Ubuntu Linux Fundamentals
Ubuntu Server - Groups - Hard Links and Soft Links

In this lesson, we'll discuss something we haven't touched on yet, linking files.

Instead of making copies of files all over your system, it is sometimes desirable to just link to one version of the file.

This is done with links.

A link is similar in concept to a shortcut in Windows.

Links are pointers to the actual file.

There are two types. Soft links, often referred to as symbolic links or sym links, and hard links. We'll discuss both in this lesson.

Before discussing that, though, we'll take a look at inode numbers, and what they are. You'll understand why shortly.

inodes

Along with a name, every file is assigned an index or inode number. The inode number is how the system actually keeps track of the file.

Typing `ls -li` (the i is to print inode numbers) will yield a line similar to the following:

```
22514 -rw-rw-r-- 1 theo theo 1167 Nov 10 20:39 file-permissions.txt
```

The leading number, `22514`, is the inode number for the file `file-permissions.txt`.

The number of inodes available on a system is limited, but is very large, depending on the file system in use.

Typing `df -i` will give you the status of inode use on your file system.

```
theo@ubuntu-server:~$ df -i
Filesystem                          Inodes   IUsed  IFree IUse% Mounted
on
udev                                121959     456 121503    1% /dev
tmpfs                               127009     589 126420    1% /run
/dev/mapper/ubuntu--server--vg-root 557056 130883 426173   24% /
tmpfs                               127009       1 127008    1% /dev/
shm
tmpfs                               127009       3 127006    1% /run/
lock
tmpfs                               127009      16 126993    1% /sys/
fs/cgroup
/dev/sda1                           124928     313 124615    1% /boot
```

```
tmpfs                              127009      4 127005    1% /run/
user/1000
```

You can see that inode use is very low on my vm except for `/dev/mapper/ubuntu-server-vg-root`.

You can get more information about a particular mount point with `tune2fs`:

```
sudo tune2fs -l /dev/sda1 | grep -i inode

Filesystem features:      ext_attr resize_inode dir_index filetype
sparse_super large_file
Inode count:              124928
Free inodes:              124615
Inodes per group:         2048
Inode blocks per group:   256
First inode:              11
Inode size:               128
```

Now, lets see how inodes relate to linking to files.

Hard Links

To create a hard link to a file, you just type

```
ln <file-name> <link-name>
```

If you have a file called `locs.txt` and you want to make a hard link to it called `hard-link-to-locs.txt`, you would type

```
ln locs.txt hard-link-to-locs.txt
```

Running `ls -li` on the directory those files are in shows they point to the same inode number.

It is essentially two names pointing to the same underlying file.

Soft Links

To create a soft link, the command is the same, but you add the `-s` option.

If you have a file called `locs2.txt` and you want to make a hard link to it called `soft-link-to-locs.txt`, you would type

```
ln locs2.txt soft-link-to-locs.txt
```

Running `ls -li` on the directory those files are in shows they point to different inode numbers.

A soft link points a new file name to the name of the original file.

`ls -li` will show an arrow pointing from the soft link to the file name it is linked to:

```
soft-link-to-locs.txt -> locs2.txt
```

When listing files, working soft links will show up light blue, and broken ones will show up red by default on a color capable system.

Behavior

Changing the name of a file with a hard link will not break the link, as it is pointing to the inode number, not the file name.

Deleting the file a hard link points to will not break the link either. The hard linked file will just be the only instance of the file left on the system.

Changing the name of a file with a soft link to it will break the link. The soft link will no longer be able to find the file.

Deleting a file with a soft link to it will also break the link.

Practical Use

I most often see links used for system files in web servers like Apache and Nginx. When you want links, potentially in several locations, to point to one "master" file. You make the changes in the original file, and all links are automatically up to date.

More Information

StackOverflow Thread discussing links
https://stackoverflow.com/questions/185899/what-is-the-difference-between-a-symbolic-link-and-a-hard-link