

34 - Hard & Symbolic Links

Hard Links description and their purpose

- Partition file system and description
- Link inode display: `ls -li`

Soft or Symbolic Links and their purposes

- Upgrade large software
- Not having multiple copies of files used commonly by software
e.g. Run Level areas `/etc/init.d/rc2.d` etc
- Shorten access path to a specific Dir or File

How to Create links

- Create links using `ln` and `cp` command

Symbolic Links

```
ln -s sourcefile destination
cp -s sourcefile destination
```

Hard links

```
ln sourcefile destination
cp -l sourcefile destination
```

- Create links using **MC**

Creating a Symbolic link:

Syntax: `ln [options] -s source destination(newlink)`
or `cp -s source destination(newlink)`

eg. `ln -s /bin/cat /home/hans/bin/cat`
Creates a new Symbolic Link called `/home/hans/bin/cat`
pointing to `/bin/cat`

Creating a Hard Link: (files which have the same inode)

Syntax: `ln source destination(newlink)`
or `cp -l source destination(newlink)`
eg. `ln /bin/ping /home/hans/bin/ping`
Creates a new Hard link called `/home/hans/bin/ping`
pointing to `/bin/ping`

Options:

- f, --force remove existing destination files
- i, --interactive prompt whether to remove destinations
- s, --symbolic make symbolic links instead of hard links
- target-directory=*DIRECTORY* specify the *DIRECTORY* in which to create the links
- v, --verbose print name of each file before linking

Important Notes:

Although the man page says that it is possible in to make a hard link to a directory, in reality it is not possible...yet.

Hard links are limited to the same partition as the original

Symbolic links are NOT limited to the same partition as the original

The command `cp source destination` copies the defered file (the file pointed to) when the *source* is a symbolic link.

eg. `cp linktest3 linktest5` (linktest3 is a symbolic link---->linktest)
Copies the content of linktest to linktest5 as a normal file.

The field Nr.2 of the command `ls -l filename` shows how many hard links a file has.

```
eg. ls -l linktest*
-rw-r--r--  3 michel  video  0 2003-11-20 08:45 linktest
-rw-r--r--  3 michel  video  0 2003-11-20 08:45 linktest2
-rw-r--r--  3 michel  video  0 2003-11-20 08:45 linktest3
```

The command `stat filename` give also shows also how many hard links a file has.

```
eg.
stat linktest
File: `linktest'
Size: 0      Blocks: 0      IO Block: 4096  regular empty file
Device: 305h/773d      Inode: 876319      Links: 2
Access: (0644/-rw-r--r--)  Uid: ( 500/ michel)  Gid:( 33/  video)
Access: 2003-11-20 08:45:10.000000000 +0100
Modify: 2003-11-20 08:45:10.000000000 +0100
Change: 2003-11-20 08:45:22.000000000 +0100
```

Symlinks and Hardlinks for directories:

We can create symbolic links but not hard links of directories.

To simulate a hard link for directories we can use the special new capability of the kernel 2.4.x and mount the directories to where we would normally want the hard link.

eg. `ln /usr/local /home/joe/local` doesn't work.
but `mount /usr/local /home/joe/local -o bind` works!

Advantages and disadvantages of each type of links

- Partitions crossing (hard links cannot)
- Deleting or moving the original

Exercise:

```
echo "testline nol" > ~/testfile1
ln ~/testfile1 ~/testlink
rm ~/testfile1
cat ~/testlink
```

- Create a file with a line in it
- hard link the file
- Delete the original file
- Content is the same as the original