File and Directory Permissions

File and Directory Permissions

After executing the command ls -l we have the following:

```
drwx------ 35 ubuntu ubuntu 1120 Dec 27 19:54 Desktop

-rwxrw-r-- 65 ubuntu ubuntu 1120 Dec 28 18:36 test.txt

lrwxr-x--- 65 ubuntu ubuntu 1120 Dec 28 18:36 file.lnk
```

we will talk about each part from above (all the Values, Numbers , names , ... etc)

File and Directory Permissions

as you can see we have three different types of file and the symbols (**d , - , l**)

d -> Directory

- -> Regular File

l -> Link (symbolic Links)

File and Directory Permissions

also we have three different Permissions and the symbols (**r , w , x**)

r - Read Permission

w - Write Permission

x - Execute Permission

File and Directory Permissions

Permissions for Files vs Directory:

Permission	File	Directory
r (4 as binary)	Allows files to be read	allows files inside directory to be read
w (2 as binary)	Allows files to be modified	allows entries inside directory to be modified
x (1 as binary)	Allows files to be executed	allows us to go inside directory (using cd command)

File and Directory Permissions

Permissions Categories:

u - User

g - Group

o - Other

a - All

Most Important Commands We Will Use

ls -l - list Files with Long List Format

chmod - change mode of file or Directory

chmod 777 - change mode of file or Directory

chmod ugo=rwx - change mode of file or Directory

id - print real and effective user and group IDs

Most Important Commands We Will Use

groups - print the groups a user is in

chown - change file owner and group

chgrp - change group ownership

Linux Special Permissions

These permissions allow the file being executed to be executed with the privileges of the owner or the group owner as well.

s or S instead of x bit

s == file/Directory already has x bit

S == file/Directory Doesn't has x bit (executable not allowed or set)

t or T instead of x bit

t == Directory already has x bit

T == Directory Doesn't has x bit (executable not allowed or set)

Linux Special Permissions (Cont.)

Three special permissions: SUID(setuid) , SGID (setgid) and sticky bit:

SUID: is a special permission assigned to a file. These permissions allow the file being executed to be executed with the privileges of the owner. For example, if a file was owned by the root user and has the setuid bit set, no matter who executed the file it would always run with root user privileges.

SGID: When the Set Group ID bit is set, the executable is run with the authority of the group. For example, if a file was owned by the users' group, no matter who executed that file it would always run with the authority of the user's group.

sticky bit: When the sticky bit is set on a directory, only the root user, the owner of the directory, and the owner of a file can remove files within said directory.

Linux Special Permissions Examples

chmod 0777 - Full permission but with no special Permission

chmod 4777 - Full permission but with setuid bit

chmod 2777 - Full permission but with setgid bit

chmod 1777 - Full permission but with sticky bit

chmod ugo+s - (s) increase the Special Permission (setuid , setgid)

chmod ugo+t - (t) increase the Special Permission (sticky bit)