LAB :: File Permissions

- Files in Linux means everything. Files are files, devices are files, directories are files everything is written in the file. All the files in the system have permissions that allow or prevent others to view, modified, delete and execute the files. In this LAB we will see how to assign or revoke file permission in the system.
- OS Ubuntu 14.04

Login to your server

- Windows: use puTTY
- Mac and Linux: use your terminal
- Username `apnic` and password `training`
- Login to your server using the above username and password.

User permission:

1. owner (u) : the users who own the file or create the file.
2. group (g) : the group to which the user belongs
3. others (o) : others (not the owner or the owners group)
4. all (a) : everyone (u, g, o)

Access permission:

1. read (r, 4) : read permission
2. write (w, 2) : write permission
3. execute (x, 1) : execute permission

All users have full permission set on their home directory. Only `root` has full access to the system.

File access permission tables:

To view the file permission of a file, perform the below command.

```
ls -l

-rw-rw-r-- 1 sysadmin1 sysadmin1 680 Apr 17 11:22 myfile
```

Let's explain it by converting this output in a table.

1	2	3	4	5	6	7
Access permission	**Link count**	**Ownership**		**File Size**	**Last modified date**	**File name**
		user/owner	**group**			
-rw-rw-r--	1	sysadmin1	sysadmin1	680	Apr 17 11:22	myfile

Let's work with column 1 means Access permission column.

type	**user**			**Group**			**others**		
1	2	3	4	5	6	7	8	9	10
-	**r**	**w**	**-**	**r**	**w**	**-**	**r**	**-**	**-**
	4	**2**	**0**	**4**	**2**	**0**	**4**	**0**	**0**
total	**6**			**6**			**4**		

From the above table, we can see that we have

1. read (r) and write (w) permission for user (u)
2. read (r) and write (w) permission for group (g)
3. read (r) permission for others (o)

Which can be achieved by the following command.

1. chmod **u+rw** myfile
2. chmod **g+rw** myfile
3. chmod **o+**r myfile

Also can be represent using numbers like below:

1. chmod **664** myfile

Exercise 1: Permission with number:

	user			**group**			**others**		
1	2	3	4	5	6	7	8	9	10
type	**r**	**w**	**X**	**r**	**w**	**x**	**r**	**w**	**x**
value	**4**	**2**	**1**	**4**	**2**	**1**	**4**	**2**	**1**
input									
add	**column 2 + 3 + 4 input**			**column 5 + 6 +7 input**			**column 8 + 9 + 10 input**		
chmod	Place result here			Place result here			Place result here		

1. Read permission: 4
2. Write Permission: 2
3. Execute Permission: 1

Command: chmod __ __ __ filename

Exercise 2: Permission with text:

user:

	1	2	3
	r	**w**	**x**
chmod			

chmod u+ __ __ __ filename
chmod u- __ __ __ filename

group:

	1	2	3
	r	**w**	**x**
chmod			

chmod g+ __ __ __ filename
chmod g- __ __ __ filename

other:

	1	2	3
	r	**w**	**x**
chmod			

chmod o+ __ __ __ filename
chmod o- __ __ __ filename

Additional file permission: Sticky Bit

The sticky bit is useful for publicly writable directories. By setting the sticky bit users are prevented from deleting or renaming any files that they do not personally own.

Assign sticky bit: `chmod +t directoryname.`
Remove sticky bit: `chmod -t directoryname.`

Exercise 3: Set the sticky bit

1. Login with `apnic` user.
2. Create a folder name pub using `sudo mkdir /opt/pub` command.
3. Check the file permission `ls -l /opt/`
4. The owner of the directory should be `root:root` and access mode should be `drwxr_xr_x` means only root can add/delete files from the directory.
5. Make this pub directory publicly accessible by using `sudo chmod 777 /opt/pub/`
6. Create a file whithout `sudo`. `touch /opt/pub/apnic_file`
7. Login with the `sysadmin1` user and create a file using `touch /opt/pub/sysadmin1_file`
8. As the permission is set to `777` anyone can delete files from the pub directory.
9. Try to delete `apnic_file` from `/opt/pub` directory while login as `sysadmin1` user.
10. The file will be deleted.
11. Now we will set the sticky bit on pub directory. Type `sudo chmod +t /opt/pub`
12. Check the file permission again `ls -l /opt/`. There should be a "`t`" after `drwxrwxrwx.`
13. Now login again with `apnic` user and create a file in `/opt/pub` directory.
14. At this stage we have 2 files in that `pub` directory own by user `apnic` and user `sysadmin1`.
15. As you are login with `apnic` user try to delete the `sysadmin1` user file.
16. As sticky bits are set, the file own by `sysadmin1` cannot be deleted by user `apnic`.

******END of LAB******