

## 6.1 Управление памятью. Основные понятия

**Менеджер памяти** - часть операционной системы, отвечающая за управление памятью.

Основные методы распределения памяти:

- Без использования внешней памяти (например: HDD)
- С использованием внешней памяти

## 6.2 Методы без использования внешней памяти

### 6.2.1 Однозадачная система без подкачки на диск

Память разделяется только между программой и операционной системой.

Схемы разделения памяти:



Схемы разделения памяти

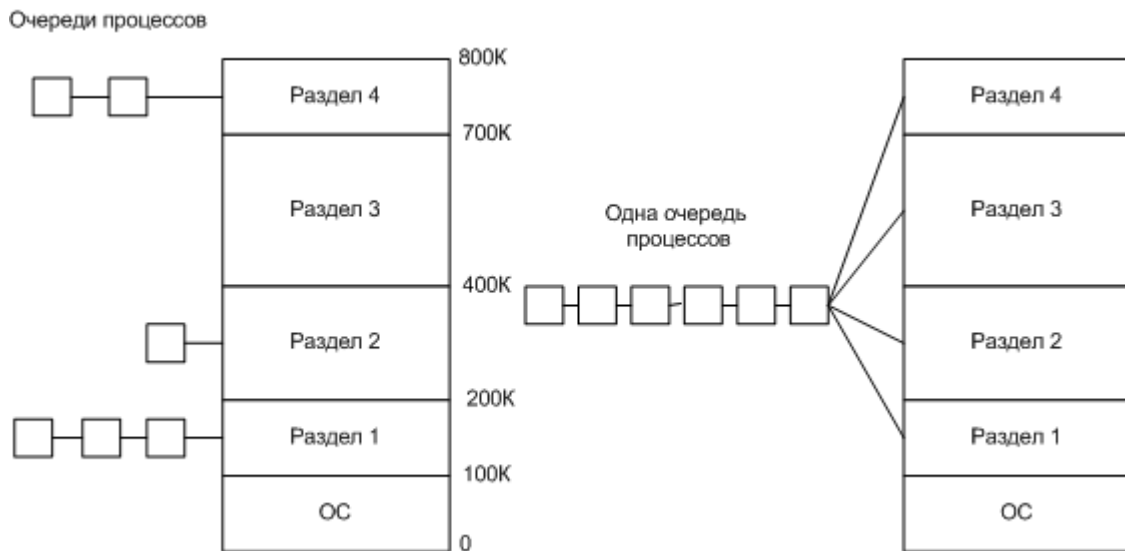
Третий вариант используется в MS-DOS. Та часть, которая находится в ПЗУ, часто называется BIOS.

### 6.2.2 Распределение памяти с фиксированными разделами.

Память просто разделяется на несколько разделов (возможно, не равных). Процессы могут быть разными, поэтому каждому разделу необходим разный размер памяти.

Системы могут иметь:

- общую очередь ко всем разделам
- к каждому разделу отдельную очередь



### Распределение памяти с фиксированными разделами

Недостаток системы многих очередей очевиден, когда большой раздел может быть свободным, а к маленькому выстроилась очередь.

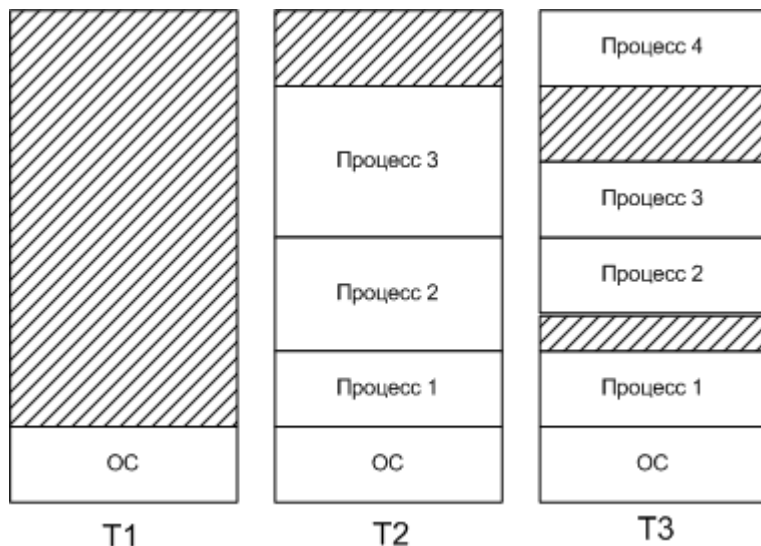
Алгоритмы планирования в случае одной очереди:

- поочередный
- выбирается задача, которая максимально займет раздел

Также может быть смешанная система.

### 6.2.3 Распределение памяти динамическими разделами

В такой системе сначала память свободна, потом идет динамическое распределение памяти.



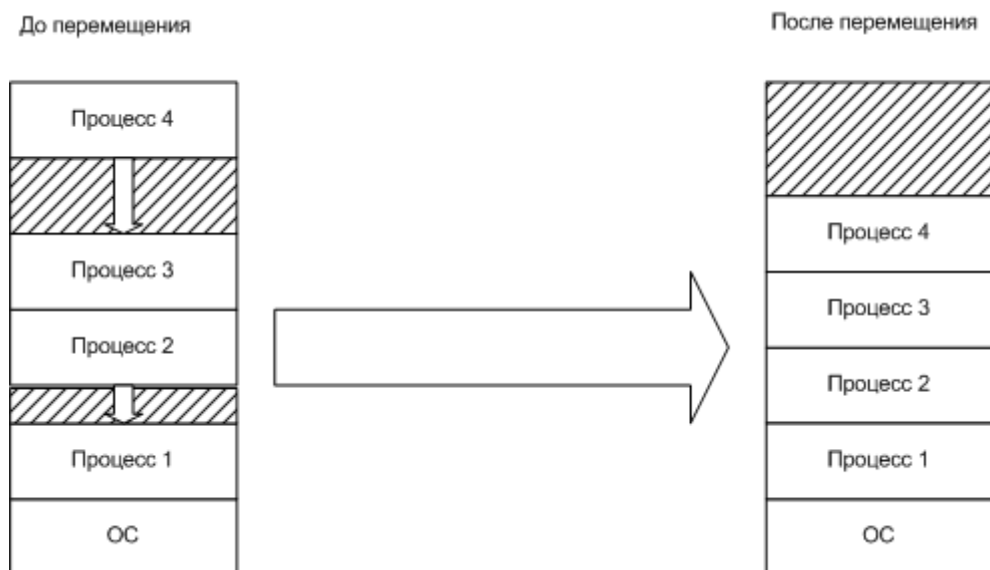
Распределение памяти динамическими разделами.

Недостатки:

- Сложность
- Память фрагментируется

## Перемещаемые разделы

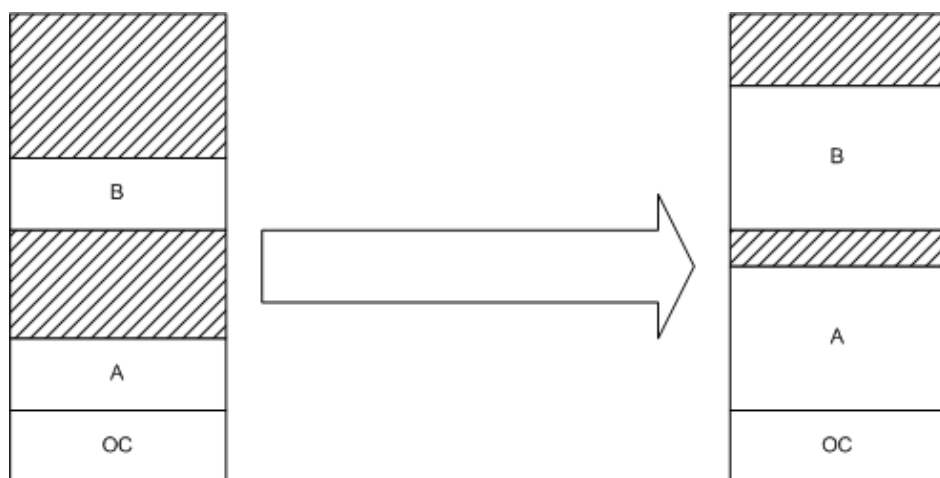
Это один из методов борьбы с фрагментацией. Но на него уходит много времени.



Перемещаемые разделы

## Рост разделов

Иногда процессу может понадобиться больше памяти, чем предполагалось изначально.



Рост разделов

## Настройка адресов и защита памяти

В предыдущих примерах мы можем увидеть две основные проблемы.

- Настройка адресов или перемещение программ в памяти
- Защита адресного пространства каждой программы

Решение обеих проблем заключается в оснащении машины специальными аппаратными регистрами.

- Базовый (указывает начало адресного пространства программы)
- Предельный (указывает конец адресного пространства программы)

### 6.3 Методы с использованием внешней памяти (свопинг и виртуальная память)

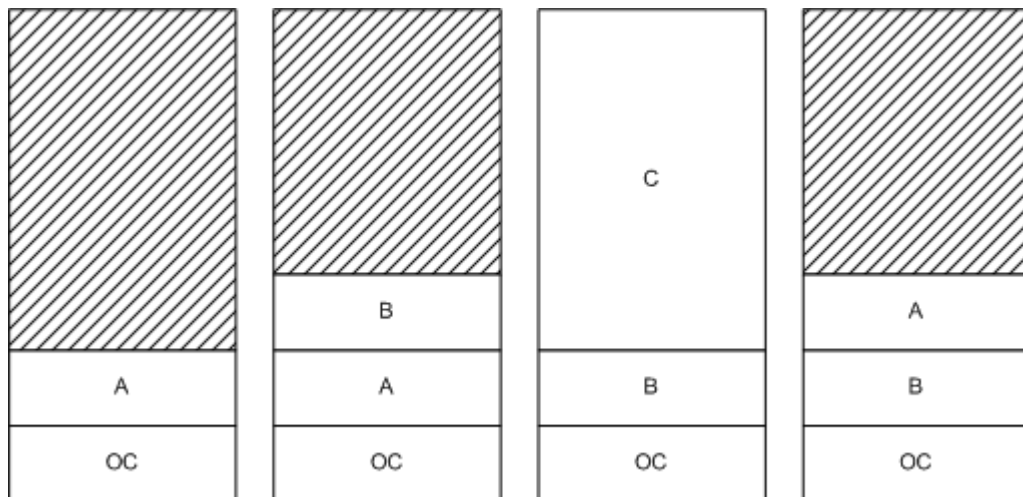
Так как памяти, как правило, не хватает. Для выполнения процессов часто приходится использовать диск.

Основные способы использования диска:

- **Свопинг** (подкачка) - процесс целиком загружается в память для работы
- **Виртуальная память** - процесс может быть частично загружен в память для работы

#### 6.3.1 Свопинг (подкачка)

При нехватке памяти процессы могут быть выгружены на диск.



т.к. процесс **С** очень большой, процесс **А** был выгружен временно на диск, после завершения процесса **С** он снова был загружен в память.

Как мы видим процесс **А** второй раз загрузился в другое адресное пространство, должны создаваться такие условия, которые не повлияют на работу процесса.

**Свопер** - планировщик, управляющий перемещением данных между памятью и диском.

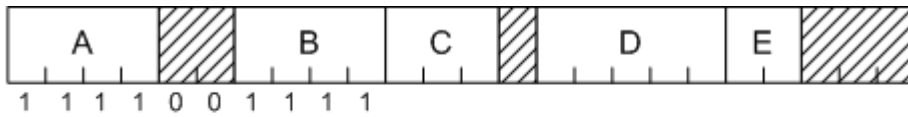
Этот метод был основным для UNIX до версии 3BSD.

#### Управление памятью с помощью битовых массивов

Вся память разбивается на блоки (например, по 32бита), массив содержит 1 или 0 (занят или незанят).

Чтобы процессу в 32Кбита занять память, нужно набрать последовательность из 1000 свободных блоков.

Такой алгоритм займет много времени.



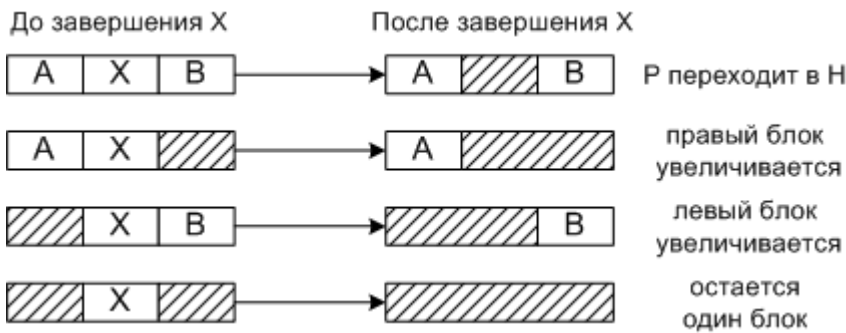
битовые массивы и списки

### Управление памятью с помощью связанных списков

Этот способ отслеживает списки занятых (между процессами) и свободных (процессы) фрагментов памяти.

Запись в списке указывает на:

- занят (P) или незанят (H) фрагмент
- адрес начала фрагмента
- длину фрагмента



Четыре комбинации соседей для завершения процесса X

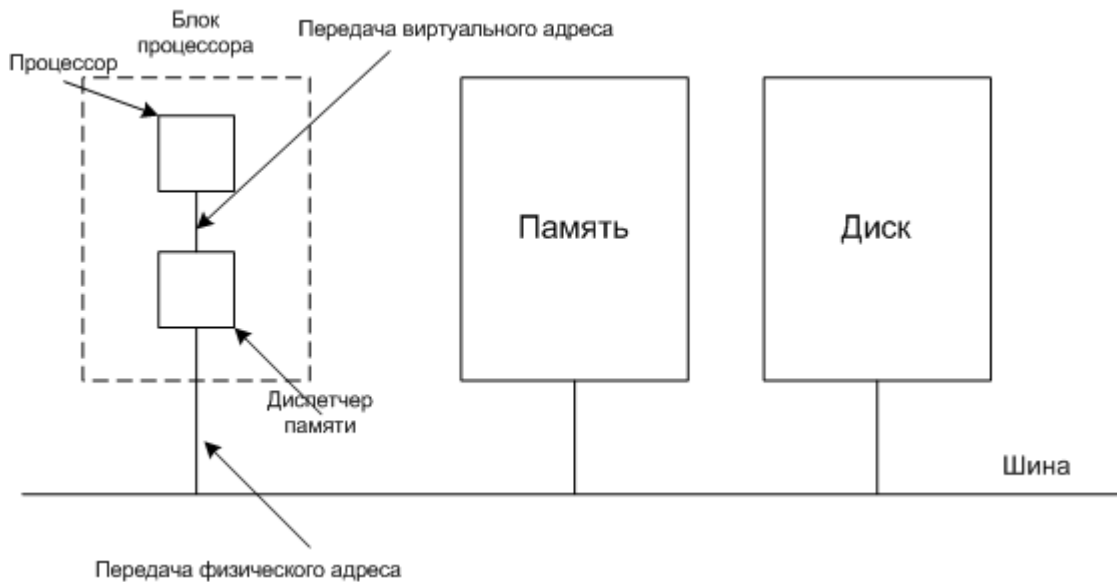
Алгоритмы выделения блока памяти:

- первый подходящий участок.
- следующий подходящий участок, стартует не сначала списка, а с того места на котором остановился в последний раз.
- самый подходящий участок (медленнее, но лучше использует память).
- самый неподходящий участок, расчет делается на то, что программа займет самый большой участок, а лишнее будет отделено в новый участок, и он будет достаточно большой для другой программы.

### 6.3.2 Виртуальная память

Основная идея заключается в разбиении программы на части, и в память эти части загружаются по очереди.

Программа при этом общается с виртуальной памятью, а не с физической.



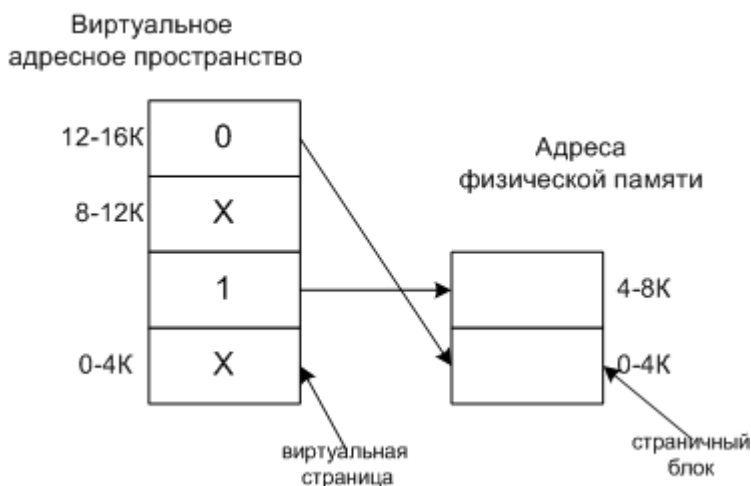
Диспетчер памяти преобразует виртуальные адреса в физические.

### Страничная организация памяти

**Страницы** - это части, на которые разбивается пространство виртуальных адресов.

**Страничные блоки** - единицы физической памяти.

Страницы всегда имеют фиксированный размер. Передача данных между ОЗУ и диском всегда происходит в страницах.



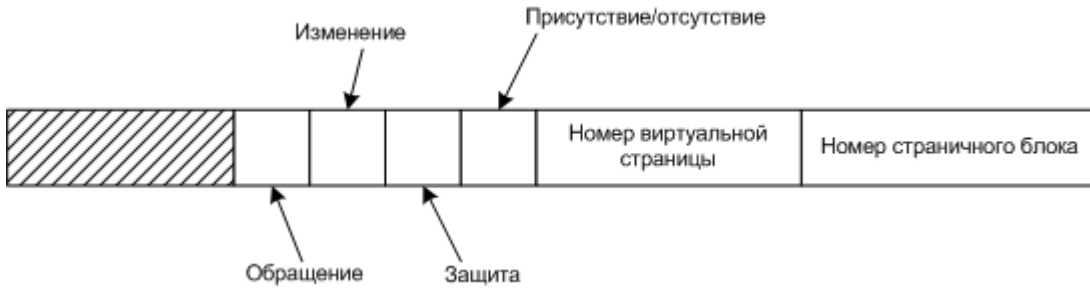
X - обозначает не отображаемую страницу в физической памяти.

**Страничное прерывание** - происходит, если процесс обратился к странице, которая не загружена в ОЗУ (т.е. X). Процессор передается другому процессу, и параллельно страница загружается в память.

**Таблица страниц** - используется для хранения соответствия адресов виртуальной страницы и страничного блока.

Таблица может быть размещена:

- в аппаратных регистрах (преимущество: более высокое быстродействие, недостаток - стоимость)
- в ОЗУ



Типичная запись в таблице страниц

Присутствие/отсутствие - загружена или не загружена в память

Защита - виды доступа, например, чтение/запись.

Изменение - изменилась ли страница, если да то при выгрузке записывается на диск, если нет, просто уничтожается.

Обращение - было ли обращение к странице, если нет, то это лучший кандидат на освобождение памяти.

*Информация о адресе страницы когда она хранится на диске, в таблице не размещается.*

Для ускорения доступа к страницам в диспетчере памяти создают **буфер быстрого преобразования адреса**, в котором хранится информация о наиболее часто используемых страниц.

*Страничная организация памяти используется, и в UNIX, и в Windows.*

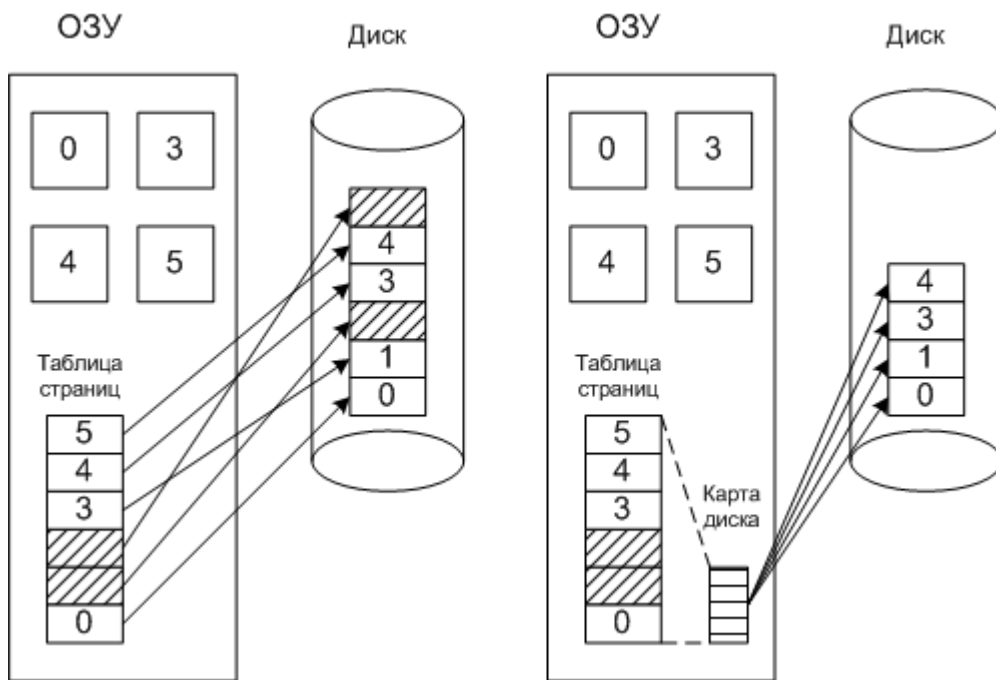
### Хранение страничной памяти на диске

#### Статическая область свопинга

После запуска процесса он занимает определенную память, на диске сразу ему выделяется такое же пространство. Поэтому файл подкачки должен быть не меньше памяти. А в случае нехватки памяти даже больше. Как только процесс завершится, он освободит память и место на диске.

На диске всегда есть дубликат страницы, которая находится в памяти.

Этот механизм наиболее простой.



Статический и динамический методы организации свопинга.

### Динамическая область свопинга

Предполагается не выделять страницам место на диске, а выделять только при выгрузке страницы, и как только страница вернется в память освобождать место на диске.

Этот механизм сложнее, так как процессы не привязаны к какому-то пространству на диске, и нужно хранить информацию (**карту диска**) о местоположении на диске каждой страницы.