ln (make links between files)

This command creates a link to a file or directory. It is almost always used like this:

ln -s TARGET LINK_NAME

The option *-s* instructs *ln* to create a symbolic link, also called *symlink* or *soft link*. I often use this to create a shortcut to a directory. Try executing this in your home directory:

ln -s /usr/local/lib llib

This creates a file called llib that links to /usr/local/lib. You can verify this by calling *ls -l*, which should show the following:

... llib -> /usr/local/lib

If you now type *cd llib*, it will take you to /usr/local/lib.

In the Linux operating system, symlinks to executable files are commonly used to provide alternative names for commands. Try issuing *ls -l /bin*. You'll see that many commands are not executable files themselves, but links to executables. Here are some examples of links in the /bin directory from my system:

lsmod -> kmod
open -> openvt
sh -> dash

This means that whenever I run *lsmod* the program *kmod* is executed, because lsmod is a link to kmod, and so on.

If you leave out the option *-s* when you call *ln*, you'll create a *hard link*. This is most probably not what you want. Hard links are a really cool and advanced feature of the Linux file system, but they can also be confusing. Let me explain. A symbolic link is a kind of pointer to a file. A hard link is the file. It's an alternative name for a file that is equivalent to the original name.

Try the following experiment. Use *df -h* to find out how much disk space you have left. Now find a large file, maybe several gigabytes in size, and create a hard link to it, like this:

ln large.img hardlink

You'll notice that the same size is reported for the hard link as for the original file:

$ ls -lh large.img hardlink
3.8G ... large.img
3.8G ... hardlink

However, if you run *df -h* again, you'll see that the amount of free disk space has not decreased. The hard link looks like a copy of large.img, but it's really just a new name for the file's content. If the file large.img is deleted, the file's content is still accessible under the name hardlink. In fact, large.img is a hard link too! From the perspective of the Linux file system, all files are hard links. Linux has a rule that a file is only deleted ("unlinked") when the last hard link to its content is removed.

The upshot of this little discussion is that hard links are wondrous and confusing things, but they can be immensely useful. For example, when you create multiple backups of the same file you can create hard links to it instead of copies. If the file is large, this will save you a lot of disk space. However, to avoid confusion, you should only do this once you feel comfortable with hard links and

know how to distinguish them from copies. (A little hint: You can use *ls -i* to display files' unique ID numbers, called “inodes”. You'll notice that hard links to the same file have identical inodes.)