**Лекция 6**

Работа команды **mv**: сначала делается link, потом unlink (можно обойтись одним сисколом **rename**)

**mkdir** — сискол для создания директории

**rmdir** — для удаления

У каждого файла можно посмотреть маску прав доступа к файлу (ls -l). Она отображает, какие права есть у каждого типа пользователя: владельца; группы пользователей, которая им владеет; остальных.

Права на чтение и на выполнение отличаются: прочитать файл — посмотреть, что там; выполнить — загрузить в процессор бинарный код и выполнить его. Это различие помогает защититься от выполнения вредоносных программ.

**chmod** — поменять права доступа к файлу (также есть сискол chmod). Вместо флагов можно задавать восьмеричную маску прав.

Если убрать право на выполнение директории, то нельзя будет в нее переходить. Если убрать право на чтение, перейти в нее можно будет, но посмотреть ее содержимое будет нельзя.

**chown** — сменить владельца файла

**chgrp** — меняет группу владельцев

**whoami** — посмотреть под каким пользователем находимся

**lsattr** — отобразить атрибуты файла

**chattr** — добавить/убрать атрибут у файла. Реализовано через сискол ioctl

**Файловая система**

Типы файлов в Linux:

1. *Обычные файлы.*

Это файлы, с которыми мы привыкли работать каждый день, они могут содержать текст, исполняемые инструкции для программ, изображения или другую информацию.

1. *Блочные файлы.*

Файлы устройств, которые обеспечивают буферизованный доступ к аппаратным компонентам. При записи данных на жесткий диск или на флешку нет смысла записывать данные сразу же после их поступления. Так мы будем только понапрасну расходовать ресурс устройства и энергию. Можно подождать пока наберется достаточное количество данных а потом записать их за один раз. Эти данные и собираются в буфере. С помощью таких файлов, файловая система и другие утилиты могут обращаться к драйверам аппаратных устройств. Такие файлы могут передать большой блок данных за небольшой один раз.

1. *Символьные файлы.*

Обеспечивают небуферизованный доступ к аппаратным компонентам и ядру. Поскольку у них нет буфера, они позволяют передавать только по одному символу за один раз. А в остальном, это такие же файлы устройств, как и блочные файлы.

1. *Символические ссылки.*

Файлы, которые указывают на другие файлы в системе по их имени. Они могут указывать как на обычные файлы, таки на каталоги или другие типы файлов в linux. По сути, это те же ярлыки Windows. В Linux еще есть жесткие ссылки, но они не имеют отношения к типу файлов, потому что реализованы на уровне файловой системы и считаются обычными файлами. Поскольку они указывают на одно и то же место на диске, это два разных файла, с одинаковым содержимым.

1. *Туннели и именованные туннели.*

Файлы, позволяющие настроить связь между двумя процессами перенаправив вывод одного процесса на вход другого. Именованные туннели используются для связи между двумя процессами и работают так же как и обычные туннели.

Создать тоннель можно с помощью **mkfifo**.

1. *Файлы сокетов.*

Файлы, обеспечивающие прямую связь между процессами, они могут передавать информацию между процессами, запущенными в разных средах или даже разных машинах. Это значит, что с помощью сокетов программы могут обмениваться данными даже по сети. По сути, сокет работает так же как туннели, но только в обе стороны.

1. *Каталоги.*

Это специальные файлы, которые позволяют объединять другие и каталоги в группы для более простой навигации и поиска. Естественно, они могут содержать как обычные, так и специальные файлы, одним словом любые типы файлов ос linux. В системе Linux, файлы организуются в папки начиная от корня (/). Обозначаются каталоги буквой d (directory)

*echo «blablabla» > /dev/null* — так можно проигнорировать поток вывода

*/dev/zero* — фейковое устройство, которое содержит бесконечный поток нулевых байт

*/dev/random* — поток случайных байт

**/proc** — содержит директории процессов, которые в свою очередь содержат информацию об этих процессах в виртуальных файлах

**lsof** — утилита, которая для данного процесса показывает список открытых файлов

*lsof -p num\_of\_proc*

**lscpu** — тулза, которая выводит информацию о процессоре

**lsusb** — выводит информацию о том, какие есть USB порты и что к ним на данный момент подключено

**Монтирование файловых систем**

Подключив некоторый диск, на котором присутствует файловая система, можно отобразить его в дереве файловых папок, чтоб оно легло где-то в поддиректории. То есть если есть у нас устройство, являющееся диском, можно его замапить на какую-то директорию, и тогда она будет отображать дерево файловых папок на этом диске.

**df** — команда, которая отображает, какие есть диски и к каким директориям они подключены. Также позволяет смотреть, сколько осталось свободного места на диске.

**mount** — команда, которая позволяет взять устройство, являющееся диском, и поставить ему в соответствие некоторую директорию.

**umount** — позволяет отмонтировать файловую систему

Также есть аналогичные сисколы.

**df -T** добавит в вывод тип файловой системы.

* FAT - первая файловая система, file allocation table.
* FAT32 - улучшение FAT с 32-битной адресацией.
* NTFS - Для Windows стандартная ФС.
* EXT (EXT2FS, EXT3FS, EXT4FS) - стандартная для Linux.
* BTRFS - (B-tree FS или Better FS) - файловая система для Linux, основанная на B-деревьях. Работает по принципу "копирование при записи" (copy-on-write).