Haile Melaku
Posted on Sep 2, 2022 • Updated on Sep 6, 2022

Shell Permissions

#linux #bash #permissions

So Linux operating system are not only multitasking but also multi-user(different users).

multi-user means that more than one person can be operating the computer at the same time.

Now let's see how to read and set file permissions in different format.

File Permissions

Linux system has permission for different files and directory's assigned access rights for the owner or user(u) of the file, the members of a group(g) of related users, and everybody(o) else.

Linux system divides the authorization in to two

- Ownership

- Permission

Linux File or Dir Ownership

There are 3 types of owner

User By default, the person who created a file becomes its owner
Group contains multiple user and all users in a group have the same permission.
Other everybody else, anybody has permission

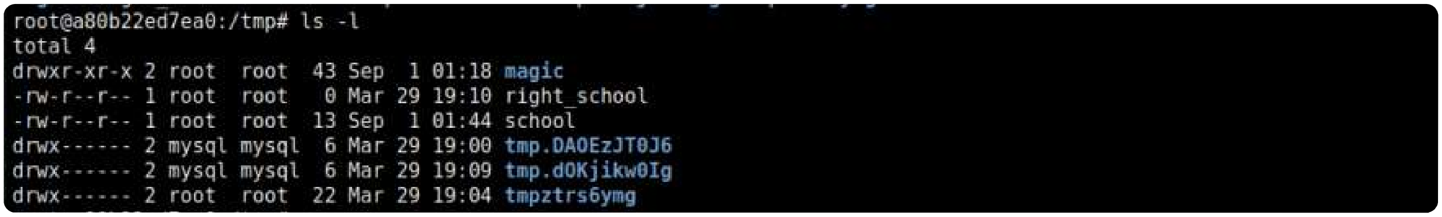
Linux File or Dir Permissions

There are 3 types of permission

Read gives permission to open and read files and the ability to lists its content on directory.
Write gives permission to modify a file and on a directory to add, remove and rename files stored in the directory.
Execute gives permission to execute or run a file.

To view the permission of a file or Dir use the command:

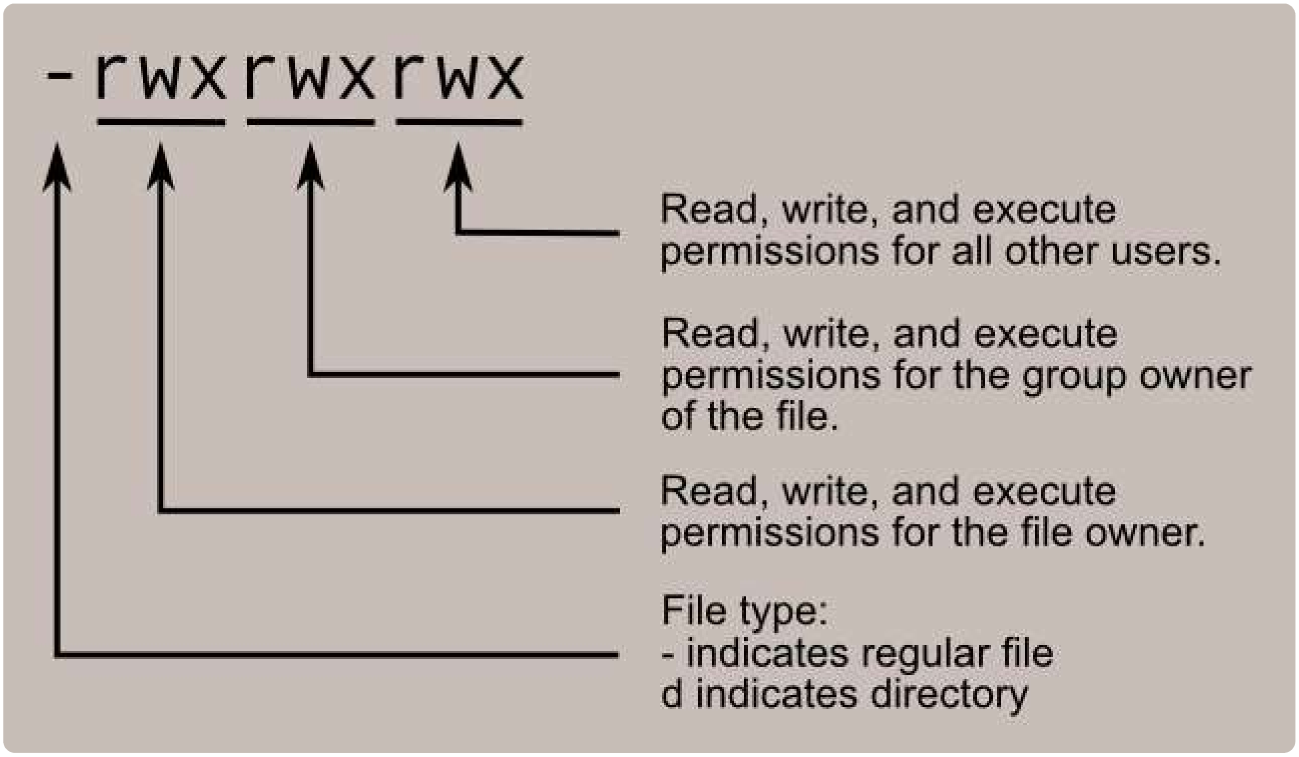
```
ls -l
```



```
root@a80b22ed7ea0:/tmp# ls -l
total 4
drwxr-xr-x 2 root  root  43 Sep  1 01:18 magic
-rw-r--r-- 1 root  root   0 Mar 29 19:10 right_school
-rw-r--r-- 1 root  root  13 Sep  1 01:44 school
drwx------ 2 mysql mysql  6 Mar 29 19:00 tmp.DAOEzJT0J6
drwx------ 2 mysql mysql  6 Mar 29 19:09 tmp.dOKjikw0Ig
drwx------ 2 root  root  22 Mar 29 19:04 tmpztrs6ymg
```

r = read permission
w = write permission
x = execute permission
– = no permission

The first '–' implies that we have selected a file.


-rwxrwxrwx
Read, write, and execute permissions for all other users.
Read, write, and execute permissions for the group owner of the file.
Read, write, and execute permissions for the file owner.
File type:
- indicates regular file
d indicates directory

We use the command chmod to set permission

```
chmod <permission> <file and dir>
```

There are two ways of setting a permission

- Absolute mode
- Symbolic mode

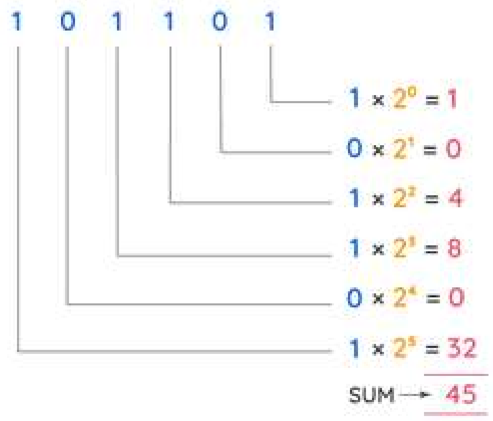
Absolute(Numeric) Mode in Linux

The absolute(numeric) mode uses numeric format to specify ownership and permission.

To really learn how find the number we need to learn how to turn binary's into numbers, don't worry if you don't get it you can just memorize the numbers but turning binary into numbers is a really easy way to do it.

Binary to numbers

It is really easy to turn binary to number just multiply each bit with 2^n then add all the number to get the decimal.



Example

```
110 = 2^2 * 1 + 2^1 * 1 + 2^0 *0 = 4*1+2*1+1*0 = 4+2+0 = 6
100 = 2^2 * 1 + 2^1 * 0 + 2^0 *0 = 4*1+2*0+1*0 = 4+0+0 = 4
```

Here is how absolute(numeric) mode works

```
 rwx rwx rwx = 111 111 111 = 777
rw- rw- rw- = 110 110 110 = 666
rwx --- --- = 111 000 000 = 700
```

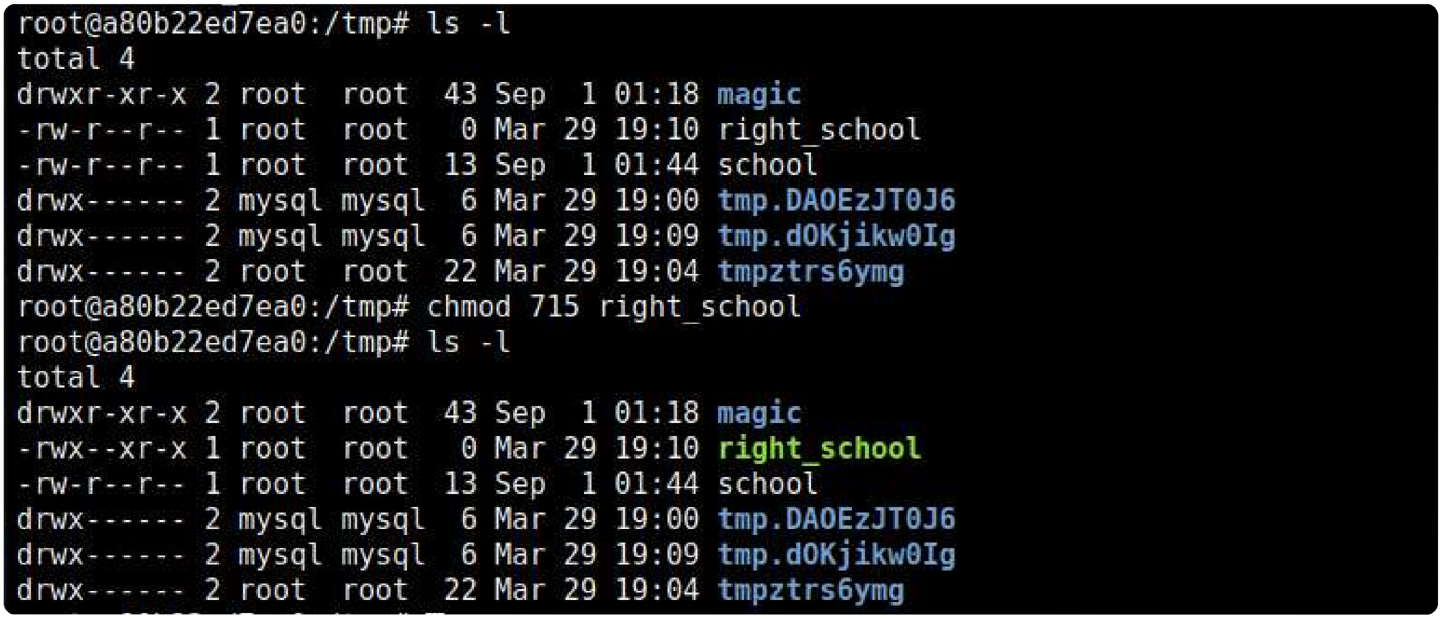
So basically we consider **-** as **0** bit and **r,w or x** as **1** bit

```
 rwx = 111 = 2^2*1 + 2^1*1 + 2^0*1 = 7
rw- = 110 = 2^2*1 + 2^1*1 + 2^0*0 = 6
r-x = 101 = 2^2*1 + 2^1*0 + 2^0*1 = 5
r-- = 100 = 2^2*1 + 2^1*0 + 2^0*0 = 4
--- = 000 = 2^2*0 + 2^1*0 + 2^0*0 = 0
```

Now that we understand how the mode works we need to use it

To give permission of -rwx--xr-x = 715 use the command

```
chmod 715 <file_name or dir name>
```



```
root@a80b22ed7ea0:/tmp# ls -l
total 4
drwxr-xr-x 2 root  root  43 Sep  1 01:18 magic
-rw-r--r-- 1 root  root   0 Mar 29 19:10 right_school
-rw-r--r-- 1 root  root  13 Sep  1 01:44 school
drwx------ 2 mysql mysql  6 Mar 29 19:00 tmp.DAOEzJT0J6
drwx------ 2 mysql mysql  6 Mar 29 19:09 tmp.dOKjikw0Ig
drwx------ 2 root  root  22 Mar 29 19:04 tmpztrs6ymg
root@a80b22ed7ea0:/tmp# chmod 715 right_school
root@a80b22ed7ea0:/tmp# ls -l
total 4
drwxr-xr-x 2 root  root  43 Sep  1 01:18 magic
-rwx--xr-x 1 root  root   0 Mar 29 19:10 right_school
-rw-r--r-- 1 root  root  13 Sep  1 01:44 school
drwx------ 2 mysql mysql  6 Mar 29 19:00 tmp.DAOEzJT0J6
drwx------ 2 mysql mysql  6 Mar 29 19:09 tmp.dOKjikw0Ig
drwx------ 2 root  root  22 Mar 29 19:04 tmpztrs6ymg
```

Symbolic Mode in Linux

The Symbolic mode uses Symbols to modify permissions of a specific owner and use of mathematical symbols to modify the Unix file permissions.

```
 + Adds a permission to a file or directory
– Removes the permission
= Sets and overrides the permissions set earlier.
```

Owners are represented as

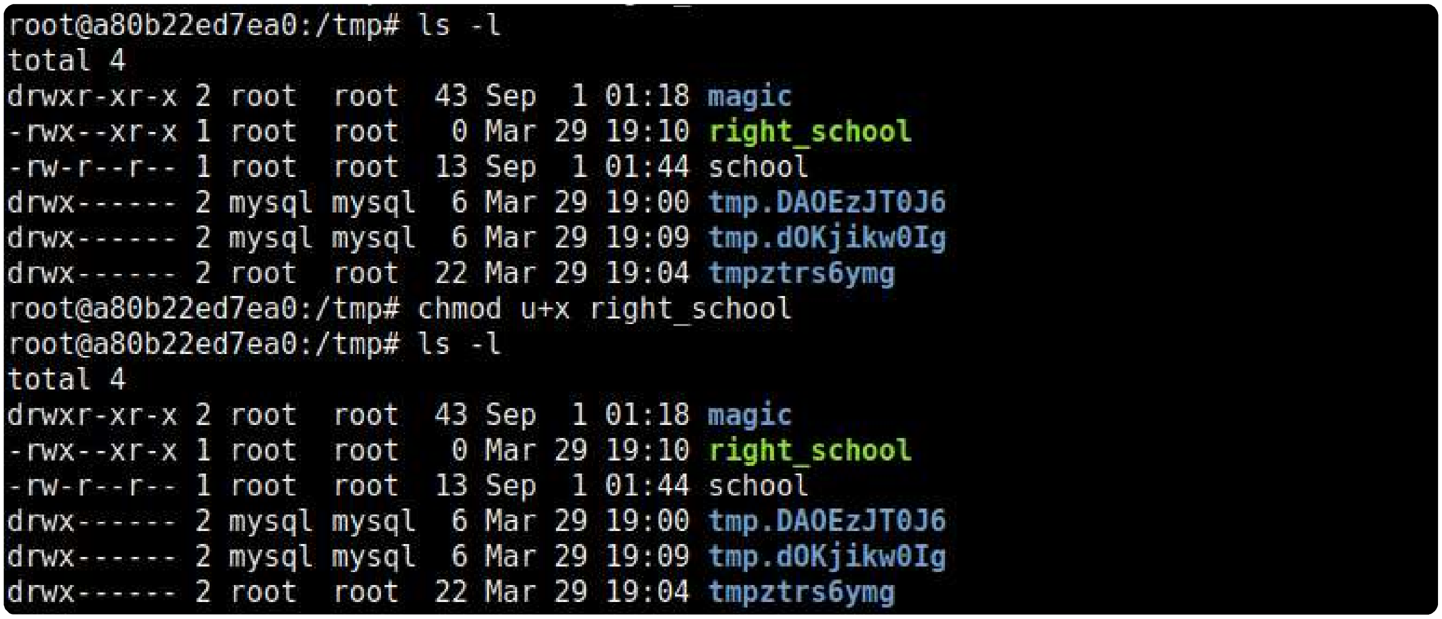
```
 u user/owner
g group
o other
a all
```

How to use this

Adding permission

to add execute permission to the user use the + operator

```
 chmod u+x <file or dir>
```

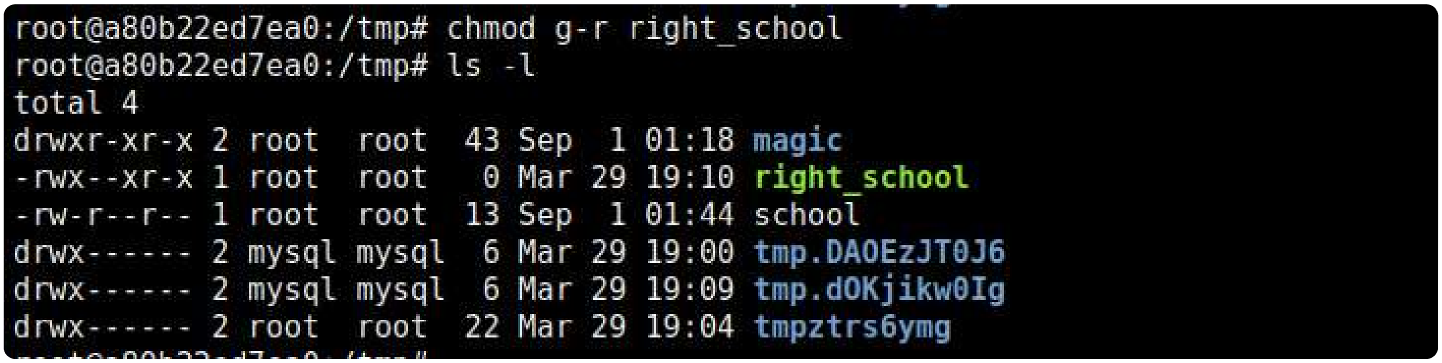


```
root@a80b22ed7ea0:/tmp# ls -l
total 4
drwxr-xr-x 2 root  root  43 Sep  1 01:18 magic
-rwx--xr-x 1 root  root   0 Mar 29 19:10 right_school
-rw-r--r-- 1 root  root  13 Sep  1 01:44 school
drwx------ 2 mysql mysql  6 Mar 29 19:00 tmp.DAOEzJT0J6
drwx------ 2 mysql mysql  6 Mar 29 19:09 tmp.dOKjikw0Ig
drwx------ 2 root  root  22 Mar 29 19:04 tmpztrs6ymg
root@a80b22ed7ea0:/tmp# chmod u+x right_school
root@a80b22ed7ea0:/tmp# ls -l
total 4
drwxr-xr-x 2 root  root  43 Sep  1 01:18 magic
-rwx--xr-x 1 root  root   0 Mar 29 19:10 right_school
-rw-r--r-- 1 root  root  13 Sep  1 01:44 school
drwx------ 2 mysql mysql  6 Mar 29 19:00 tmp.DAOEzJT0J6
drwx------ 2 mysql mysql  6 Mar 29 19:09 tmp.dOKjikw0Ig
drwx------ 2 root  root  22 Mar 29 19:04 tmpztrs6ymg
```

Remove permission

to remove read permission to the group use the - operator

```
chmod g-r <file or dir>
```

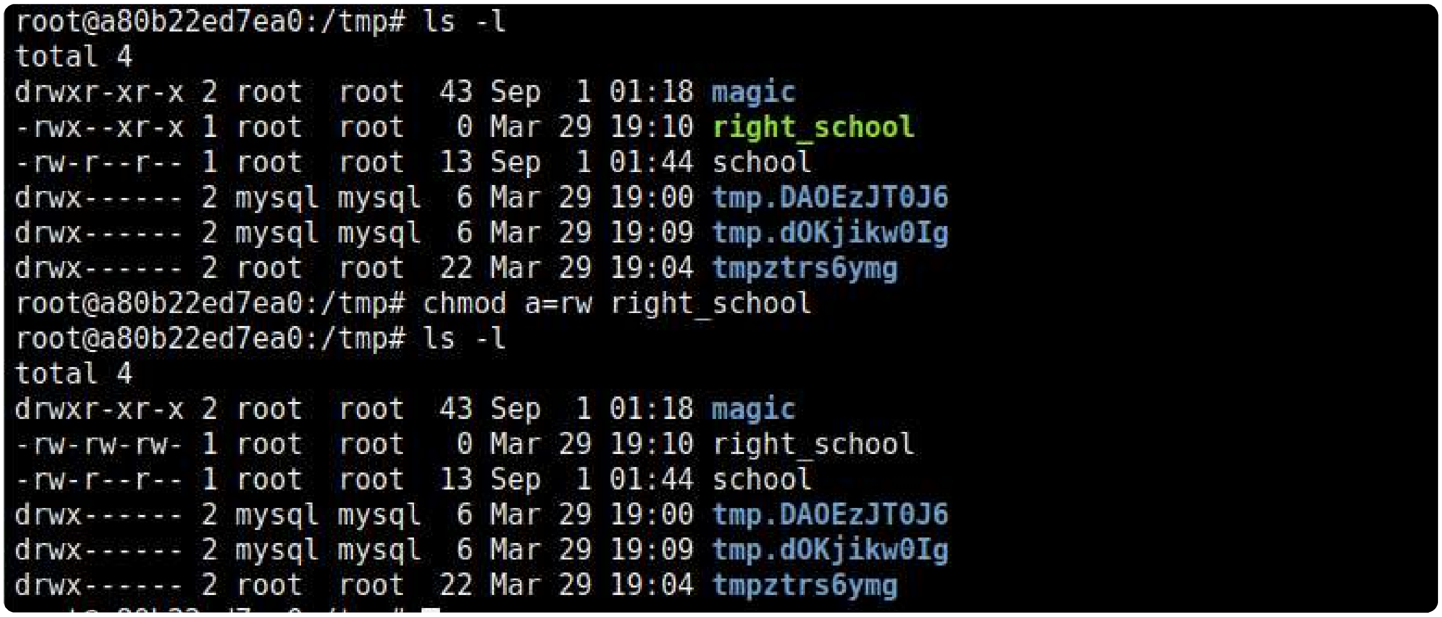


```
root@a80b22ed7ea0:/tmp# chmod g-r right_school
root@a80b22ed7ea0:/tmp# ls -l
total 4
drwxr-xr-x 2 root  root  43 Sep  1 01:18 magic
-rwx--xr-x 1 root  root   0 Mar 29 19:10 right_school
-rw-r--r-- 1 root  root  13 Sep  1 01:44 school
drwx------ 2 mysql mysql  6 Mar 29 19:00 tmp.DAOEzJT0J6
drwx------ 2 mysql mysql  6 Mar 29 19:09 tmp.dOKjikw0Ig
drwx------ 2 root  root  22 Mar 29 19:04 tmpztrs6ymg
```

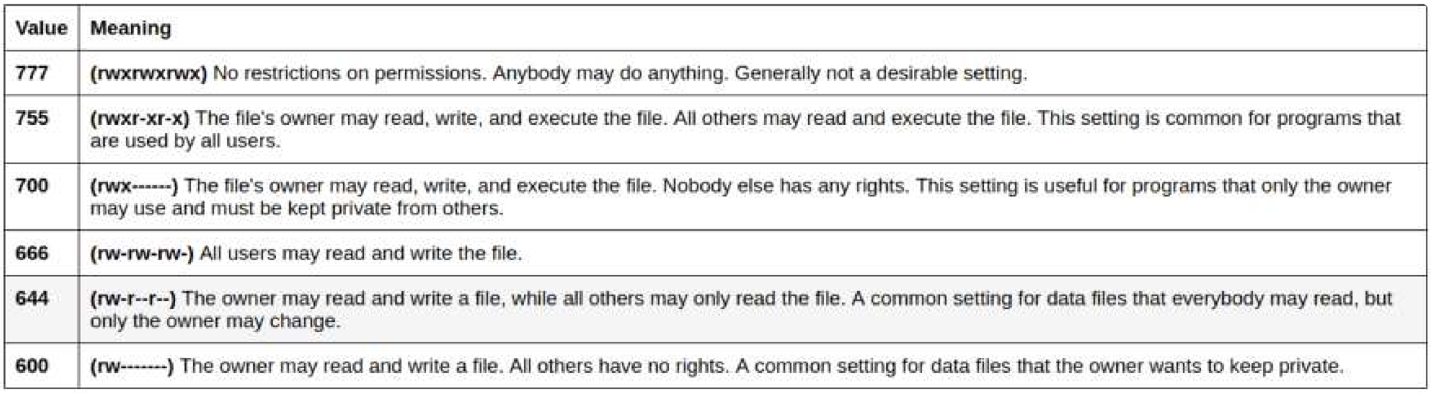
setting permission

to set a read and write permission to all use the operator =

```
chmod a=rw <file or dir>
```



```
root@a80b22ed7ea0:/tmp# ls -l
total 4
drwxr-xr-x 2 root  root  43 Sep  1 01:18 magic
-rwx--xr-x 1 root  root   0 Mar 29 19:10 right_school
-rw-r--r-- 1 root  root  13 Sep  1 01:44 school
drwx------ 2 mysql mysql  6 Mar 29 19:00 tmp.DAOEzJT0J6
drwx------ 2 mysql mysql  6 Mar 29 19:09 tmp.dOKjikw0Ig
drwx------ 2 root  root  22 Mar 29 19:04 tmpztrs6ymg
root@a80b22ed7ea0:/tmp# chmod a=rw right_school
root@a80b22ed7ea0:/tmp# ls -l
total 4
drwxr-xr-x 2 root  root  43 Sep  1 01:18 magic
-rw-rw-rw- 1 root  root   0 Mar 29 19:10 right_school
-rw-r--r-- 1 root  root  13 Sep  1 01:44 school
drwx------ 2 mysql mysql  6 Mar 29 19:00 tmp.DAOEzJT0J6
drwx------ 2 mysql mysql  6 Mar 29 19:09 tmp.dOKjikw0Ig
drwx------ 2 root  root  22 Mar 29 19:04 tmpztrs6ymg
```



Value	Meaning
777	**(rwxrwxrwx)** No restrictions on permissions. Anybody may do anything. Generally not a desirable setting.
755	**(rwxr-xr-x)** The file's owner may read, write, and execute the file. All others may read and execute the file. This setting is common for programs that are used by all users.
700	**(rwx------)** The file's owner may read, write, and execute the file. Nobody else has any rights. This setting is useful for programs that only the owner may use and must be kept private from others.
666	**(rw-rw-rw-)** All users may read and write the file.
644	**(rw-r--r--)** The owner may read and write a file, while all others may only read the file. A common setting for data files that everybody may read, but only the owner may change.
600	**(rw-------)** The owner may read and write a file. All others have no rights. A common setting for data files that the owner wants to keep private.

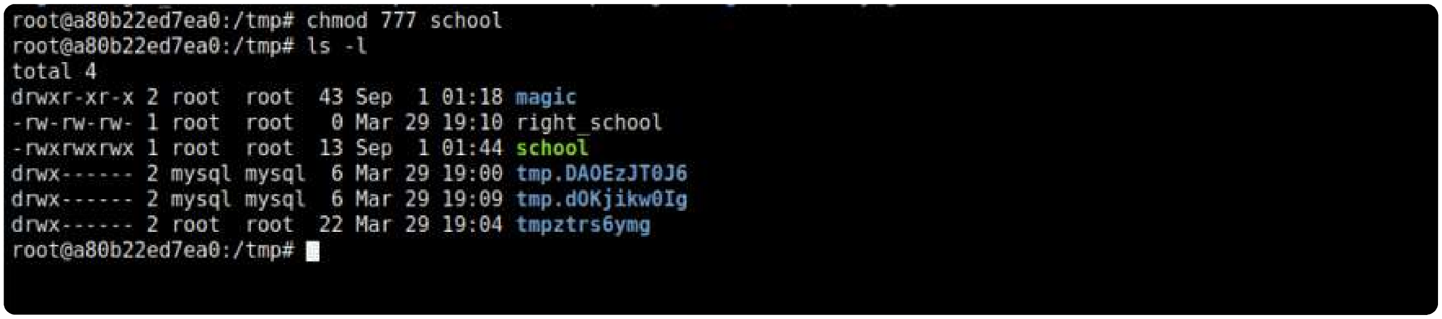
Now that we seen how permission work, we will see some commands and how to use them.

chmod

The first command we will see is chmod which is used to change the permissions of a file or directory.

To use the command:

```
chmod <PERMISSION_MOD> <FILE_OR_DIR>
```



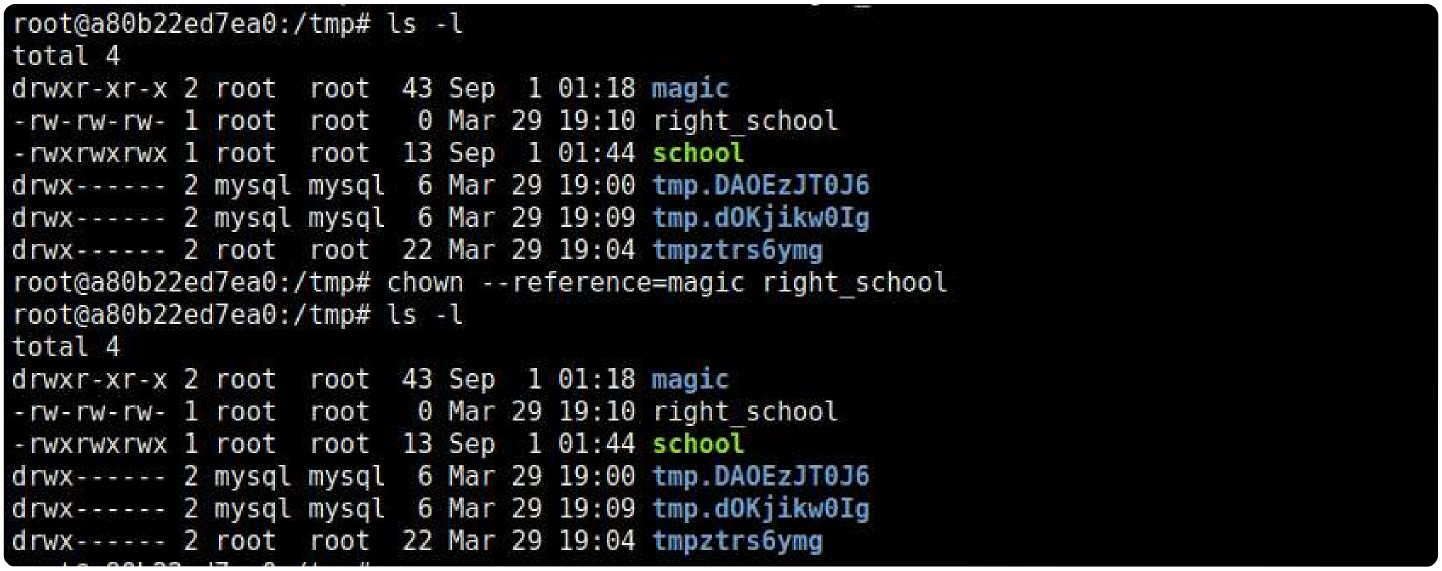
```
root@a80b22ed7ea0:/tmp# chmod 777 school
root@a80b22ed7ea0:/tmp# ls -l
total 4
drwxr-xr-x 2 root  root  43 Sep  1 01:18 magic
-rw-rw-rw- 1 root  root   0 Mar 29 19:10 right_school
-rwxrwxrwx 1 root  root  13 Sep  1 01:44 school
drwx------ 2 mysql mysql  6 Mar 29 19:00 tmp.DAOEzJT0J6
drwx------ 2 mysql mysql  6 Mar 29 19:09 tmp.dOKjikw0Ig
drwx------ 2 root  root  22 Mar 29 19:04 tmpztrs6ymg
root@a80b22ed7ea0:/tmp# 
```

Usually implemented options include:

- **-R** Recursive, i.e. include objects in subdirectories.
- **-v** verbose, show objects changed (unchanged objects are not shown).

Use --reference=REF_FILE to set the permission of the new file relative to the ref_file.

```
chown --reference=REF_FILE FILE
```



```
root@a80b22ed7ea0:/tmp# ls -l
total 4
drwxr-xr-x 2 root  root  43 Sep  1 01:18 magic
-rw-rw-rw- 1 root  root   0 Mar 29 19:10 right_school
-rwxrwxrwx 1 root  root  13 Sep  1 01:44 school
drwx------ 2 mysql mysql  6 Mar 29 19:00 tmp.DAOEzJT0J6
drwx------ 2 mysql mysql  6 Mar 29 19:09 tmp.dOKjikw0Ig
drwx------ 2 root  root  22 Mar 29 19:04 tmpztrs6ymg
root@a80b22ed7ea0:/tmp# chown --reference=magic right_school
root@a80b22ed7ea0:/tmp# ls -l
total 4
drwxr-xr-x 2 root  root  43 Sep  1 01:18 magic
-rw-rw-rw- 1 root  root   0 Mar 29 19:10 right_school
-rwxrwxrwx 1 root  root  13 Sep  1 01:44 school
drwx------ 2 mysql mysql  6 Mar 29 19:00 tmp.DAOEzJT0J6
drwx------ 2 mysql mysql  6 Mar 29 19:09 tmp.dOKjikw0Ig
drwx------ 2 root  root  22 Mar 29 19:04 tmpztrs6ymg
```

> Tip: for more info checkout chmod or use **man chmod**

---

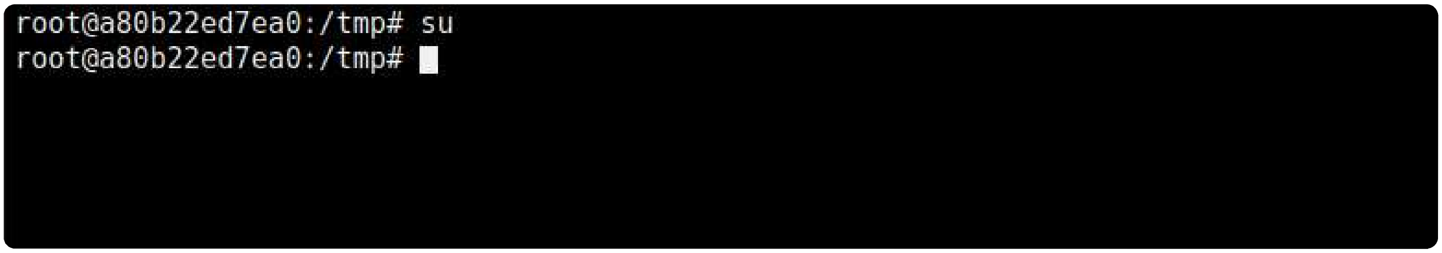
su

su is a program that can give you temporary access to the superuser's privileges.

To exit the superuser session, type exit and we will return to your previous session.

```
su
```

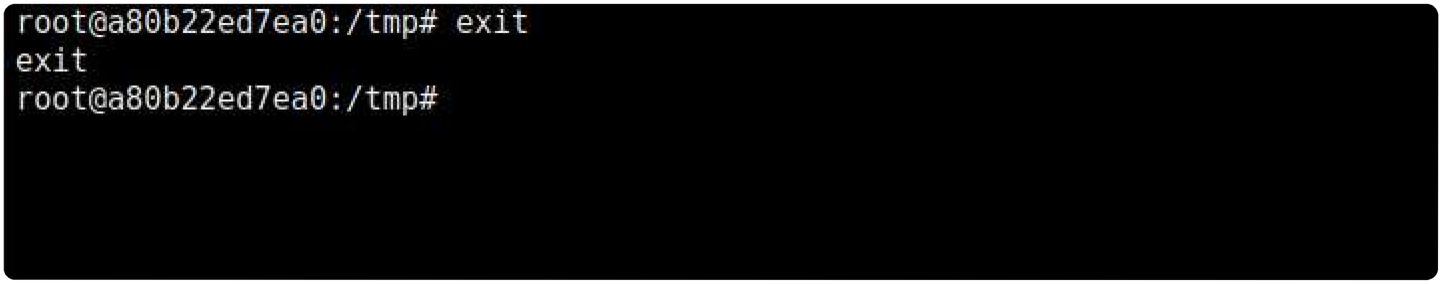
```
su <USER_NAME>
```



```
root@a80b22ed7ea0:/tmp# su
root@a80b22ed7ea0:/tmp# █
```

To exit a shell use exit command

```
exit
```

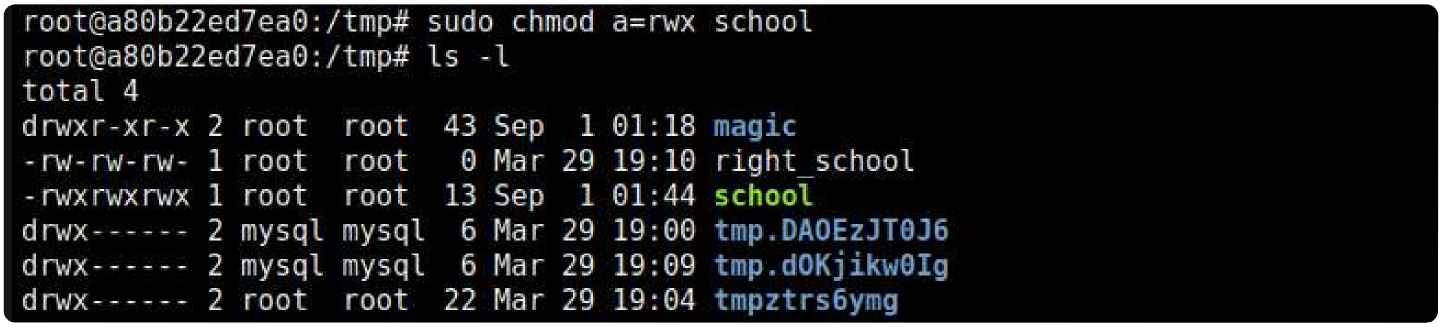


```
root@a80b22ed7ea0:/tmp# exit
exit
root@a80b22ed7ea0:/tmp#
```

sudo

sudo command is used to execute a command as the superuser, the desired command is simply preceded with the sudo command.

```
sudo <SOME_COMMAND>
```



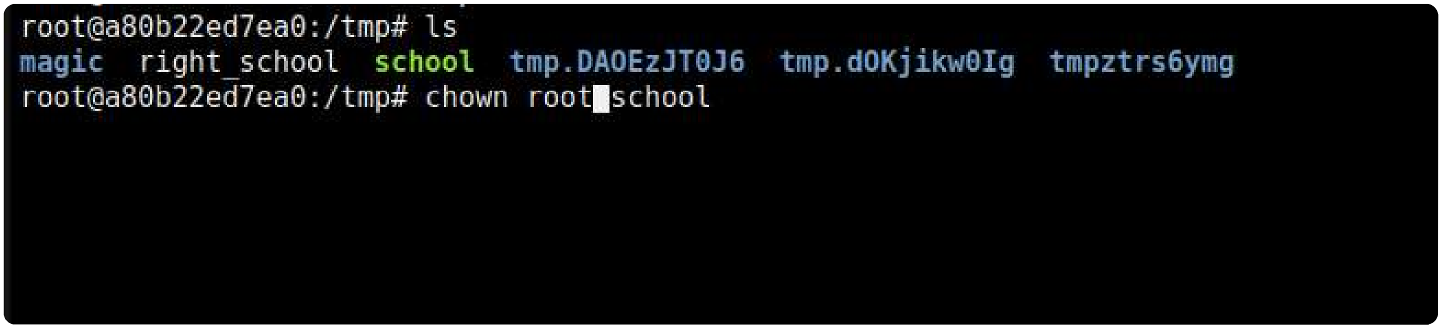
```
root@a80b22ed7ea0:/tmp# sudo chmod a=rwx school
root@a80b22ed7ea0:/tmp# ls -l
total 4
drwxr-xr-x 2 root  root  43 Sep  1 01:18 magic
-rw-rw-rw- 1 root  root   0 Mar 29 19:10 right_school
-rwxrwxrwx 1 root  root  13 Sep  1 01:44 school
drwx------ 2 mysql mysql  6 Mar 29 19:00 tmp.DAOEzJT0J6
drwx------ 2 mysql mysql  6 Mar 29 19:09 tmp.dOKjikw0Ig
drwx------ 2 root  root  22 Mar 29 19:04 tmpztrs6ymg
```

chown

We use the chown command to change the ownership of a file, like changing the owner of file1 form me to you.

```
chown [OPTIONS] USER[:GROUP] FILE(s)
```

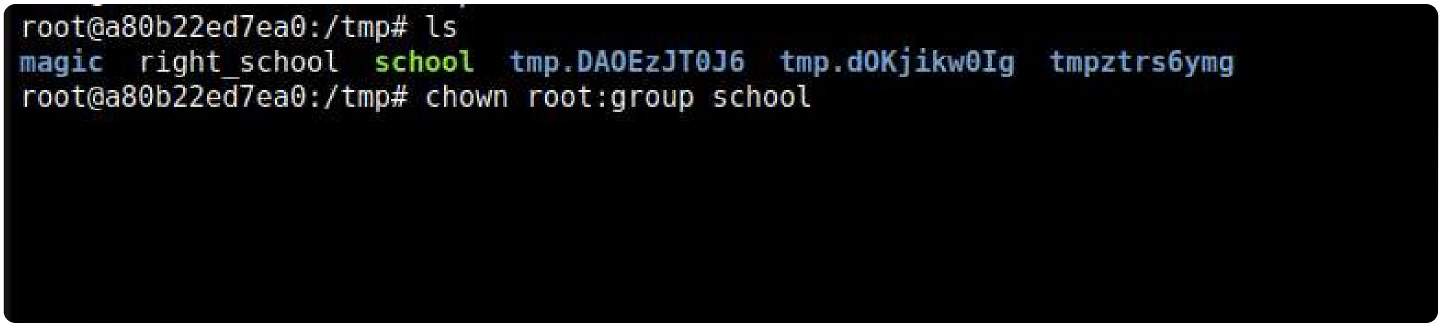
```
chown <CHANGED_OWNER> <FILE>
```



```
root@a80b22ed7ea0:/tmp# ls
magic  right_school  school  tmp.DAOEzJT0J6  tmp.dOKjikw0Ig  tmpztrs6ymg
root@a80b22ed7ea0:/tmp# chown root school
```

you can also change the owner and the group at the same time

```
chown USER:GROUP FILE
```



```
root@a80b22ed7ea0:/tmp# ls
magic  right_school  school  tmp.DAOEzJT0J6  tmp.dOKjikw0Ig  tmpztrs6ymg
root@a80b22ed7ea0:/tmp# chown root:group school
```

To recursively operate on all files and directories under the given directory, use the -R (--recursive) option.

```
chown -R USER:GROUP DIRECTORY
```

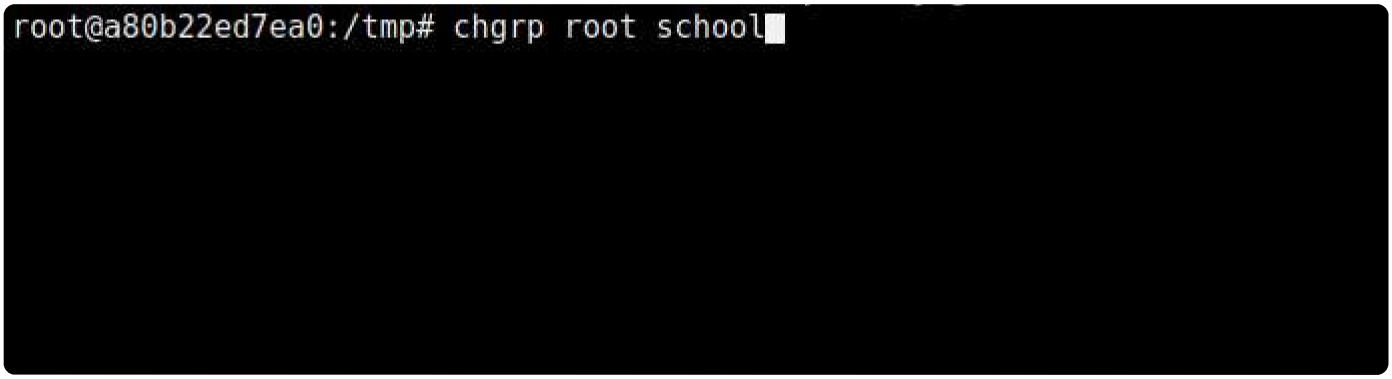
The --reference=ref_file option allows you to change the user and group ownership of given files to be same as those of the specified reference file (ref_file).

```
chown --reference=REF_FILE FILE
```

chgrp

We use the chgrp command to change the group ownership of a file or directory.

```
chgrp <NEW_GROUP> <FILE>
```



```
root@a80b22ed7ea0:/tmp# chgrp root school
```

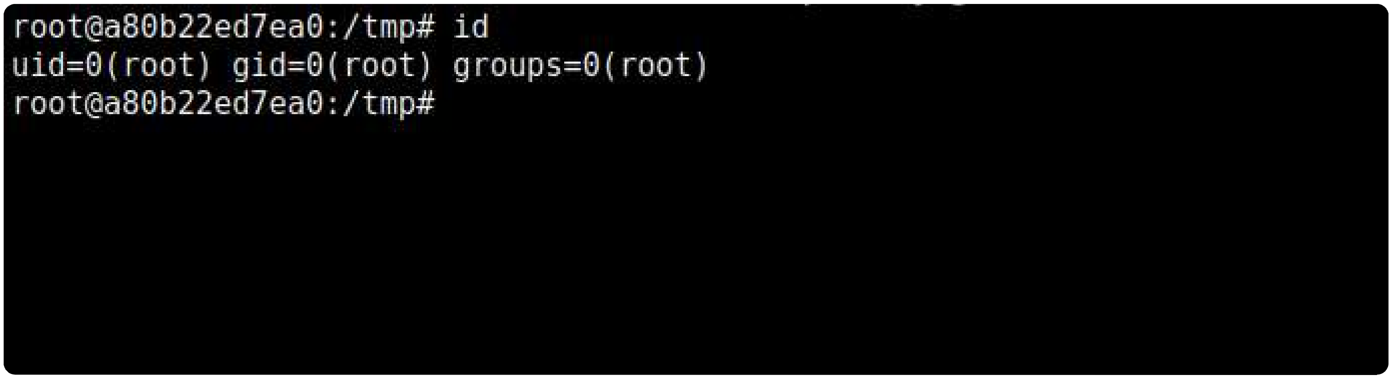
> Tip: for more info use the man page

---

id

We use the id command to print the user and group name and ID of the current user or any other user in the server.

```
id [OPTION]… [USER]
```



```
root@a80b22ed7ea0:/tmp# id
uid=0(root) gid=0(root) groups=0(root)
root@a80b22ed7ea0:/tmp#
```

> use the man page for more info on options

---

groups

We use the groups command to prints the names of the primary and any supplementary groups for each given username and manage users with the same security and access privileges.

```
groups [username]...
```

Image description

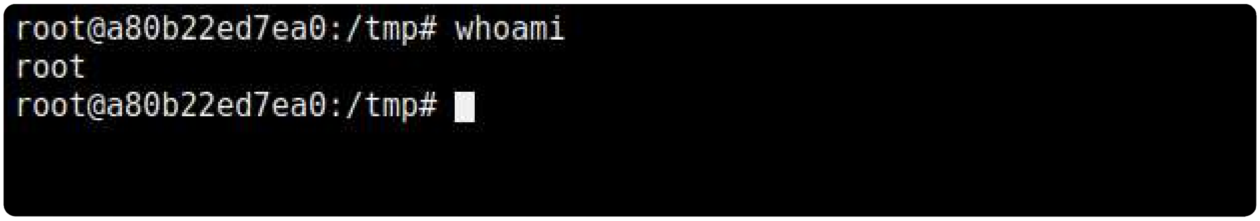
> use the man page for more info

---

whoami

we use the whoami to displays user, group and privileges information for the user who is currently logged on to the local system.

```
whoami
```



```
root@a80b22ed7ea0:/tmp# whoami
root
root@a80b22ed7ea0:/tmp# 
```

> use the man page for more info

---

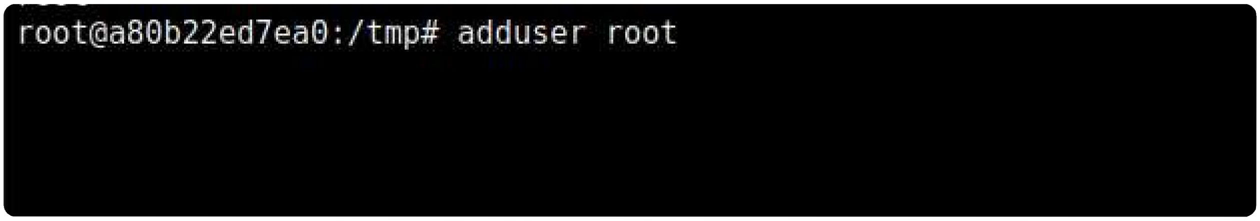
adduser

we use the adduser to add a new user to your current Linux machine.

But you need to install adduser using the command

```
sudo apt-get install adduser
```

```
adduser <username>
```



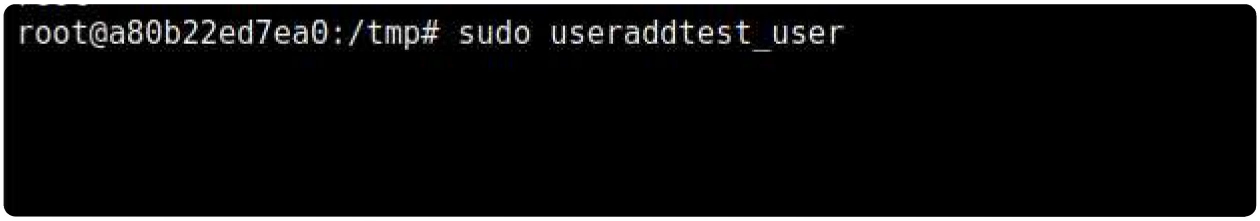
```
root@a80b22ed7ea0:/tmp# adduser root
```

> for more info read adduser

useradd

we use the useradd to add user accounts to your system.

```
useradd [options] name_of_the_user
```



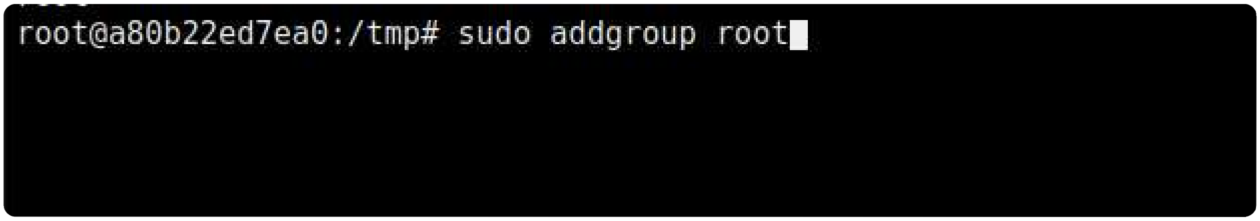
```
root@a80b22ed7ea0:/tmp# sudo useraddtest_user
```

> for more info read useradd

addgroup

we use the addgroup to add a new group to your current Linux machine.

```
sudo addgroup <groupname>
```



```
root@a80b22ed7ea0:/tmp# sudo addgroup root
```

> for more info read addgroup

Top comments (0)

Code of Conduct • Report abuse

Haile Melaku

A 4th year cse student at Adama science and technology also a full stack software engineer

LOCATION
Ethiopia

EDUCATION
Adama science and technology

JOINED
Apr 10, 2021

More from Haile Melaku

C - Variables, if, else, while
#c #programming #devops #linux

Shell, init files, variables and expansions
#variables #linux #shell #bash

Shell, I/O Redirections and filters
#linux #ubuntu #pips #shell