Взято: http://wiki.kryukov.biz/wiki/%D0%9F%D1%80%D0%B0%D0%B2%D0%B0_%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D0%B0

Права доступа в Linux реализованы очень просто. У каждого файла в системе существуют свои собственные права доступа.

Внимание! Права доступа не наследуются так, как это принято в Windows и No

В Windows можно определить права доступа на директорию, и они автоматически распространяются на все файлы и поддиректории. В Linux права доступа сохраняются в inode файла, и поскольку inode у каждого файла свой собственный, права доступа у каждого файла свои.

Если посмотреть на вывод программы ls –l, в первом столбце показаны права доступа файла.

```
 $ ls -l
итого 4
-rw-r--r--  1 artur users 28 2005-06-03 12:50 hardlink
lrwxrwxrwx  1 artur users  4 2005-06-03 12:51 slink -> test
-rw-r--r--  1 artur users  0 2005-06-03 13:21 test
$
```

В первом поле десять символов. Первый символ — это тип файла, остальные девять показывают права. Права доступа делятся на три группы:

```
 r w x | r w x | r w x
user  | group | other
```

- **user** — права хозяина файла.
- **group** — права группы которой принадлежит файл.
- **other** — права всех остальных пользователей системы.

При обращении программы к файлу сначала проверяется, является ли пользователь, с правами которого выполняется программа, хозяином файла? Если да, тогда на программу распространяются права хозяина файла, все остальные права игнорируются. Если нет, тогда проверяется, принадлежит ли файл группе, с правами которой выполняется программа? Если да, тогда применяются права доступа для группы, а все остальные права игнорируются. Если файл не принадлежит ни пользователю, ни группе, с правами которых выполняется программа, тогда применяются права для всех остальных.

Внимание! Права доступа пользователя и группы не суммируются. Если прогр

Права доступа к файлам

Права доступа к файлам и директориям имеют различное значение. Для файлов:

- **r** — право на чтение данных из файла.
- **w** — право на изменение содержимого файла (запись).
- **x** — право на исполнение файла.

Все права достаточно просты в понимании, единственно на что следует обратить внимание — *w* не дает права на удаление файла, только на изменение содержимого.

Теперь о праве на исполнение. Это право можно установить для любого файла. Получается, что потенциально любой файл в системе можно исполнить? Это действительно так. В Linux является ли файл исполняемым или нет, определяется не по его расширению (*понятие расширение файла отсутствует в файловой системе Linux*), а по правам доступа. Если у файла установлено право
x
, его можно запустить на выполнение.

Что происходит, когда мы пытаемся выполнить файл? Мы набираем имя файла в командной строке и нажимаем Enter. В первую очередь проверяется, а имеет ли пользователь права на исполнение этого файла? Если имеет, тогда система смотрит, а это исполняемый бинарный файл? В Linux все исполняемые бинарные файлы в начале

файла имеют заголовок ELF. Если это исполняемый бинарный файл, тогда, согласно его заголовку, происходит распределение оперативной памяти, и управление передается программе.

Если файл не бинарный, тогда считается, что это текстовый файл. В начале первой строки файла ищется последовательность символов #!. После этой последовательности, в той же строке, указывается программа, которую необходимо запустить и передать ей в командной строке текущий файл. Например, для файлов, написанных на языке shell script, первая строка будет выглядеть так:

```
#! /bin/sh
```

Для программ, написанных на perl, так:

```
#! /bin/perl
```

Во всех интерпретируемых языках программирования # — это символ комментария. То есть первая строка считается комментарием и программой не выполняется. При указании интерпретатора можно писать аргументы командной строки. Например:

```
#! /bin/sed -f command
```

Если в файле в первой строке нет этих символов, тогда все зависит о программы оболочки, в которой запускается программа. Если используется bash, то он считает, что файл содержит программу, написанную на языке shell script, запускает копию себя любимого и передает этой копии файл на интерпретацию. Если в файле действительно находится программа, то он ее выполняет. Если в файле находится «Война и мир» графа Льва Николаевича Толстого, то на экране появляются сообщения об ошибках shell script: «Я не знаю оператор Пьер Безухов. Наташа Ростова — это оператор или функция?»

Права доступа к директориям

Права доступа к директориям интерпретируются по-другому:

- **r** — право на чтение директории. Прочитать содержимое директории — получить список файлов.
- **w** — право на изменение содержимого директории — создание и удаление файлов в этой директории.
- **x** — право на «вхождение» в директорию.

Внимание! Обратите внимание на то, что если вы имеете право на запись в директ

Теперь о праве **x** для директории. Это право позволяет Вам войти в директорию: cd dir.

Внимание! Право **x** на директорию среди д

Предположим, что другой пользователь системы (user2) написал программу /home/user2/bin/programm. Права доступа у этой программы r-xr-xr-x. Он попросил Вас ее протестировать. При попытке запуска этой программы пользователем user1 было получено сообщение: Доступ запрещен. Почему могла возникнуть такая ситуация? Дело в том, что при обращении к файлу система сначала проверяет право **x** у всех директорий, стоящих в пути этого файла, и только затем права на сам файл. Если хотя бы у одной директории право
x
отсутствует, доступ к этой директории и всему ее содержимому для Вас запрещается. Например:

/ home/ user2/ bin/ programm
r-x r-x r-- r-x r-x

Как видно из примера, у директории /home/user2 отсутствует право на исполнение. Поэтому доступ к директориям /home/user2, /home/user2/bin и файлу program запрещен.

Предположим, что для вас права доступа к директории /usr/home/user - rw-. Что вы сможете сделать в этой директории? Вообщем то ничего серьезного, сначала надо зайти в директорию, а вам нельзя. Хотя список файлов вы сможете получить, но при этом на экран будет выведена куча предупреждений. А вот удалять, создавать, читать и редактировать файлы в директории вы точно не сможете.

Права доступа к символьным ссылкам

Если посмотреть на права символьных ссылок, то они всегда выглядят так: rwxrwxrwx. Дело в том, что права на символьную ссылку не имеют особого значения. При использования ссылки драйвер файловой системы пересчитывает реальный путь к файлу и применяет права доступа, определенные для реального пути уже без учета символьной ссылки.