**чего ждут процессы в очереди готовых процессов?**

получения кванта

поток -- единица чего?

поток -- единица диспетчеризации

**что такое поток?**

Поток -- непрерывная часть кода программы, которая может выполняться параллельно с другими частями кода

**кто владелец ресурсов в системе?**

процесс

**какой процесс рассматривается как поток? как он называется?**

процесс, в котором не создается потоков. он наз. главным потоком

**Что такое pipe**

системный вызов, создающий неименованный программный канал

**Программный канал**

буфер типа FIFO в области системы (ядра); реализация потоковой модели передачи данных

**Кто может взаимодействовать через неименованные программные каналы? Почему?**

Только процессы-родственники.  
Потому что процессы-потомки наследуют дескрипторы открытых файлов процесса-предка => наследуется дескриптор неименованного программного канала

**Что делает exec? Как говорят и как на самом деле?**

создает таблицу страниц для адресного пространства программы, переданной ему в качестве параметра, а затем заменяет старый адрес новой таблицы страниц.

**Процесс выполнения exec**

1. Начинается с проверок: путь к файлу, права доступа к файлу, заголовок (действительно ли исполняемый)
2. Если проверки пройдены, создается новое адресное пространство для программы, передаваемой exec в качестве параметра (таблицы страниц с дескрипторами
3. Высвобождается старое адресное пространство
4. Устанавливается новое адресное пространство
5. Инициализация аппаратного контекста:

* в регистры заносятся нули
* в IP заносится адрес точки входа программы

1. в CR3 заносится адрес новых таблиц страниц

**какие точки входа ОС есть?**

* системные вызовы
* исключения
* аппаратные прерывания

**Процессы параллельны, то есть ...**

выполняются тогда, когда получат процессорное время

**Мы видим два приглашения к вводу. Что происходит?**

Параллельные процессы блокированы в ожидании завершения ввода: первому процессу был передан квант, он выдал приглашение и блокировался, затем квант был передан второму (и т.д.) в итоге ввести требуется для первого

**Какую проблему решает wait?**

Проблему процессов-сирот (чтобы процесс предок не завершался раньше потомков)

**Что делает exec (по версии какого-то мужика)**

Чего нет у неименованного программного канала?

**Процесс "зомби"**

Процесс, у которого отобраны все ресурсы, кроме последнего (дескриптора)

**Зачем нужен процесс "зомби"**

Чтобы предок смог получить и обработать статус завершения потомка

**Флаги процессов: 1**

Был fork(), но не было exec() -- установлен copy-on-write

**Флаги процессов: 4**

Выполнение процесса с правами super-user

**Флаги процессов: 0**

Нормальное выполнение -- copy-on-write сброшен

**Коды состояния процессов: D**

uninterruptible sleep (usually IO) -- блокировка процесса в ожидании завершения операции ввода-вывода (прервать нельзя)

**Коды состояния процессов: R**

running or runnable (on run queue) -- выполняющийся или находящийся в очереди готовых процессов

**Коды состояния процессов: S**

interruptible sleep (waiting for an event to complete) -- блокировка процесса в ожидании завершения события (может быть прервана)

**Коды состояния процессов: Z**

defunct "zombie" process, terminated but not reaped but its parent

**super-user**

Привелегированный процесс, который может обращаться к структурам и функциям ядра

**Обычный файл**

Данные, которые предполагается хранить длительное время на вторичных носителях

**Hardlink (жесткая ссылка)**

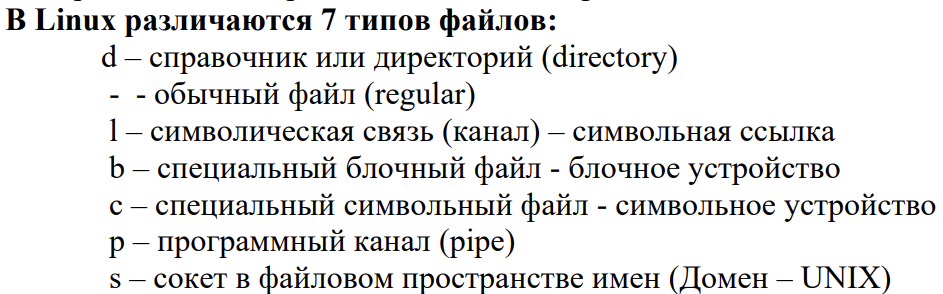
Еще одно равноправное имя файла

**Softlink (мягкая ссылка)**

Специальный файл, содержащий строку символов, которая является путем к файлу

**inode**

индексный дескриптор (информация о файлах, каталогах, кроме данных и имени)



**Когда система проверяет наоичие потомков? Какой процесс запускается? Кто и что получает?**

При завершении любого процесса система проверяет, не остались ли у него незавершенные потомки.   
Если остались, то запускается процесс усыновления терминальным процессом.  
Он получает указатель на нового потомка, а потомки -- указатель на его дескриптор

**fork()**

Системный вызов, используемый для создания процессов в Unix и Unix-подобных ОС

**Что создает fork()? Что наследуется?**

fork() создает новый процесс, который является копией процесса-предка, т.к. наследуются:

* адресное пространство предка
* сигнальная маска
* маска режима создания файлов
* дескрипторы открытых файлов
* обработчики сигналов