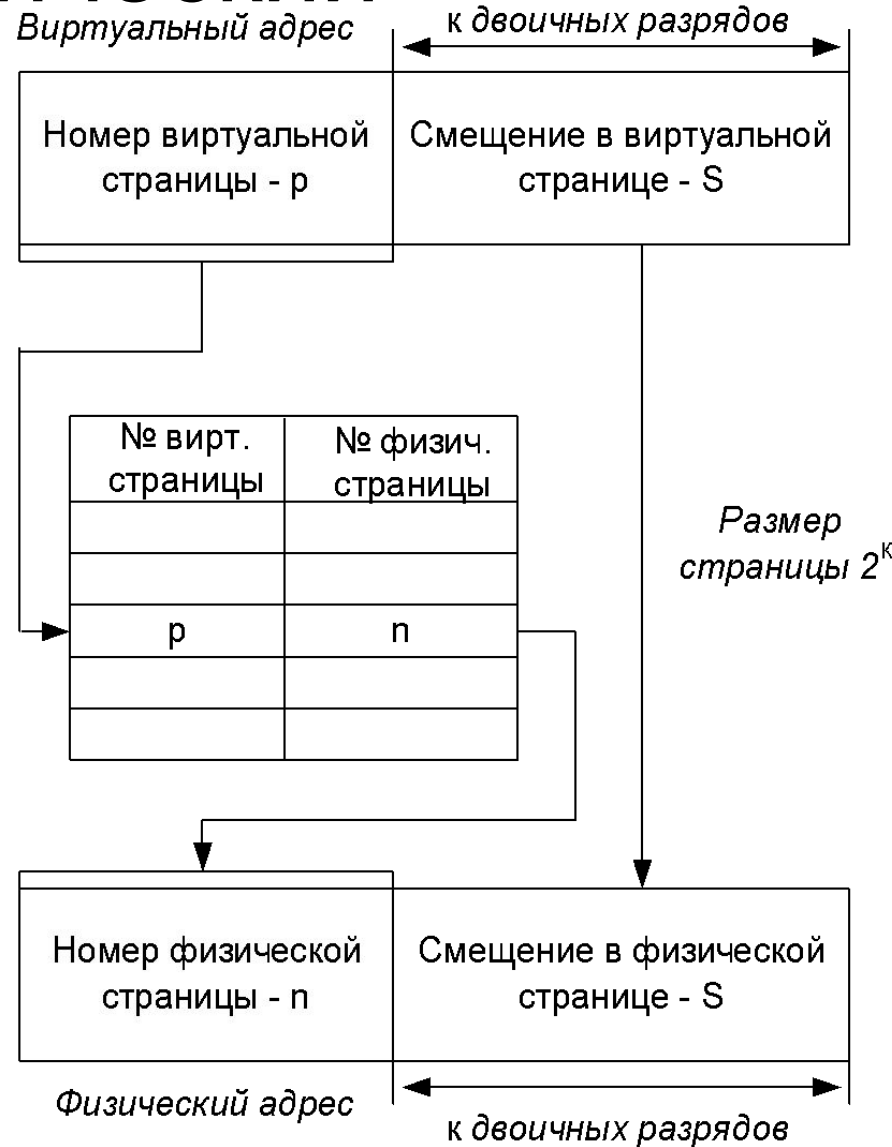




Схема преобразования виртуального адреса в физический





Сегментное распределение

Виртуальное адресное пространство процесса делится на **сегменты**, размер которых определяется программистом с учетом смыслового значения содержащейся в них информации.

- *Отдельный сегмент может представлять собой подпрограмму, массив данных и т.п.*
- При загрузке процесса часть сегментов помещается в оперативную память (при этом для каждого из этих сегментов операционная система подыскивает подходящий участок свободной памяти), а часть сегментов размещается в дисковой памяти.



Сегментное распределение

- ▶ Сегменты одной программы могут занимать в оперативной памяти несмежные участки.
- ▶ Во время загрузки система создает таблицу сегментов процесса (аналогичную таблице страниц), в которой для каждого сегмента указывается начальный физический адрес сегмента в оперативной памяти, размер сегмента, правила доступа, признак модификации, признак обращения к данному сегменту за последний интервал времени и другая информация.



Способности функционирования системы с сегментной организацией

- ▶ 1) при отсутствии нужных сегментов в памяти происходит прерывание для обработки возникшей исключительной ситуации, т.е. необходимые сегменты загружаются из внешней памяти;
- ▶ 2) при необходимости освобождения памяти по определенному алгоритму определяются сегменты, которые выгружаются на диск;
- ▶ 3) при каждом обращении к оперативной памяти выполняется преобразование виртуального адреса в физический.
- ▶ 4) при обращении к памяти проверяется, разрешен ли доступ требуемого типа к данному сегменту.



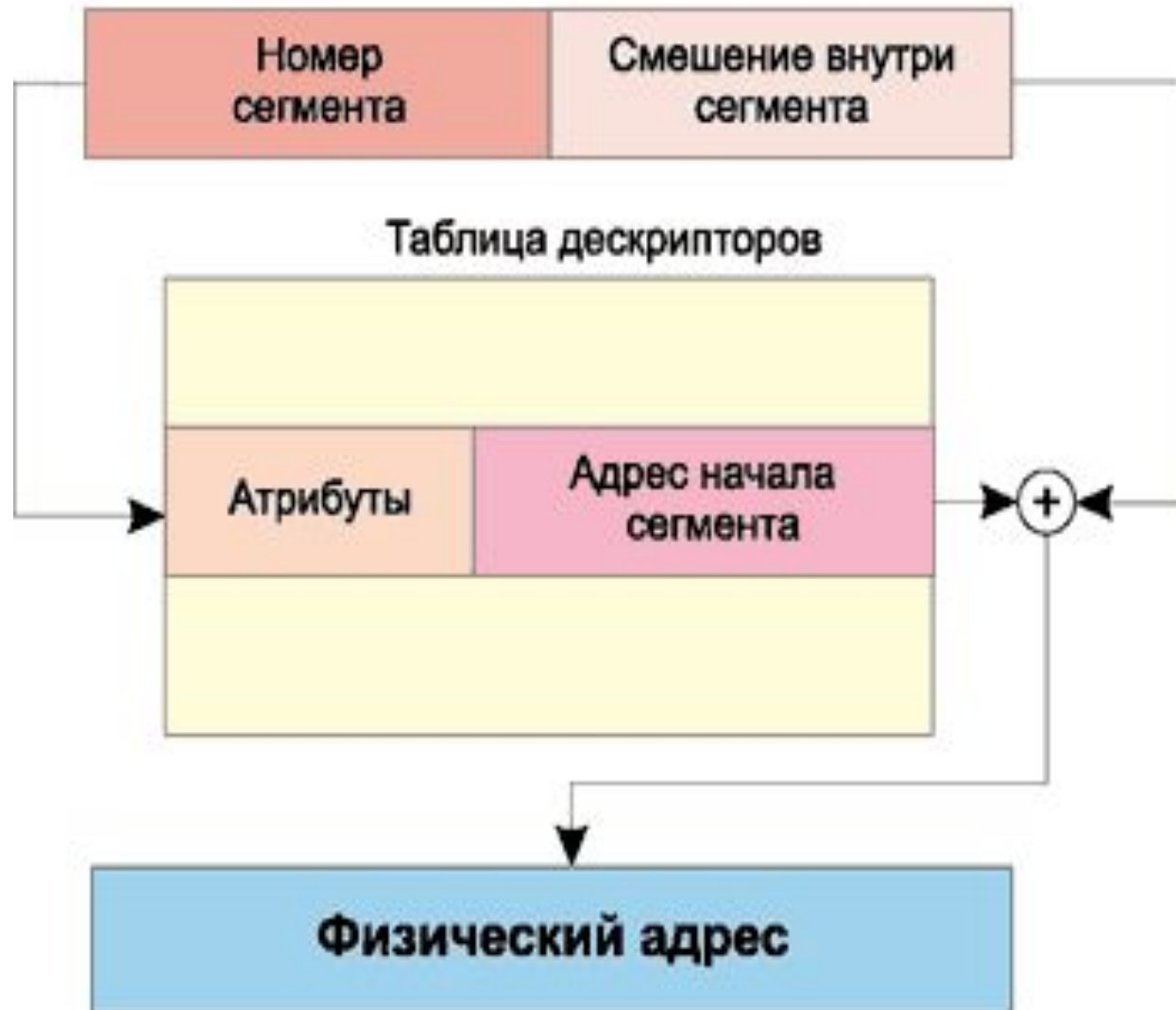
Способности функционирования системы с сегментной организацией

- ▶ Недостатком данного метода распределения памяти является фрагментация на уровне сегментов и более медленное по сравнению со страничной организацией преобразование адреса.



Сегментное распределение

Логический адрес





Сегментное распределение

Виртуальное адресное пространство процесса A

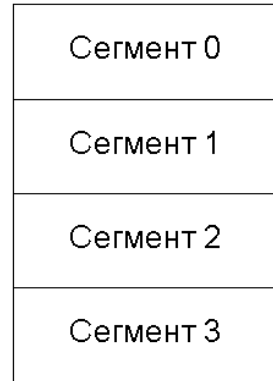


Таблица сегментов процесса A

0	
1	
2	Выгружен
3	Выгружен

Виртуальное адресное пространство процесса B

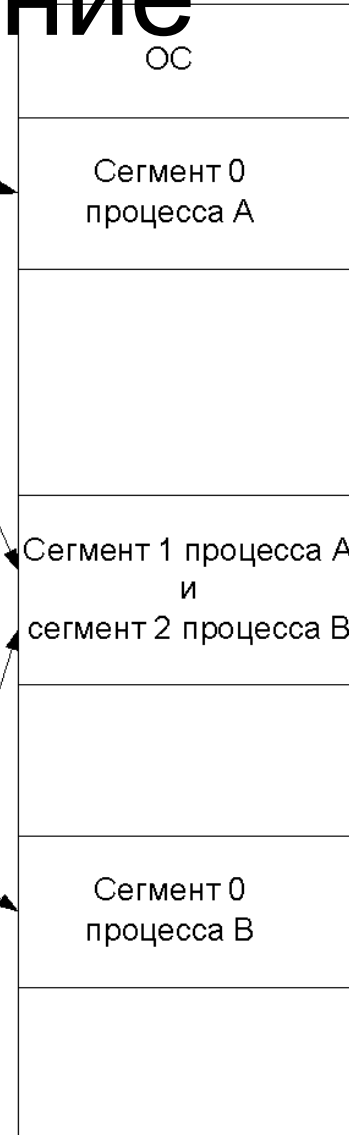


Таблица сегментов процесса B

0	
1	Выгружен
2	

Разделенный сегмент

ОС





Сегментно-страничное распределение

- ▶ • Виртуальное пространство процесса делится на сегменты, а каждый сегмент в свою очередь делится на виртуальные страницы, которые нумеруются в пределах сегмента.
- ▶ • Оперативная память делится на физические страницы.
- ▶ • Загрузка процесса выполняется операционной системой постранично, при этом часть страниц размещается в оперативной памяти, а часть на диске.



Сегментно-страничное распределение

- ▶ Для каждого сегмента создается своя таблица страниц, структура которой полностью совпадает со структурой таблицы страниц, используемой при страничном распределении.
- ▶ Для каждого процесса создается таблица сегментов, в которой указываются адреса таблиц страниц для всех сегментов данного процесса.
- ▶ Адрес таблицы сегментов загружается в специальный регистр процессора, когда активизируется соответствующий процесс.

Схема преобразования виртуального адреса в физический для сегм

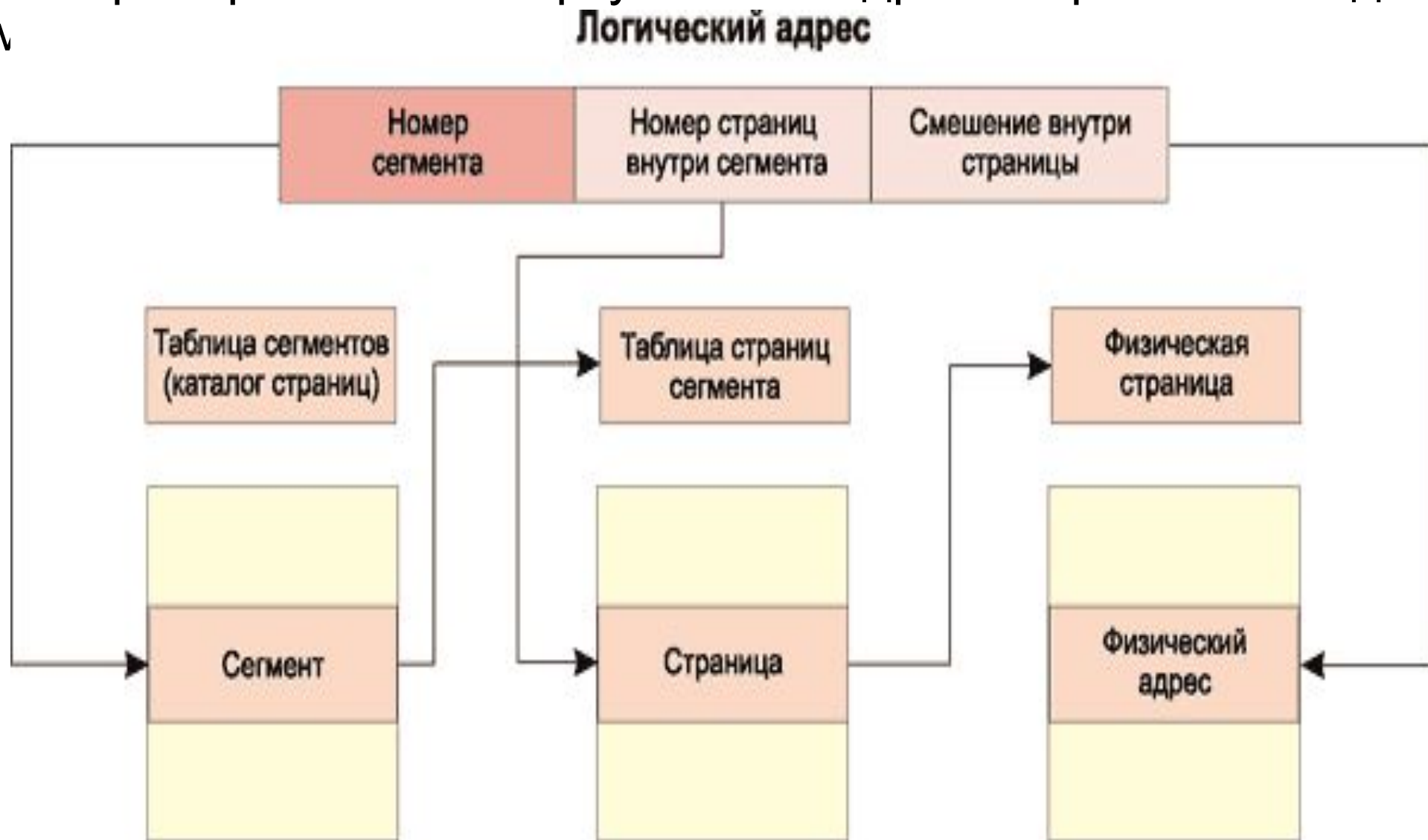
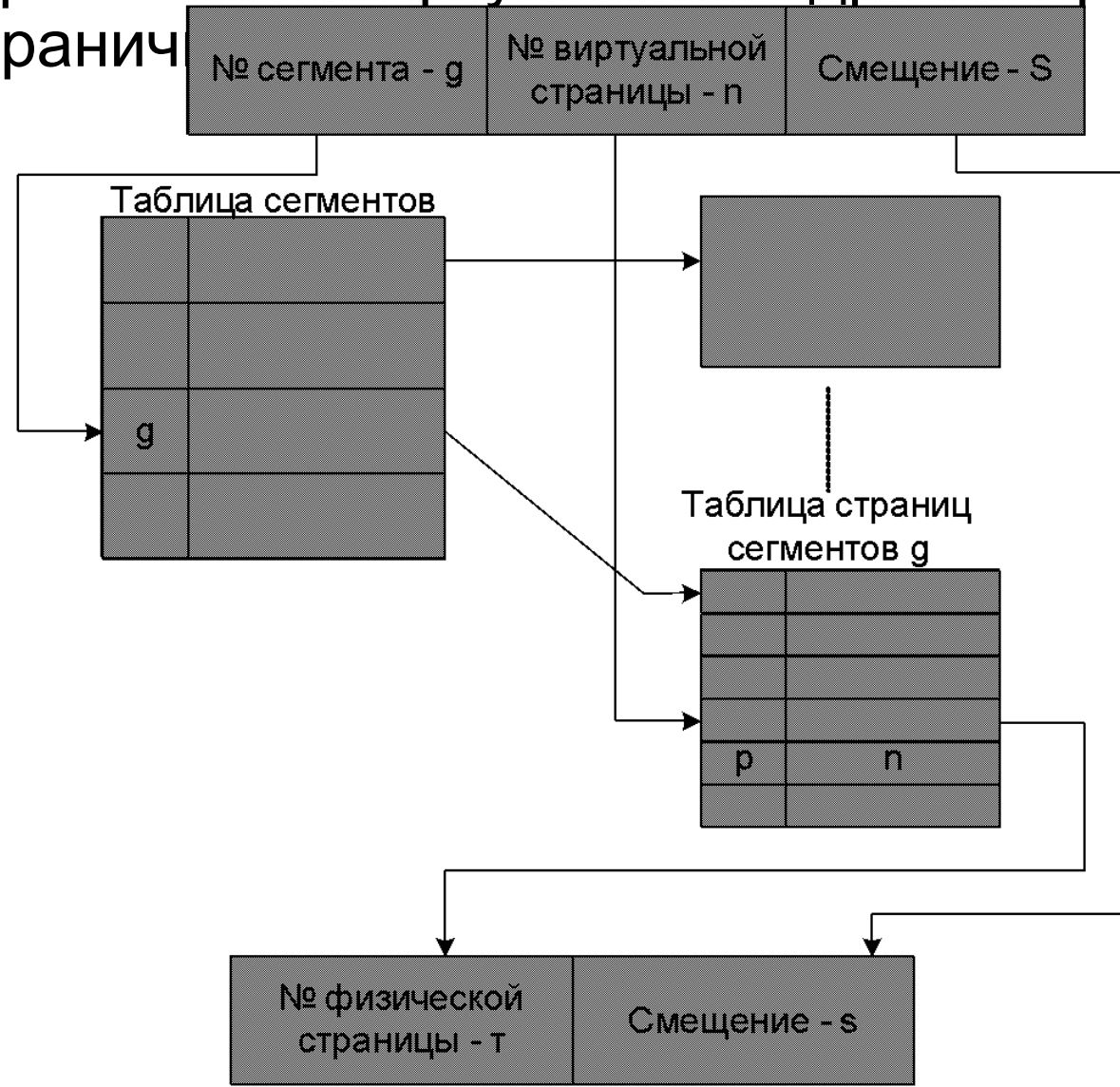


Схема преобразования виртуального адреса в физический для сегментно-страничного



Физический адрес



Сегментно-страничное распределение

- ▶ Данный метод представляет собой комбинацию страничного и сегментного механизмов управления памятью.
- ▶ Так же как и при сегментной организации памяти, виртуальное адресное пространство процесса разделено на сегменты (это позволяет определять разные права доступа к разным частям кодов и данных программы).
- ▶ Перемещение данных между памятью и диском осуществляется не сегментами, а страницами. Для этого каждый виртуальный сегмент и физическая память делятся на страницы равного размера, что позволяет более эффективно использовать память, сократив до минимума фрагментацию



Сегментно-страничное распределение

► Достоинства:

- эффективное использование памяти
- минимальная фрагментация