

LPIC-1 101-500 – Lesson 22

104.6 Create and change hard and symbolic links



Links

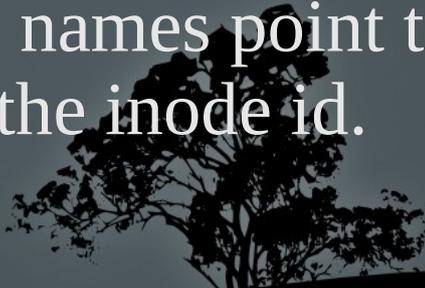
- Sometime the need arises for a file to exist in different locations or with different names
- It will be ineffective to have two copies of the same file because there should be sync mechanism in place to sync changes
- The problem described above can be easily solved with the use of hard links or soft links (aka symbolic links or symlinks)



Symbolic Links (Symlinks)

- **Symlinks** are simple pointers to other files or directories that are called **targets**. Targets can be placed on the same filesystem as the symlink or a separate local or network filesystem
 - They can point to normal files, directories or other types of files
 - They are labeled: `lrwxrwxrwx`, during long listing (`ls -l`)
 - They don't have their own permissions but they point to the permissions of the targets
 - They are a different file from their target and they have a different inode
- 

Hard Links

- Hard links are alternative names for the same file and they share the same inode
 - In effect all file names are hard links pointing to an inode
 - Because they share the same inode they are restricted to the same filesystem
 - They respond somewhat faster
 - They apply only in normal files, not directories or other types
 - There is no easy way to detect which file names point to the same file (inode) unless you look into the inode id.
- 

Find and list symbolic links

- ```
ls -l /lib | grep lrwxrwxrwx # show
all symlinks
under /lib

...
lrwxrwxrwx 1 root root 15 2011-08-
27 07:51 libbz2.so.1 -> libbz2.so.1.0.4
...

find /etc -type l -exec ls -l {} \
find symlinks under /etc

...
lrwxrwxrwx 1 root root 15 2011-10-14
13:51 /etc/blkid.tab -> /dev/.blkid.tab
...
```

There no easy way to tell which symlinks point to a target

# Find and list hard links

- `# ls -li /sbin/mkfs.ext2` # the third column shows that  
# the number of hard links is  
# 5. In the first column we  
# see the inode id

```
2752605 -rwxr-xr-x 5 root root 60408 2011-03-17 20:09
/sbin/mkfs.ext2
```

- `# df /sbin/mkfs.ext2` # find in which  
# filesystem  
# /sbin/mkfs.ext2 belongs

| Filesystem | 1K-blocks | Used     | Available | Use% |
|------------|-----------|----------|-----------|------|
| Mounted on |           |          |           |      |
| /dev/sda2  | 124958620 | 74728000 | 43883052  | 64%  |
| /          |           |          |           |      |

- `# find / -xdev -inum 2752605` # find all file names with  
**inode=2752605**. The `-xdev` option forces `find` to look  
only the the root filesystem (/)

# Create symlinks with `ln`

- `$ ln -s sfile.test symlink.test # create a symlink  
symlink.test with the sfile.test file as a target in the  
working directory`
- `$ ls -li s*.test # target and symlink have  
# different inodes`  
  
`131724 -rw-r--r-- 1 theo theo 0 2011-11-25 07:34  
sfile.test  
131726 lrwxrwxrwx 1 theo theo 10 2011-11-25 07:35  
symlink.test -> sfile.test`
- `$ ln -s /etc/fstab # create a symbolic link with the same name  
as the target, in the working directory`
- `$ ls -l fstab # verify the symlink`  
`lrwxrwxrwx 1 theo theo 10 2011-11-25 07:41 fstab ->  
/etc/fstab`
- `$ ln -s /etc/fstab fstab.slink # create a symlink named  
fstab.slink and target /etc/fstab, in the current directory`

# Create symlinks with `ln`

- `$ ln hfile.test hardlink.test # create hard link hardlink.test using as target the inode of file hfile.test in the current directory`
- `$ ls -li h*.test # both files share the same # inode`  
`131728 -rw-r--r-- 2 theo theo 0 2011-11-25 07:48 hardlink.test`  
`131728 -rw-r--r-- 2 theo theo 0 2011-11-25 07:48 hfile.test`
- `# ln /usr/bin/ssh /usr/local/bin/ssh-new # create a new hard-link ssh-new in another directory. Both files must be on the same filesystem!`
- `$ ln /usr/bin/ssh # we are not allowed to # create hard links to a # file we are not owners!`  
`ln: creating hard link './ssh' => '/usr/bin/ssh':`  
`Operation not permitted`

# Properties of symbolic links

- `$ chmod 640 symlink.test # this will change the permissions of the target not the symlink`
- `$ ls -l s*.test # only the permissions of the target have # changed`  
`-rw-r----- 1 theo theo 0 2011-11-25 07:34 sfile.test`  
`lrwxrwxrwx 1 theo theo 10 2011-11-26 06:09 symlink.test -> sfile.test`
- `$ rm symlink.test # deletes the symlink not the target!`
- `$ rm sfile.test # deleting the target will result in a broken link`

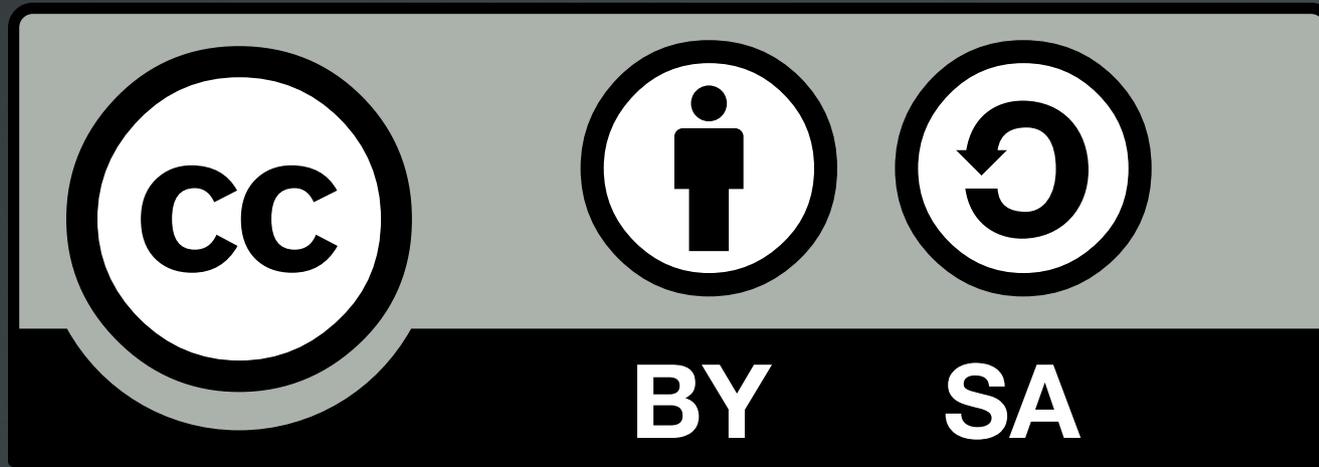


# Properties of hard links

- `$ chmod 640 hardlink.test # this will change the permissions of all hard links of a file because they share the same inode`
- `$ ls -li h*.test # verify the permissions`  
131728 -rw-r----- 2 theo theo 0 2011-11-25 07:48 hardlink.test  
131728 -rw-r----- 2 theo theo 0 2011-11-25 07:48 hfile.test
- `$ rm hardlink.test # deleting a hard link does not delete the file. It just reduces the number of hard links`
- `$ ls -li h*.test # only 1 hard link remains`  
131728 -rw-r----- 1 theo theo 0 2011-11-25 07:48 hfile.test
- `$ rm hfile.test # deleting the last hard link point to the 131728 inode will delete the file definitely!`



# License



The work titled "LPIC-1 101-500 – Lesson 22" by Theodotos Andreou is distributed with the Creative Commons Attribution ShareAlike 4.0 International License.

