File permissions

Contents:

1: Understanding Linux file permissions

Every file and directory on a Linux system has a set of *permissions* that control the kinds of access that different people have.
Permissions come in two parts: *access modes* and *ownership* - the *what* and the *who* respectively.

1.1: Access modes

There are three different kinds of access: *Read*, *Write* and *Execute*.
The meaning of these is slightly different between files and directories:

	Read	Write	Execute
File	Contents can be read	Contents can be changed	Can be run as a script or program
Directory	Files can be listed	Files can be added, deleted, renamed, etc	Can be traversed into/through

1.2: Ownership

Every file and directory also has a *user* and a *group* associated with it.
The user is the account that owns the file, while the group is the group of users (if any) that it's shared with.
In your home directory, the group will usually the same as the user, but in shared directories the group will usually be the project or class it belongs to.

1.3: Permissions

The permissions for a file make up the ownership information, plus three sets of access modes:

- User access
- Group access
- Others access (everyone else)

For example, given a script with yourself as the user and your project group as the group, you might want permissions like this:

	Read	Write	Execute
User	r	w	x
Group	r	-	x
Others	-	-	-

You have full access to the script (`rwx`).

Your project group can read and run the script, but cannot edit it (`r-x`).

Everyone else has no access to the script at all (`---`).

2: Viewing file permissions

To see the current permissions on a file, run `ls -l` on it:

```
$ ls -l myfile
-rwxr-x---    1 jsmith   cs1234project       1918 Mar  5 08:36 myfile
```

The bolded fields are the access-modes string, the user and the group respectively.

The access-modes are the user, group and other modes concatenated together - in this case `rwx`, `r-x` and `---` (the same permissions as in the table above)

There's also a single-character prefix showing the type of object: `-` for files (as in this example), `d` for directories and `l` for soft links.

Note that soft links always show up as `lrwxrwxrwx`, because the actual permissions applied are those of the file the link points to.

3: Setting File Permissions

Be sure to understand Secure File Permissions before changing things, so you don't accidentally give the wrong people access to your files.

You can set the access modes for a file or directory using the `chmod` command.

Run `man chmod` for full details, but as a brief example: `chmod u=rwx g=rx o=rx myfile` will set the permissions on `myfile` to `-rwxr-xr-x`.

You can also set the group of a file using the `chgrp` command: `chgrp groupname myfile` (see `man chgrp` for more details.)

However, you can't change the owner of a file unless you are the root user. If you need to change the ownership of a file, contact System Support.

Last edited by jbc 21/02/2018