* permissions
* Permissions determine who can access files
* And specify who can read, write, modify files/directories on a system

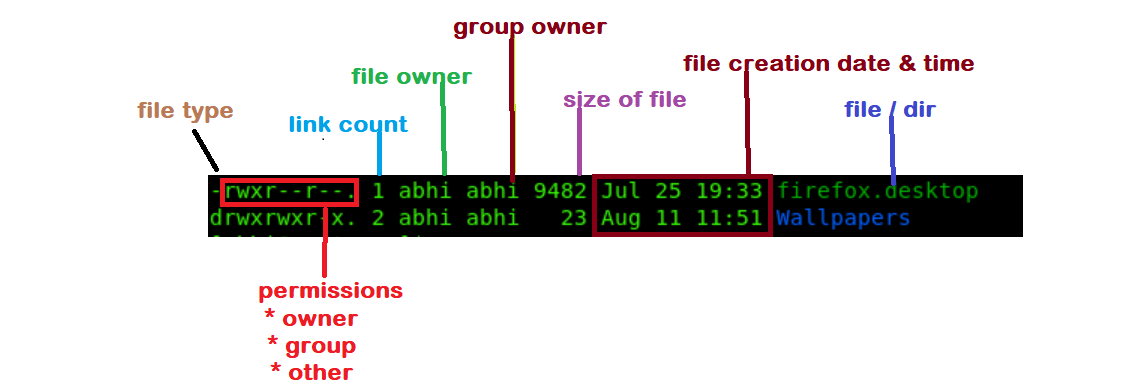
Note : Use ls -l command to check File/Dir permissions

>ll

>ll -d dirname ---to check particular Dir permission

>ll filename ---to check particular file permission

>ls -ltr



1. File type
2. Owner Permissions.
3. Group Permissions.
4. Other User Permission.
5. Link Count.
6. Owner of file/directory.
7. Group Owner of File or Directory.
8. File Size.
9. Creation Date and Time.
10. File/Directory Name

File types in Linux:

- Normal file >text files, binary files etc..

d Directory

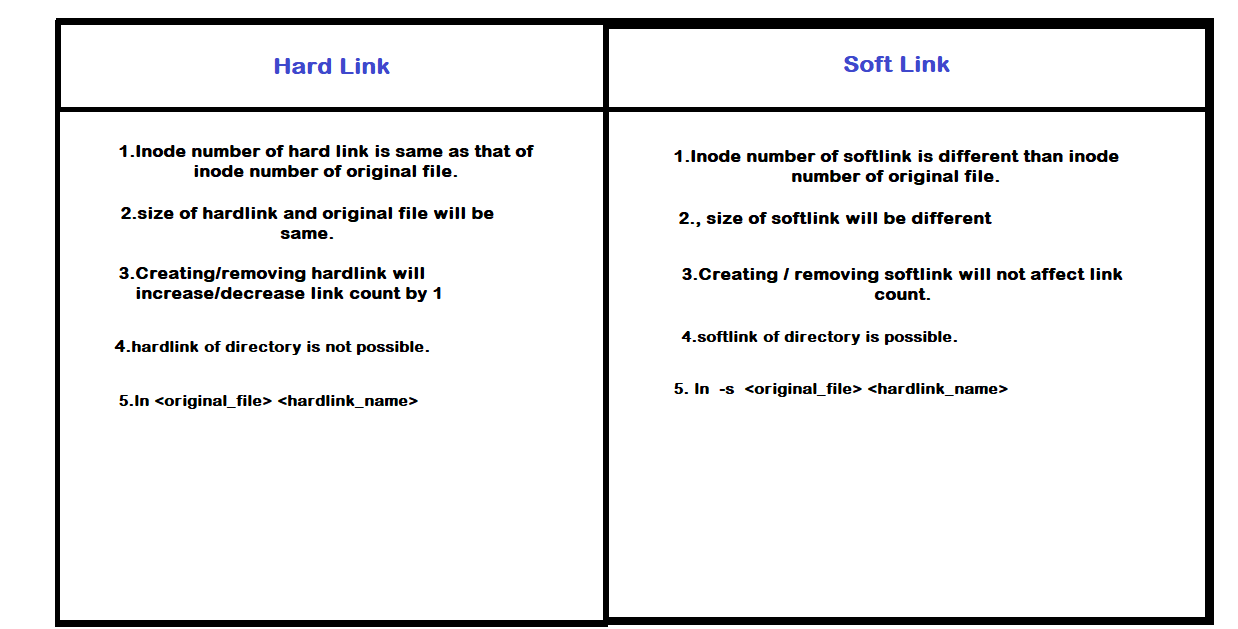
l Link file

***Link Count:*** It shows count of links of file/directory that has been created.

>default link count for Dir =2

>default link count for File =1

Diff between Hard link and soft link



Inode number: inode number is generated to keep track of all the files on a Linux system

User and group ownership:

owner --->current user, who created the file

Group --->primary group of file owner

other --->other users

read r 4 --->open or view/list

write w 2 --->edit

execute x 1 --->run

change owner and group owner

**Lets take an example, “demo” dir owner and group is *root***



Now I want to make user *abhi* the owner of the dir “demo”for that use the following command

*#chown abhi demo --->change owne*r



Same with the group

*#chgrp grp-name demofile.txt --->change group owner*

*#chown user:Grp demofile.txt --->change owner,grp at once*

Giving permissions to user/group/others individually:

chmod u+w demofile.txt --->add user permission to write

chmod u-w demofile.txt --->remove

chmod g+rwx demofile.txt --->add rwx perm. to grp owner

chmod g-r,o=rw demofile.txt --->

chmod go=r dir1 --->assign grp and owner read permission

Giving permissions to u-g-o at once :

chmod ugo+rw demofile.txt

or

chmod a+rwx demo.txt --->a-all at once

octal vs binary

Octal Binary Permission

1 001 --x

2 010 -w-

3 011 -wx

4 100 r--

5 101 r-x

6 110 rw-

7 111 rwx

Default Permission for root and local user

User Directory File

Root 755 644

localUser 775 664

Giving permissions to user/group/others using octal number:

#chmod 421 /abhi/

#ll -d /abhi/

dr---w---x 2 root root 11 aug 23 07:10 /abhi/

#chmod 642 demofile.txt

-rw-r--w- 1 root root 11 aug 23 07:11 demofile.txt

Umask

-->umask is used to set default permissions for newly created files and directories

Calculate umask:

full permission - default permission =umask

>For root user default umask is 022

*777 - 755 =022 root -->dir*

*666 - 644 =022 root -->file*

>For local user default umask is 002

*777 - 775 =002 local user -->dir*

*666 - 664 =002 local user -->file*

#umask [to check current umask]

#umask 000 [to set umask value temp]

#vi /etc/bashrc or /etc/profile [permanently set umask for all users]

#source /etc/bashrc [to apply changes]

Calculate umask:

0 : read, write and execute

1 : read and write

2 : read and execute

3 : read only

4 : write and execute

5 : write only

6 : execute only

7 : no permissions

Ex. Suppose you want to set umask 013 which means u-g-o will have following permissions

User --->read, write and execute

Group --->read and write

Others --->read only

(max perm) 777 - 013 (umask) =764 [ rwx rw- r-- ]

special permission

-->special permisssions allows user to acess command/file with owner permission

suid ->

Ex. We all know local user doesn’t have permission to run “dmidecode” command. If the root user gave suid permission for this command then local user will able to run this command

It means no which user run this command command it will execute with root permissions

*#Chmod u+s /sbin/dmidecode*

s ----to allow acess

S ---to deny acess

sgid ->

if we give SGID permission to particular directory and if file created in that directory by root user or local user that files will get directory group ownership automatically.

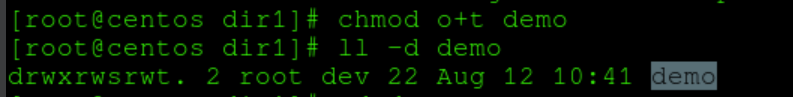




sticky bit ->prevent file from deletion

If Sticky bit is enabled on a folder, the folder contents are deleted by only owner who created them and the root user, no one else can delete other users data in this folder







ACL -Access Control List

ACL use to set permission over file and directory to specific user or specific group.

#getfacl --to check permission

#setfacl --to apply acl permission

#setfacl -m u:sam:rw file.txt



-x to remove



Set acl to group dev:



To remove group acl :

