Permissions

. Act to allow someone to do something.

. In Linux, permissions control who can access and modify files and directories.

. Permission check command = ll & ls –ld

Tyes of permissions

1. Read (syntax=r)
2. Write (syntax=w)
3. Excute(syntax=x)

Commands

chmod

Syntax= chmod ugo (+-) rwx file/dir. Name (alphabetic method)

Example = chmod u+r file/dir. Name

chmod g+w admin (puting write permission on group)

chmod o+r admin (puting write permission on other)

chmod g+w-r admin

chmod u+r,g-w,o+r admin (appling multiple permissions at once)

u	*owner permission*
g	*groupowner*
o	*other permission*

Numeric method

r	*4*
w	*2*
x	*1*

rw	*4+2=6*
wx	*2+1=3*
xr	*1+4=5*

Syntax: chmod 777 dir./file name

. Full permission of the directory = 7

. Full permission of the file = 6

Example = chmod 635 admin

Change owner & group owner

Commands

. chown new owner name file/dir. Name (owner change)

Example = chown alexa admin

Change group owner

chgrp new groupname file/dir. Name

Example = chgrp harry manu

Change both at the same time:

Syntax: chown ownername:grupowner file/dir.

Example = chown alexa:harrry admin

Umask

In Linux, umask is a command and a variable that controls the default file permissions for new files and directories . When you create a file or directory, the permissions set by the umask are applied by default.

Umask (0022) by defaults

Set umask command= umask 025 (Temporary)

Setting a Parmanet Umask for a Particular user:

open .bashrc file and enter the umask

Setting the umask for every user: open vim /etc/login.defs file and enter the umask

Special Permission

1. Acl
2. Suid
3. Sgid
4. Stickybit

Acl

ACL (Access Control List) in Linux is an extended permission system that allows defining more precise and flexible permissions for specific users or groups, in addition to traditional permissions (such as user, group, and others) . It provides a more granular method for controlling access to files and directories.

Commands

1.Syntax: setfacl optons u:username:permission file/dir name (for user)

Example: setfacl –m u:admin:rwx manu

2. setfacl –m g:harry:rw alexa (for group)

3.getfacl filename = file information

Example = getfacl mannu

4. setfacl –x g:groupname file/dir. Name (deleting a group or user you added)

Example = setfacl -x g:harry alexa

setfacl -x u:admin mannu

5. setfacl -b file/dir. Name (it will delete all the users and groups added)

Example = setfacl -b mannu

(note: + sign will apper in the permission)

Suid

When the SUID bit is set on an executable file, that file runs with the owner's privileges, no matter who executes it

Syntax: chmod u+s filename(add suid)

Chmod u-s filename(remove suid)

(note: 's' sign will apper in the owner permission)

Sgid

1. SGID on Executable Files

- **When you set the SGID bit on an executable, anyone running it temporarily assumes the permissions of the file's group, instead of their own group.**

2. SGID on Directories

- **When applied to a directory, new files and subdirectories inherit its group ownership, regardless of the creator’s primary group.**

Synatx: chmod g+s file/dir. Name

Example: chmod g+s manu

(note: ‘s’ sign will apper in the group permission)

Stickybit

Stop unwanted deletion

No one other than root, owner, director owner can delete the files created in dirctory.

Synatx: chmod o+t dir. Name

Example = chmod o+t sinu

(note: ‘t’ sign will apper in the permission)

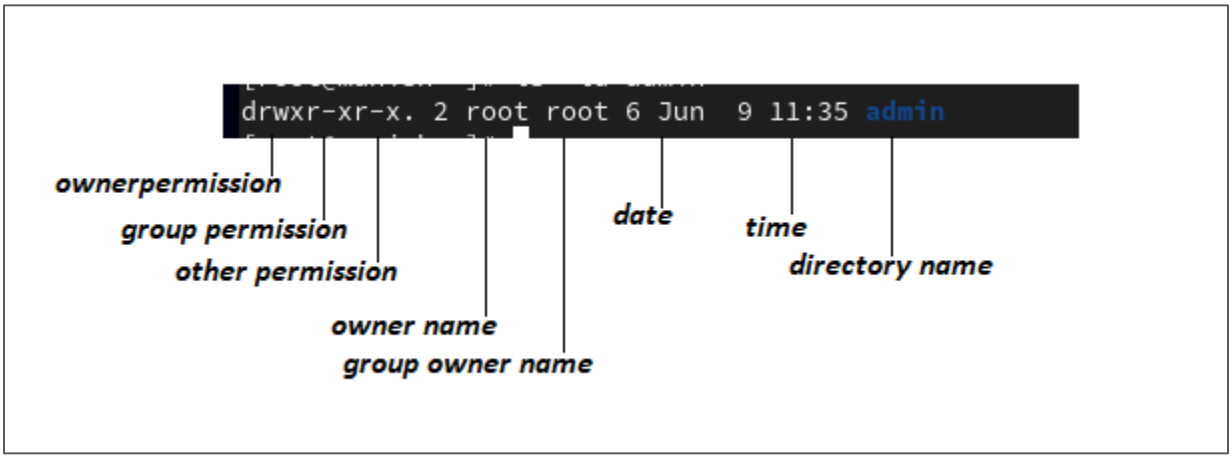
	File	Directory
read (r)	*can be ready*	*can be listed*
weite (w)	*can be changed*	*can be changed/copy/delete*
execute(x)	*can be executed*	*have acces to contents*

ll Command

```
[root@manish ~]# ll
total 31212
-rw-r--r--. 1 root root      298 Feb  1 19:45 '!'
drwxr-xr-x. 2 root root        6 Feb 13 10:56  kajl
dr-xrw-r--+ 2 root root        6 Jun  9 07:22  kalo
-rw-r--r--. 1 root root 31948800 Feb  1 19:30  komal
-rw-r--r--. 1 root root      112 Jun  5 15:49  lines
drwxr-xr-x. 3 root root       31 Feb 18 09:41  manish
drwxr-xr-x. 2 root root       27 Feb  3 15:14  msd
dr-x-w---x. 2 root root        6 Jun  9 07:41  o
-rw--w-r--. 1 root root        0 Jun  9 07:28  pihu
drwxr-xr-x. 3 root root       17 Feb 13 15:44  root@192.168.102.130
drwxr-xr-x. 2 root root       16 Feb  1 18:56  sita
-rw-r--r--. 1 root root       31 Dec  7  2024  vegetarian
[root@manish ~]#
```

ls –ld(view permission for only one dir. Or file)

```
[root@manish ~]# ls -ld msd
drwxr-xr-x. 2 root root 27 Feb  3 15:14 msd
[root@manish ~]#
```



chmod Command

```
[root@manish ~]# ls -ld admin
drwxr-xr-x. 2 root root 6 Jun  9 11:35 admin
[root@manish ~]# chmod u-xr admin
[root@manish ~]# ls -ld admin
d-w-r-xr-x. 2 root root 6 Jun  9 11:35 admin
[root@manish ~]# chmod g+w admin
[root@manish ~]# ls -ld admin
d-w-rwxr-x. 2 root root 6 Jun  9 11:35 admin
[root@manish ~]# chmod o-x admin
[root@manish ~]# ls -ld admin
d-w-rwxr--. 2 root root 6 Jun  9 11:35 admin
[root@manish ~]# chmod u+r,g-w,o+x admin
[root@manish ~]# ls -ld admin
drw-r-xr-x. 2 root root 6 Jun  9 11:35 admin
[root@manish ~]#
```

chmod command (numeric method)

```
[root@manish ~]# touch alexa
[root@manish ~]# ls -ld alexa
-rw-r--r--. 1 root root 0 Jun  9 11:52 alexa
[root@manish ~]# chmod 531 alexa
[root@manish ~]# ls -ld alexa
-r-x-wx--x. 1 root root 0 Jun  9 11:52 alexa
[root@manish ~]# chmod 666 alexa
[root@manish ~]# ls -ld alexa
-rw-rw-rw-. 1 root root 0 Jun  9 11:52 alexa
[root@manish ~]# chmod 444  alexa
[root@manish ~]# ls -ld alexa
-r--r--r--. 1 root root 0 Jun  9 11:52 alexa
[root@manish ~]# chmod 222   alexa
[root@manish ~]# ls -ld alexa
--w--w--w-. 1 root root 0 Jun  9 11:52 alexa
[root@manish ~]# chmod 424   alexa
[root@manish ~]# ls -ld alexa
-r---w-r--. 1 root root 0 Jun  9 11:52 alexa
[root@manish ~]#
```

Change Owenr & group Owner (chown & chgrp command)

```
[root@manish ~]# ls -ld alexa
-r---w-r--. 1 root root 0 Jun  9 11:52 alexa
[root@manish ~]# chown manu alexa
[root@manish ~]# ls -ld alexa
-r---w-r--. 1 manu root 0 Jun  9 11:52 alexa
[root@manish ~]# chgrp zip alexa
[root@manish ~]# ls -ld alexa
-r---w-r--. 1 manu zip 0 Jun  9 11:52 alexa
[root@manish ~]# chown root:root alexa
[root@manish ~]# ls -ld alexa
-r---w-r--. 1 root root 0 Jun  9 11:52 alexa
[root@manish ~]#
```

Umask

```
[root@manish ~]# umask
0022
[root@manish ~]#
```

```
[root@manish ~]# umask
0022
[root@manish ~]# umask 644
[root@manish ~]# umask
0644
[root@manish ~]# bash
Hello, you are logged in as root
[root@manish ~]# umask
0022
[root@manish ~]#
```

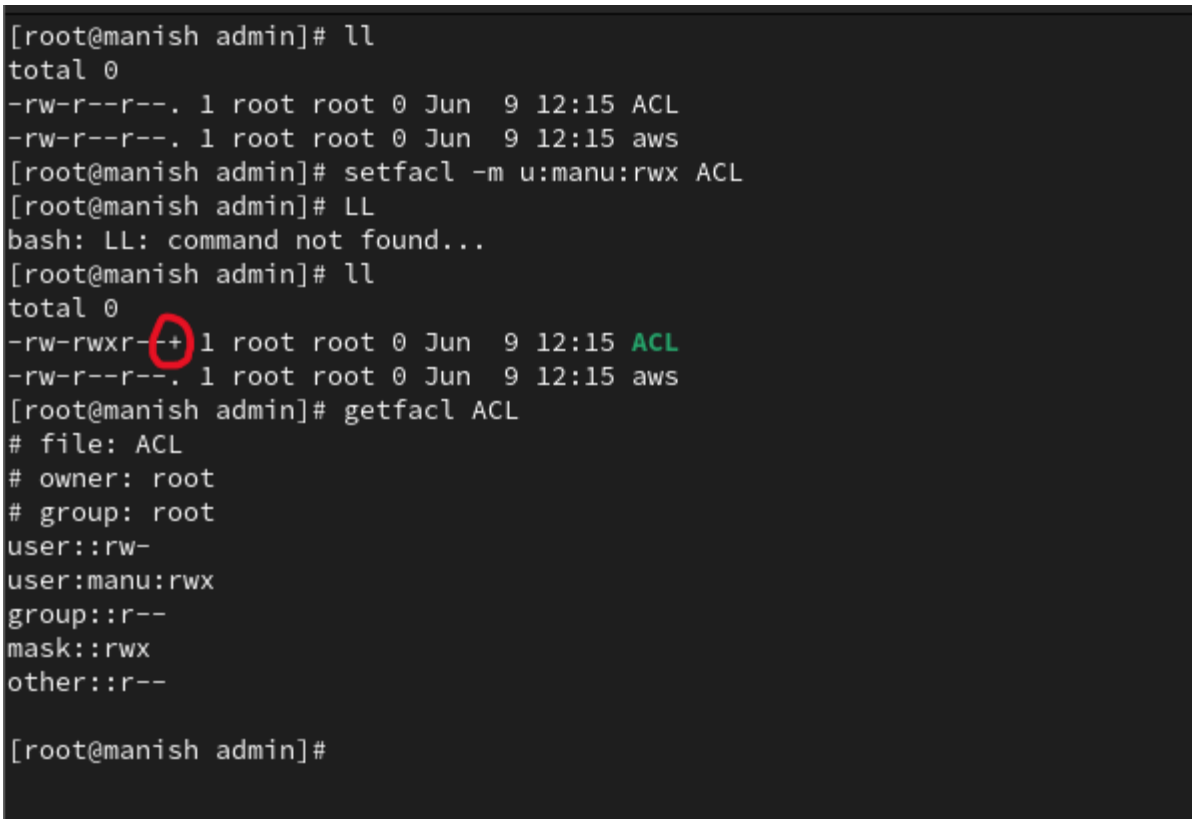
vim .bashrc (for a particular user)

```
alias mv= mv -i

echo "Hello, you are logged in as $(whoami)"
export komal=manish
export mannu="manish komal gudddu"
umask    0022
~
~
~
".bashrc" 27L, 544B
```

vim /etc/login.defs (set for every users)

```
# Currently ERASECHAR, KILLCHAR and ULIMIT are not supported

# Default initial "umask" value used by login(1) on non-PAM enabled systems.
# Default "umask" value for pam_umask(8) on PAM enabled systems.
# UMASK is also used by useradd(8) and newusers(8) to set the mode for new
# home directories if HOME_MODE is not set.
# 022 is the default value, but 027, or even 077, could be considered
# for increased privacy. There is no One True Answer here: each sysadmin
# must make up their mind.
UMASK           0022
```

ACL (setfacl command) (for a user)



setfacl (for a group)

```
[root@manish admin]# ll
total 0
-rw-rwxr--+ 1 root root 0 Jun  9 12:15 ACL
-rw-r--r--. 1 root root 0 Jun  9 12:15 aws
[root@manish admin]# setfacl -m g:zip:rwr  aws
setfacl: Option -m: Invalid argument near character 9
[root@manish admin]# setfacl -m g:zip:rwx   aws
[root@manish admin]# getfacl aws
# file: aws
# owner: root
# group: root
user::rw-
group::r--
group:zip:rwx
mask::rwx
other::r--

[root@manish admin]# ll
total 0
-rw-rwxr--+ 1 root root 0 Jun  9 12:15 ACL
-rw-rwxr--+ 1 root root 0 Jun  9 12:15 aws
[root@manish admin]#
```

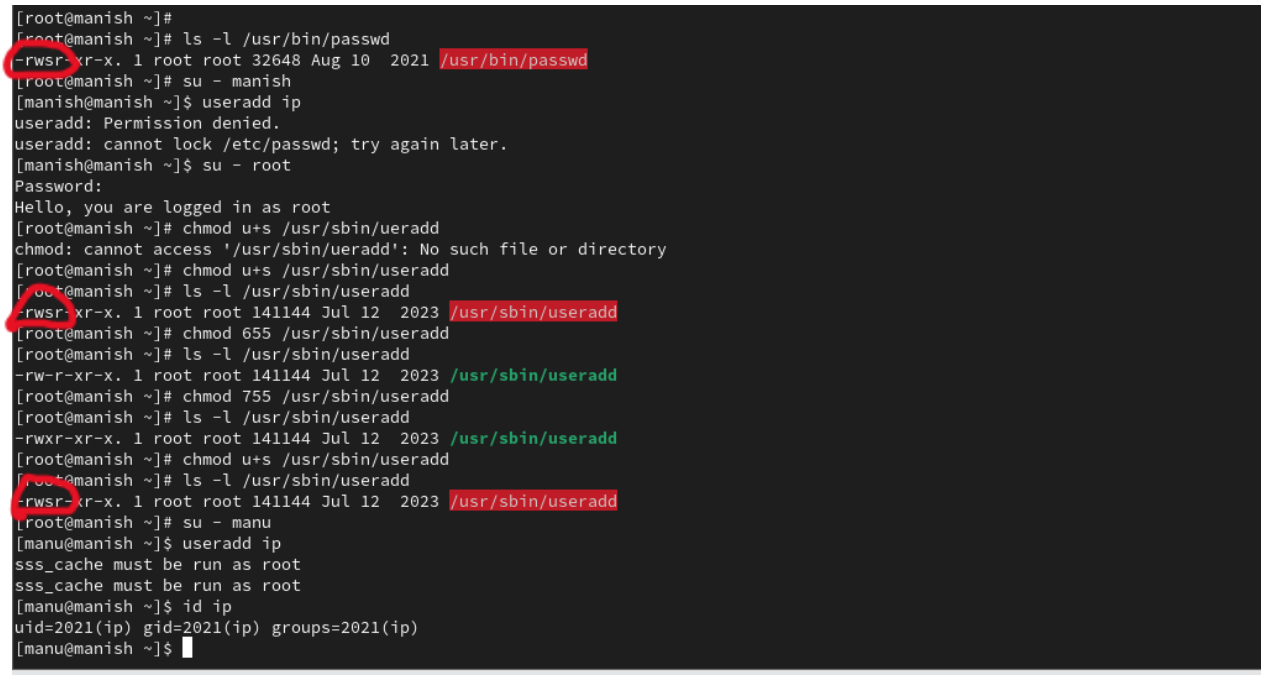
setfacl -x (deleting group and user)

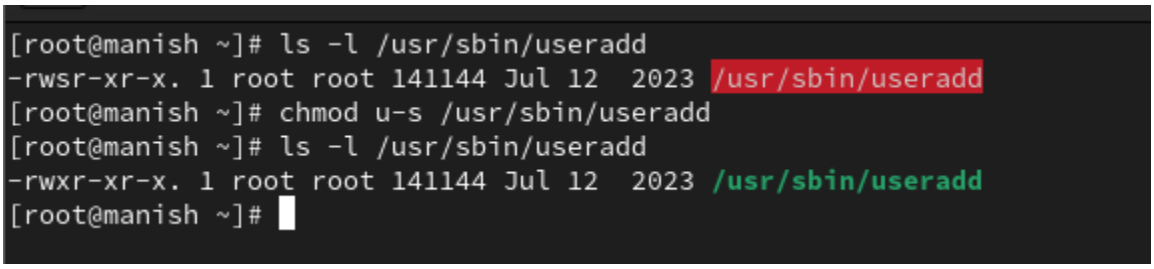
```
[root@manish admin]# ll
total 0
-rw-rwxr--+ 1 root root 0 Jun  9 12:15 ACL
-rw-rwxr--+ 1 root root 0 Jun  9 12:15 aws
[root@manish admin]# setfacl -x g:zip aws
[root@manish admin]# getfacl aws
# file: aws
# owner: root
# group: root
user::rw-
group::r--
mask::r--
other::r--

[root@manish admin]# setfacl -x u:manu ACL
[root@manish admin]# getfacl ACL
# file: ACL
# owner: root
# group: root
user::rw-
group::r--
mask::r--
other::r--

[root@manish admin]#
```

setfacl -b (all deleting group & user)

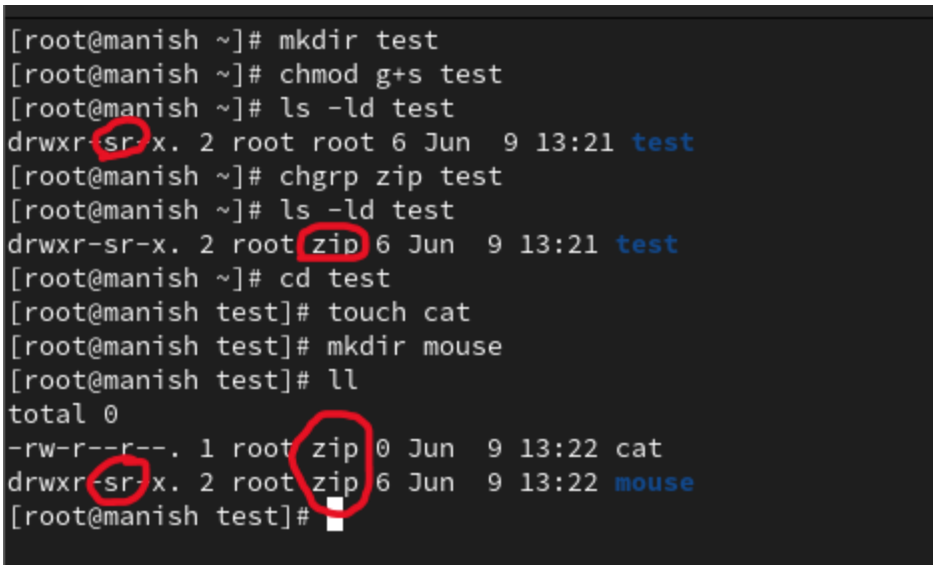
```
[root@manish ~]# ls -ld kalo
dr-xrw-r--+ 2 root root 6 Jun  9 07:22 kalo
[root@manish ~]# setfacl -m u:manu:wrx kalo
[root@manish ~]# setfacl -m g:zip:wrx kalo
[root@manish ~]# getfacl kalo
# file: kalo
# owner: root
# group: root
user::r-x
user:manu:rwx
group::-w-
group:zip:rwx
mask::rwx
other::r--

[root@manish ~]# setfacl -b kalo
[root@manish ~]# getfacl kalo
# file: kalo
# owner: root
# group: root
user::r-x
group::-w-
other::r--

[root@manish ~]#
```

Suid (will only be applied to executable files)


[root@manish ~]#
[root@manish ~]# ls -l /usr/bin/passwd
-rwsr-xr-x. 1 root root 32648 Aug 10 2021 /usr/bin/passwd
[root@manish ~]# su - manish
[manish@manish ~]$ useradd ip
useradd: Permission denied.
useradd: cannot lock /etc/passwd; try again later.
[manish@manish ~]$ su - root
Password:
Hello, you are logged in as root
[root@manish ~]# chmod u+s /usr/sbin/ueradd
chmod: cannot access '/usr/sbin/ueradd': No such file or directory
[root@manish ~]# chmod u+s /usr/sbin/useradd
[root@manish ~]# ls -l /usr/sbin/useradd
-rwsr-xr-x. 1 root root 141144 Jul 12 2023 /usr/sbin/useradd
[root@manish ~]# chmod 655 /usr/sbin/useradd
[root@manish ~]# ls -l /usr/sbin/useradd
-rw-r-xr-x. 1 root root 141144 Jul 12 2023 /usr/sbin/useradd
[root@manish ~]# chmod 755 /usr/sbin/useradd
[root@manish ~]# ls -l /usr/sbin/useradd
-rwxr-xr-x. 1 root root 141144 Jul 12 2023 /usr/sbin/useradd
[root@manish ~]# chmod u+s /usr/sbin/useradd
[root@manish ~]# ls -l /usr/sbin/useradd
-rwsr-xr-x. 1 root root 141144 Jul 12 2023 /usr/sbin/useradd
[root@manish ~]# su - manu
[manu@manish ~]$ useradd ip
sss_cache must be run as root
sss_cache must be run as root
[manu@manish ~]$ id ip
uid=2021(ip) gid=2021(ip) groups=2021(ip)
[manu@manish ~]$


[root@manish ~]# ls -l /usr/sbin/useradd
-rwsr-xr-x. 1 root root 141144 Jul 12 2023 /usr/sbin/useradd
[root@manish ~]# chmod u-s /usr/sbin/useradd
[root@manish ~]# ls -l /usr/sbin/useradd
-rwxr-xr-x. 1 root root 141144 Jul 12 2023 /usr/sbin/useradd
[root@manish ~]#

Sgid


[root@manish ~]# mkdir test
[root@manish ~]# chmod g+s test
[root@manish ~]# ls -ld test
drwxr-sr-x. 2 root root 6 Jun 9 13:21 test
[root@manish ~]# chgrp zip test
[root@manish ~]# ls -ld test
drwxr-sr-x. 2 root zip 6 Jun 9 13:21 test
[root@manish ~]# cd test
[root@manish test]# touch cat
[root@manish test]# mkdir mouse
[root@manish test]# ll
total 0
-rw-r--r--. 1 root zip 0 Jun 9 13:22 cat
drwxr-sr-x. 2 root zip 6 Jun 9 13:22 mouse
[root@manish test]#

Sticky bit

```
[root@manish ~]# mkdir /sinu
[root@manish ~]# chmod o+t /sinu/
[root@manish ~]# cd /
[root@manish /]# ls -ld sinu
drwxr-xr-t. 2 root root 6 Jun  9 13:45 sinu
[root@manish /]# chmod 777 sinu
[root@manish /]# chmod o+t /sinu/
[root@manish /]# ls -ld sinu
drwxrwxrwt. 2 root root 6 Jun  9 13:45 sinu
[root@manish /]# su - manu
[manu@manish ~]$ cd sinu
-bash: cd: sinu: No such file or directory
[manu@manish ~]$ cd /sinu
[manu@manish sinu]$ touch sinu
[manu@manish sinu]$ su - root
Password:
Hello, you are logged in as root
[root@manish ~]# useradd minku
[root@manish ~]# su - minku
[minku@manish ~]$ touch /sinu/manu
[minku@manish ~]$ rm -rvf /sinu/sinu
rm: cannot remove '/sinu/sinu': Operation not permitted
[minku@manish ~]$ rm -rvf /sinu/manu
removed '/sinu/manu'
[minku@manish ~]$
```

Complete

Permissions

THANK YOU