

# Управление памятью

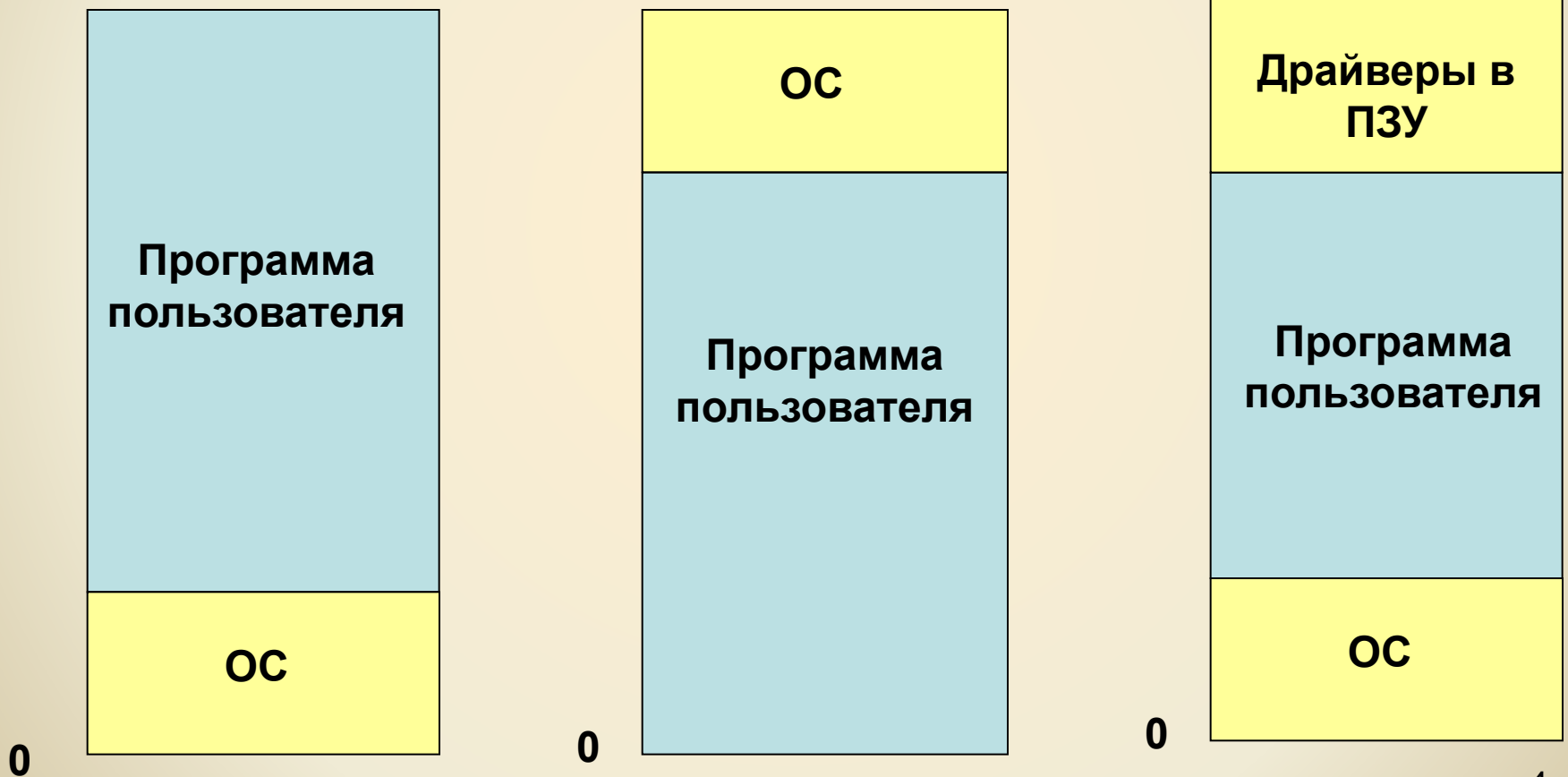
- Требования к управлению памятью
- Распределение памяти
  - Фиксированное распределение
  - Распределение разделами переменной длины
  - Страничная организация
  - Свопинг
- Виртуальная память
  - Понятие и организация
  - Стратегии размещения и замещения
- Управление памятью в UNIX и Linux
- Управление памятью в Windows

# Требования к управлению памятью

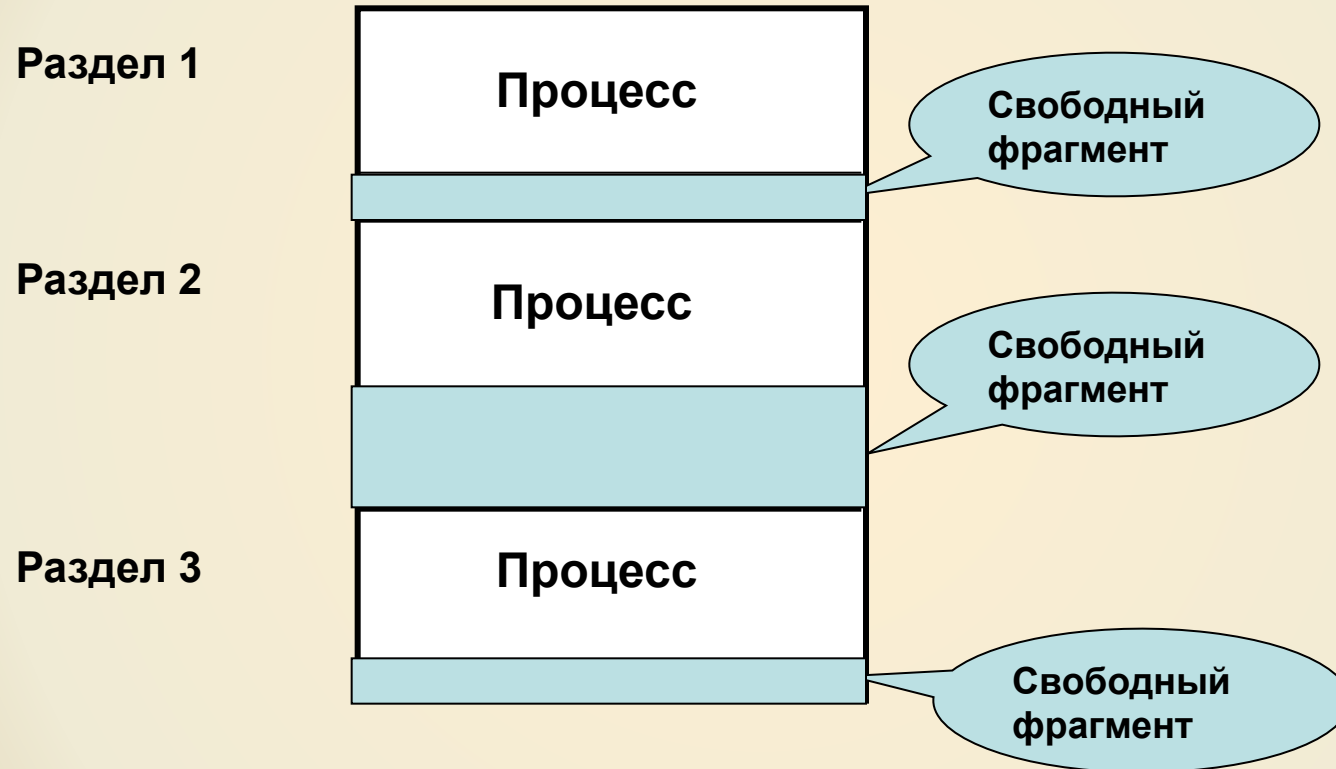
- Перемещение
- Защита
- Совместное использование
- Логическая организация
- Физическая организация

# Распределение памяти

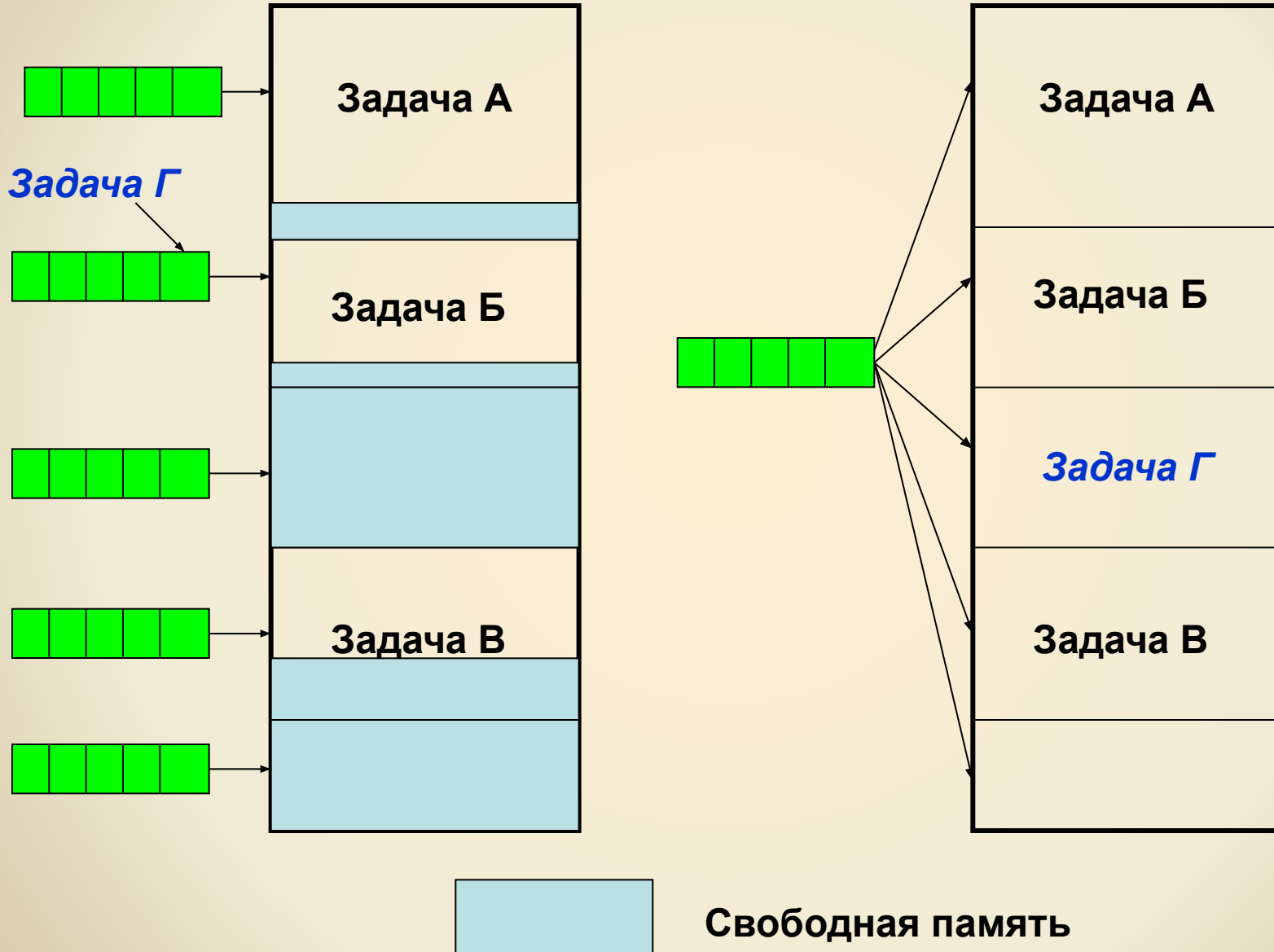
Однозадачная система



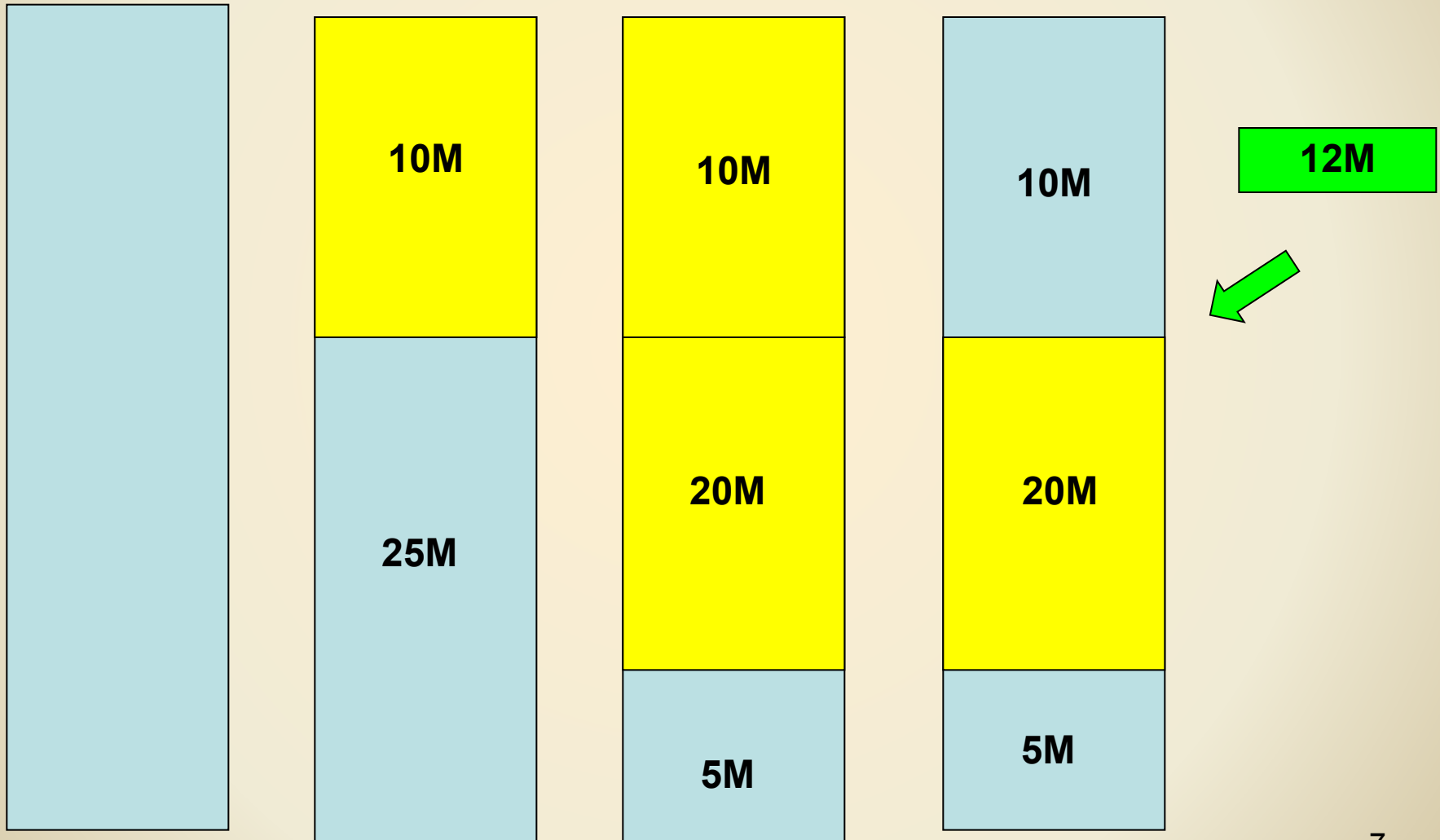
# Распределение статическими разделами одинаковой длины



# Распределение статическими разделами переменной длины



# Распределение динамическими разделами

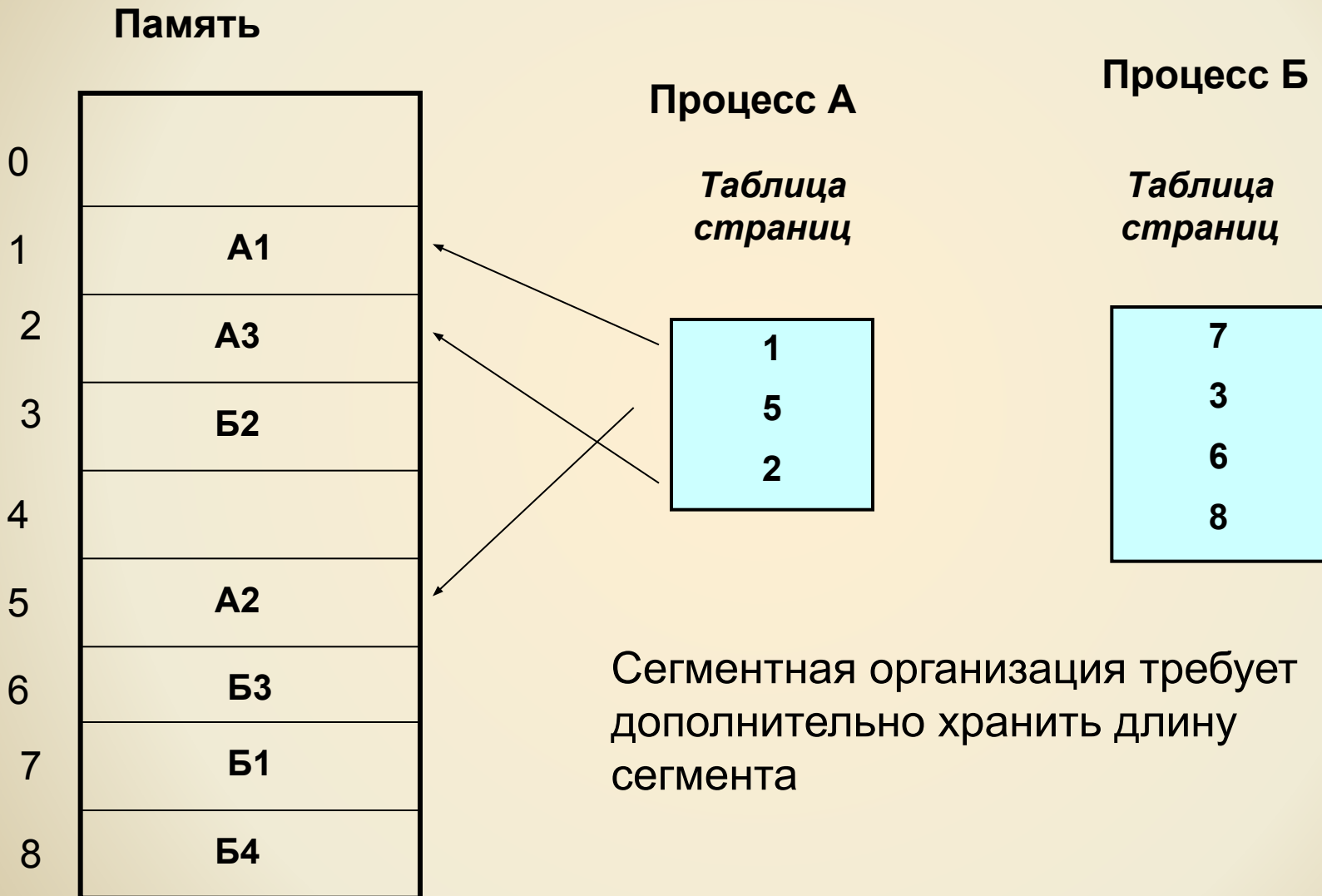


# Алгоритмы размещения

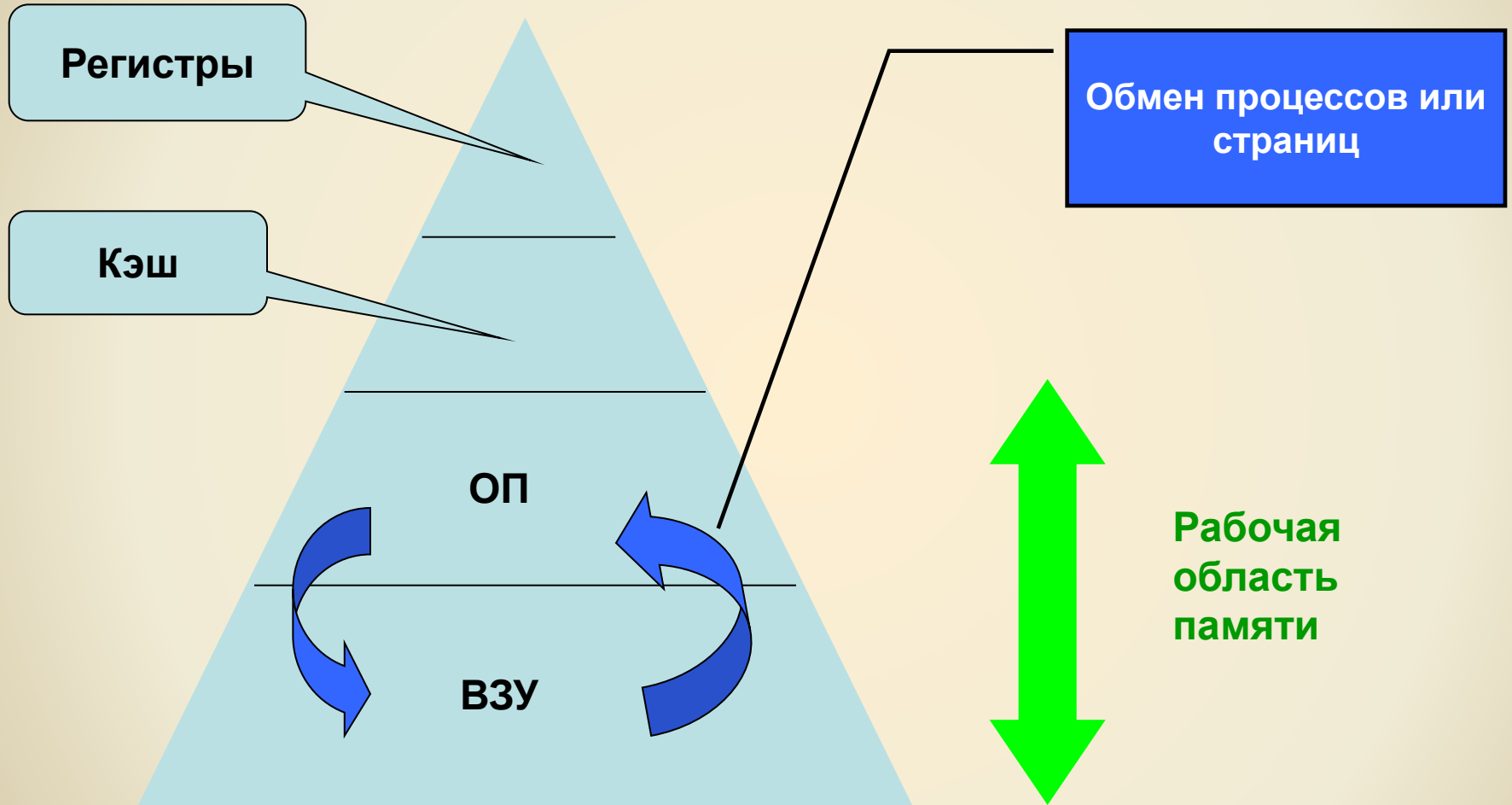
- **Наилучший подходящий**
  - Список свободных участков упорядочен по возрастанию объема
  - Оставшийся свободный фрагмент попадет в начало списка
  - Этот фрагмент будет настолько мал, что в нем не сможет разместиться какой-либо раздел
- **Первый подходящий**
  - Список свободных участков упорядочивается по адресам
  - В среднем необходимо просматривать половину списка
  - Память для небольших задач будет выделяться в начале списка
- **Наименее подходящий**
  - Список свободных участков упорядочивается по убыванию объема
  - Остаток после размещения задачи достаточно большой для размещения любой задачи



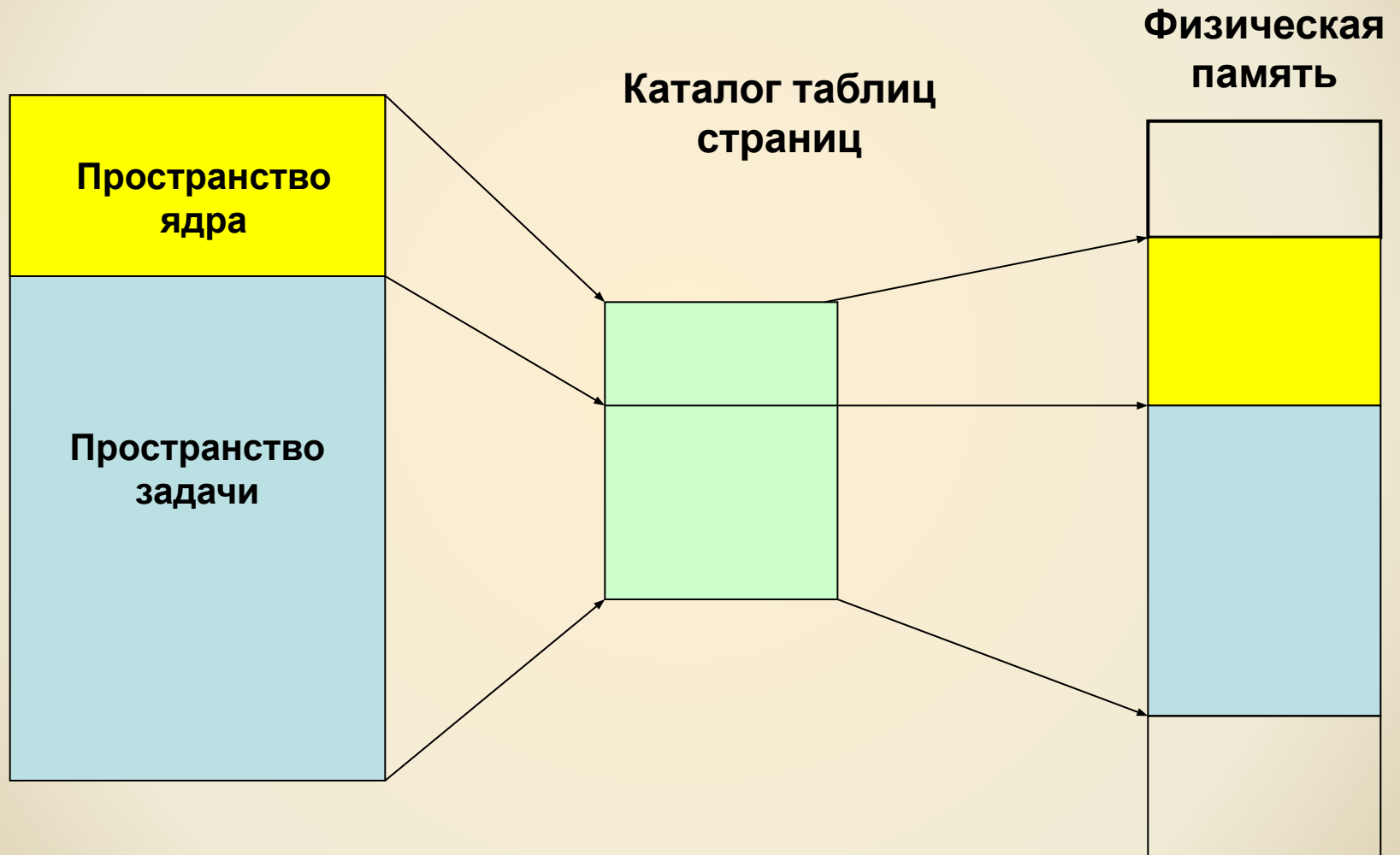
# Страничная организация

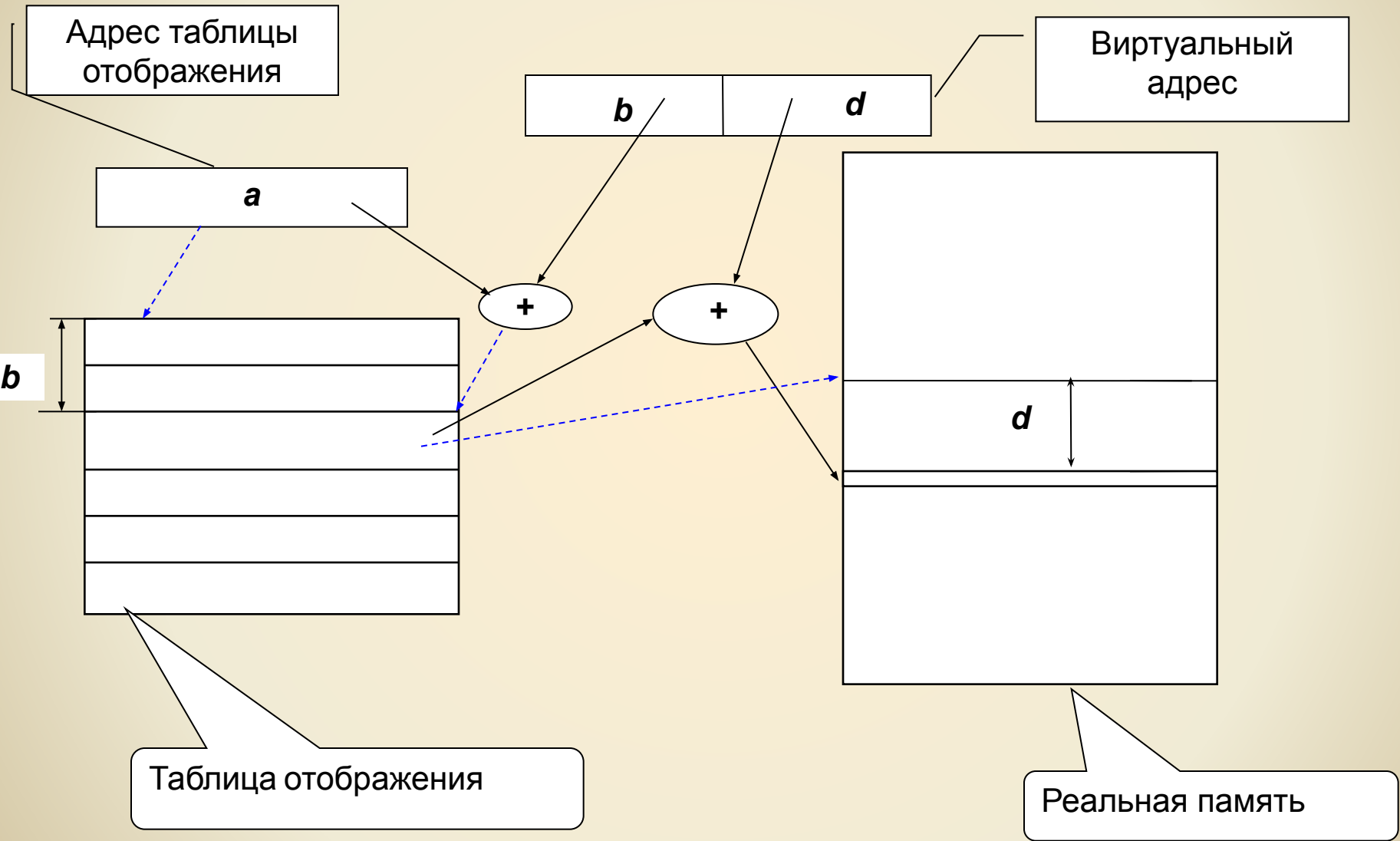


# СВОПИНГ



# Виртуальная память





# Страничная организация

Виртуальный адрес

Номер страницы	Смещение
----------------	----------

Запись таблицы страниц

<b>P</b>	<b>M</b>	Управляющие биты	Номер кадра
----------	----------	------------------	-------------

# Сегментная организация

Виртуальный адрес

Номер сегмента	Смещение
----------------	----------

Запись таблицы страниц

Р	М	Управляющие биты	Длина	Начальный адрес сегмента
---	---	------------------	-------	--------------------------

# Стратегии размещения и замещения

- Стратегии выборки – определяет какие блоки загружать из ВЗУ
  - По требованию
  - Предварительно
- Стратегии размещения - где размещать подгружаемые блоки (сегментная организация)
  - Первый подходящий
  - Очередной подходящий
  - Наименее подходящий
- Стратегии замещения – какие блоки выгружать из оперативной памяти на диск
  - Самая старая страница
  - Не использовавшаяся в последнее время страница
  - Страница, не использовавшаяся дольше всего

## Управление памятью в UNIX

- SVR4 использует две отдельные схемы управления памятью.
- Страничная система распределяет кадры основной памяти среди процессов.
- Для управления памятью ядра используется другой механизм.



# Страничная система

- **Таблица страниц**

- Для каждого процесса используется одна таблица страниц, в которой каждой странице виртуальной памяти процесса соответствует одна запись. Эта запись имеет следующую структуру.
  - **Номер кадра.** Указывает кадр в физической памяти.
  - **Возраст.** Указывает, как долго страница находится в памяти без обращения к ней.
  - **Копирование при записи.** Устанавливается, когда страница разделяется несколькими процессами. Если один из процессов производит запись в страницу, сначала должны быть сделаны отдельные копии страницы для каждого из совместно использующих ее процессов.
  - **Модифицирована.** Указывает, изменено ли содержимое страницы.
  - **Обращения.** Указывает, что к странице было обращение.
  - **В памяти.** Указывает, что страница находится в оперативной памяти.
  - **Защита.** Указывает, что разрешена запись.

- **Дескриптор дискового блока**

- В этой таблице каждой странице процесса соответствует запись, описывающая дисковую копию этой страницы.
  - **Номер устройства свопинга.** Номер логического устройства вторичной памяти, хранящего соответствующую страницу.
  - **Номер блока устройства.** Расположение блока страницы на устройстве вторичной памяти.
  - **Тип памяти.** Вторичная память может представлять модуль свопинга или исполнимый файл. В последнем случае имеется признак, указывающий, должна ли распределяемая виртуальная память быть предварительной очищенной.

## • **Запись таблицы кадров**

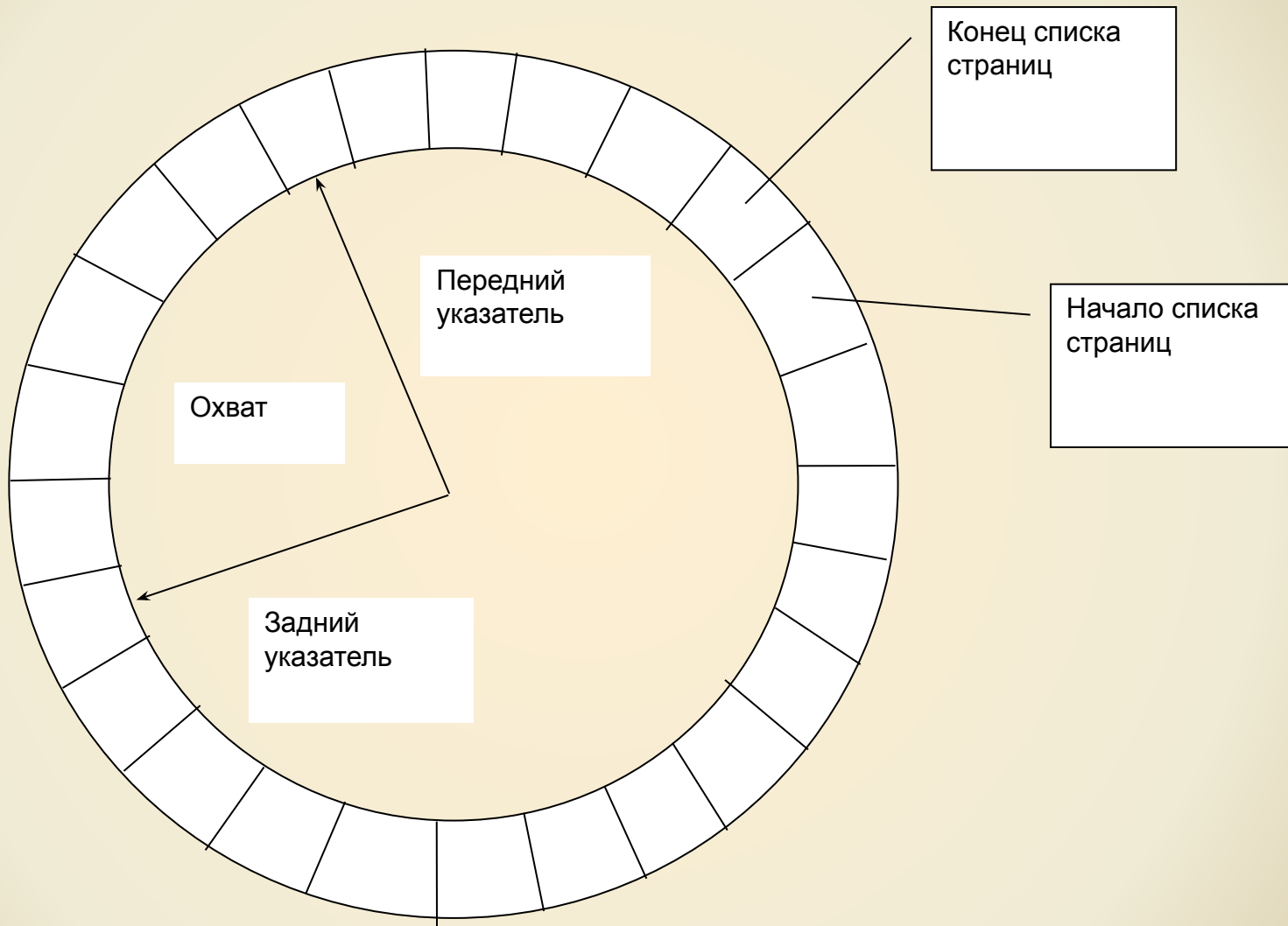
- Описывает каждый кадр реальной памяти, таблица проиндексирована номерами кадров.
  - **Состояние страницы.** Указывает свободен ли кадр или содержит страницу. В этом случае указывает статус страницы: на устройстве свопинга, в выполняемом файле или выполняется прямое обращение к памяти.
  - **Количество ссылок.** Количество процессов, использующих страницу.
  - **Логическое устройство.** Логическое устройство, содержащее копию страницы.
  - **Номер блока.** Расположение блока копии страницы на логическом устройстве.
  - **Указатель на данные кадра.** Указатель на другие записи таблицы в списке свободных страниц или в хэш-таблице.

- **Таблица использования свопинга**
- Для каждого устройства свопинга имеется своя таблица, в которой для каждой страницы на этом устройстве имеется своя запись.
  - **Количество ссылок.** Количество записей таблицы страниц, указывающих на страницы на устройстве свопинга.
  - **Идентификатор страницы.** Идентификатор страницы в модуле страничной памяти.

- Откачку страниц, не входящих в рабочие наборы процессов, производит специальный системный процесс - stealer. Он начинает работать, когда количество страниц в списке свободных страниц достигает установленного нижнего порога.
- Алгоритм замещения в SVR4 представляет собой усовершенствованный часовой алгоритм (алгоритм с двумя стрелками). Алгоритм использует бит обращений для каждой из страниц памяти.
  - Работа алгоритма характеризуется двумя параметрами.

**Частота сканирования.** Частота, с которой указатели сканируют список страниц (кадров в секунду).

**Охват.** Интервал между передним и задним указателями.



# Распределение памяти ядра

- В системе двойников память распределяется блоками длиной  $2^K$ .

$$L \leq K \leq U,$$

где  $2^L$  – min размер выделяемого блока памяти,

$2^U$  – max распределяемый блок ( $2^U$  – размер всей доступной для распределения памяти).

- При запросе памяти размером  $S$ , таким, что

$$2^{U-1} < S \leq 2^U$$

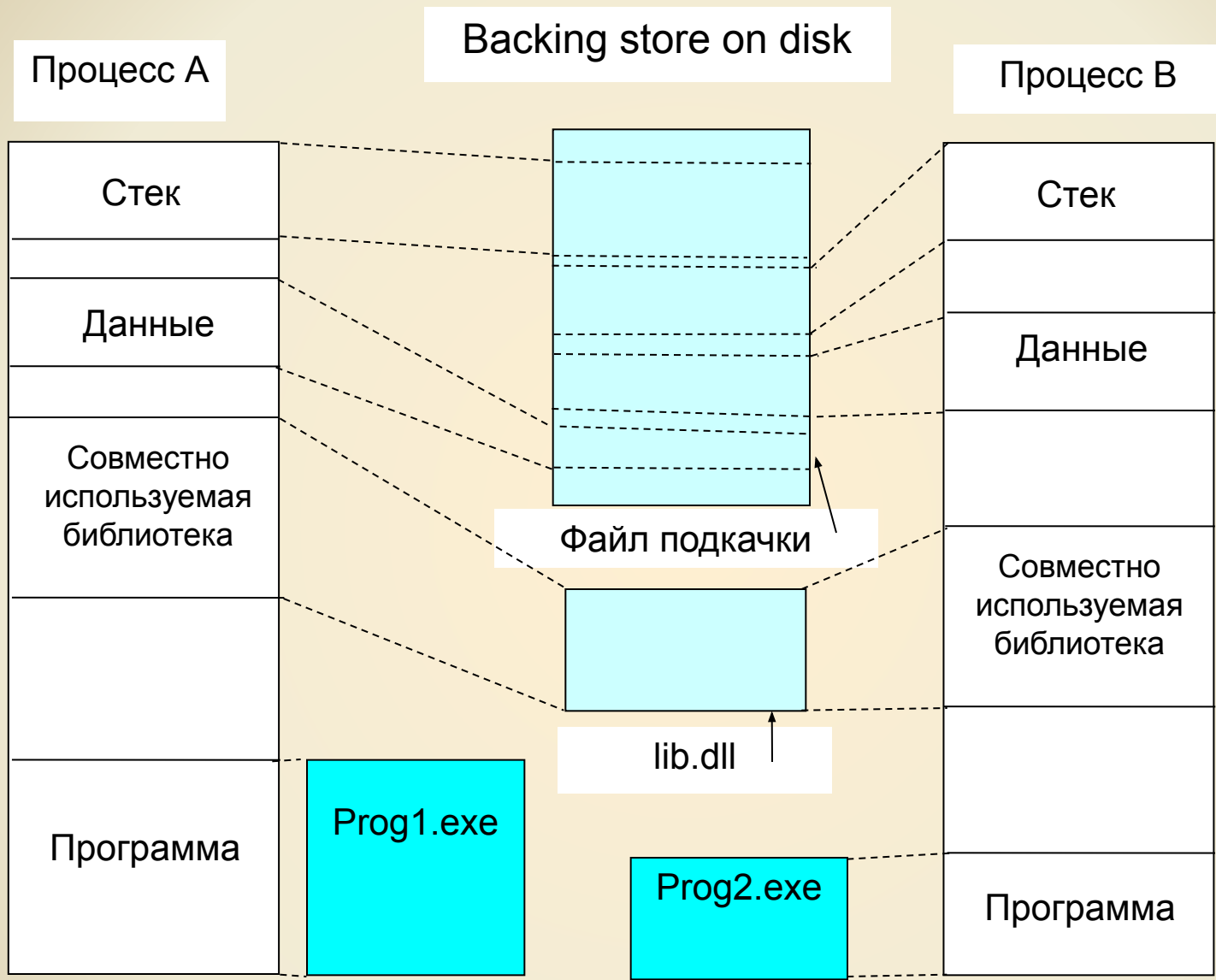
выделяется весь блок

- В противном случае, блок разделяется на два эквивалентных двойника с размерами  $2^{U-1}$
- Если  $2^{U-2} < S \leq 2^{U-1}$  то по запросу выделяется один из двойников, в противном случае один из двойников снова делится пополам. Этот процесс продолжается до тех пор, пока не будет сгенерирован наименьший блок, размер которого не меньше  $S$ .

# Управление памятью в Windows

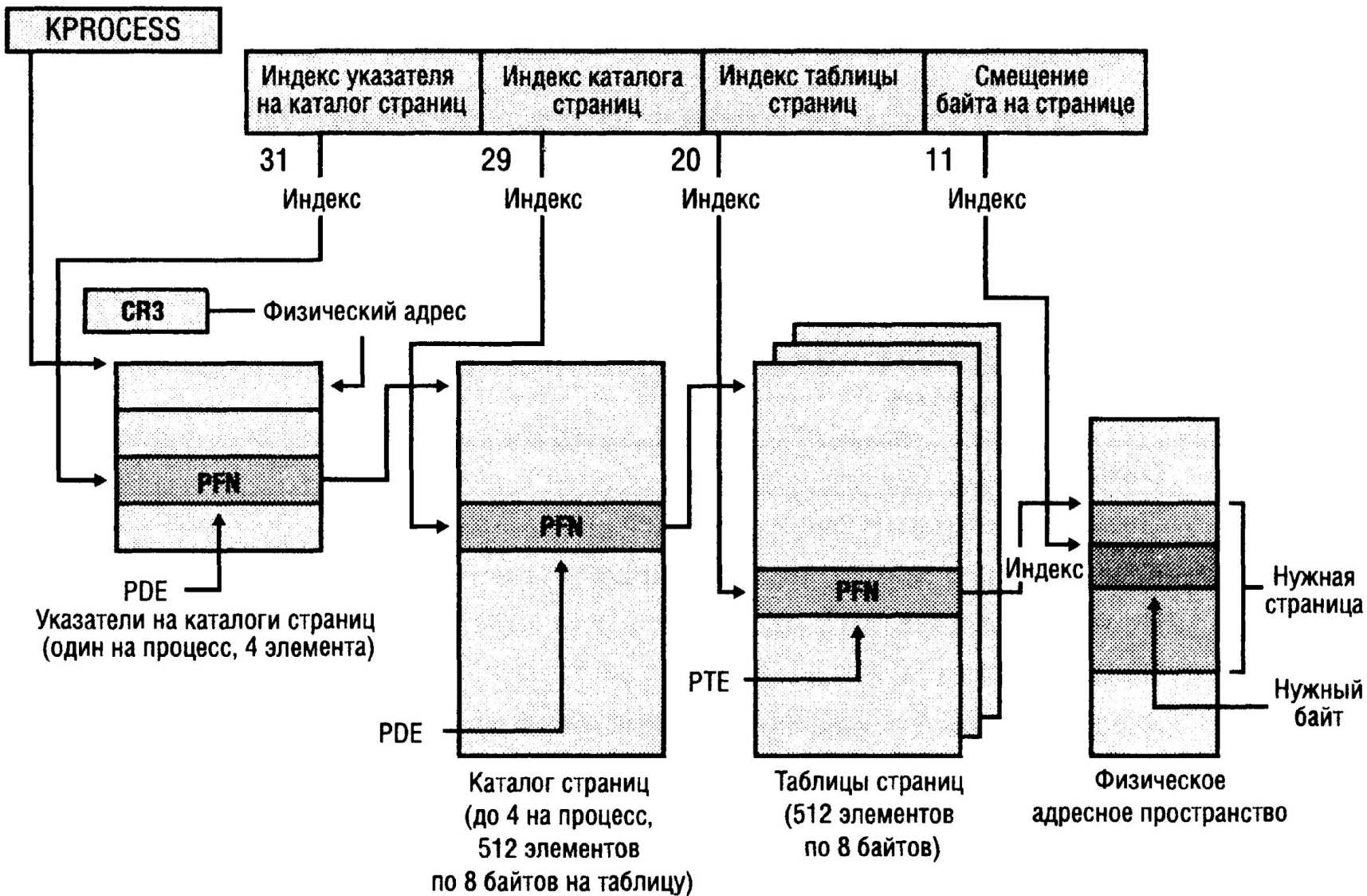






# СИСТЕМНЫЕ ВЫЗОВЫ

<b>VirtualAlloc</b>	<b>Зарезервировать или фиксировать область</b>
<b>VirtualFree</b>	<b>Освободить область или отменить фиксацию области</b>
<b>VirtualProtect</b>	<b>Изменить режим доступа (чтение/запись/выполнение) к области</b>
<b>VirtualQuery</b>	<b>Узнать состояние области</b>
<b>VirtualLock</b>	<b>Сделать область резидентной в памяти (то есть запретить ее выгрузку)</b>
<b>VirtualUnlock</b>	<b>Разрешить выгрузку области</b>
<b>CreateFileMapping</b>	<b>Создать объект отображаемого файла и (по желанию) присвоить ему имя</b>
<b>MapViewOfFile</b>	<b>Отобразить файл (часть файла) на адресное пространство</b>
<b>UnmapViewOfFile</b>	<b>Удалить отображаемый файл из адресного пространства</b>
<b>OpenFileMapping</b>	<b>Открыть созданный ранее объект отображаемого файла</b>



# Запись таблицы страниц (PTE)

Биты	20	3	1	1	1	1	1	1	1	1	1
Страничный блок	Не используется	G	L	D	A	C	Wt	W	U	V	

<b>G:</b> Глобальная страница для всех процессов	<b>Wt:</b> Сквозная запись (без кэширования)
<b>L:</b> Большая (4 Мбайт) страница	<b>U:</b> К странице возможен доступ в режиме пользователя
<b>D:</b> Страница «грязная»	<b>W:</b> Запись в страницу разрешена
<b>A:</b> К странице были обращения	<b>V:</b> Действительная запись таблицы
<b>C:</b> Кэширование разрешено/запрещено	

# Дескриптор сегмента кода

