

Shell – командный интерпретатор, содержится в командных оболочках, таких как bash, zsh и т.д.

fork() создает новый процесс-потомок. Системный вызов exec() переводит процесс на новое адресное пространство, что делается для запуска на выполнение новой программы, которая передается exec() в качестве параметра.

Линки

Есть хард и софт линки.

Софт линки:

- Могут ссылаться на файлы и каталоги;
- После удаления, перемещения или переименования файла становятся недействительными;
- Права доступа и номер inode отличаются от исходного файла;
- При изменении прав доступа для исходного файла, права на ссылку останутся неизменными;
- Можно ссылаться на другие разделы диска;
- Содержат только имя файла, а не его содержимое.

Хард линки:

- Работают только в пределах одной файловой системы;
- Нельзя ссылаться на каталоги;
- Имеют ту же информацию inode и набор разрешений что и у исходного файла;
- Разрешения на ссылку изменяться при изменении разрешений файла;
- Можно перемещать и переименовывать и даже удалять файл без вреда ссылке.

Опции при создании:

- **-d** - разрешить создавать жесткие ссылки для директорий суперпользователю;
- **-f** - удалять существующие ссылки;
- **-i** - спрашивать нужно ли удалять существующие ссылки;
- **-P** - создать жесткую ссылку;
- **-r** - создать символическую ссылку с относительным путем к файлу;

- **-s** - создать символическую ссылку.

Создание софт линка: `ln -s dest_file_name softlink_name`

Создание хард линка: `ln dest_file_name softlink_name`

Сравнить inode (индексный дескриптор): `ls -li`

Программные каналы (pipe)

Каналы Unix – это механизм IPC (межпроцессное взаимодействие), который перенаправляет вывод одной программы на вход другой программы.

Linux поддерживает программные каналы двух типов: **именованные** и **неименованные**. *Именованные* программные каналы создаются командой **mknod** (в нашем случае на macOS **mkfifo**). Программный канал - это специальный файл, в который можно «писать» информацию и из которого эту информацию можно «читать».

Создание именованных программных каналов:

```
mkfifo pipe_name
```

Работа с программным каналом:

```
echo "kek" > pipe
```

Открываем другой терминал и считываем информацию

```
cat pipe
```

Для чего нужны именованные каналы:

Именованные каналы не занимают память на диске

Если вы выполните `du -s pipe`, вы увидите, что он не занимает места. Это потому, что именованные каналы похожи на конечные точки для чтения и записи из буфера памяти и в него. Все, что записывается в именованный канал, фактически сохраняется во временном буфере памяти, который сбрасывается, когда операция чтения выполняется из другого процесса.

Сниженный ввод-вывод

Поскольку запись в именованный канал означает сохранение данных в буфере памяти, операции с большими файлами значительно сокращают дисковый ввод-вывод.

Связь между двумя очень разными процессами

Выходные данные события можно мгновенно и очень эффективно получить из другого процесса с помощью именованных каналов. Поскольку чтение и запись происходят одновременно, время ожидания практически равно нулю.

Создание безымянных (конвейеры) программных каналов:

```
ls | cat
```

```
echo hey | cat
```

Они создаются на лету вашей оболочкой Unix всякий раз, когда вы используете символ `|`.

Приоритеты процессов

Приоритет процесса определяет, как часто именно этот процесс, по сравнению с другими запущенными процессами, стоящими в очереди на выполнение, будет исполняться процессором. В ОС Linux значение приоритета процесса варьируется в диапазоне значений от -20 до 19 (т.е. получается 40 возможных значений: -20, -19, -18 ... 0, 1, 2 ... 19) и называется ***niceness*** (сокращенно NI).

Чем меньше это значение, тем выше приоритет будет у такого процесса.

Узнать приоритет процесса по его айди: `ps -o pid,comm,nice` **706**

Изменить приоритет процесса по айди: `renice -n 10 -p 1343`

Запуск в фоновом режиме `&` в конце или `nohup`