Assignment no.1

1. Go to the ssh folder

```
root@localhost:/etc/ssh
[root@localhost ~]# cd /etc/ssh
[root@localhost ssh]#
```

2. After this command, this file will open we had to edit it anyway

```
[root@localhost ssh]# vi /etc/ssh/sshd_config
```

```
root@localhost:/etc/ssh — /usr/bin/vim /etc/ssh/sshd_config

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options override the
# default value.

# To modify the system-wide sshd configuration, create a  *.conf  file under
#  /etc/ssh/sshd_config.d/  which will be automatically included below
Include /etc/ssh/sshd_config.d/*.conf

# If you want to change the port on a SELinux system, you have to tell
# SELinux about this change.
# semanage port -a -t ssh_port_t -p tcp #PORTNUMBER
#
Port 444
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

"/etc/ssh/sshd_config" 130L, 3668B                    23,1          5%
```

3. Use this command :set number

```
root@localhost:/etc/ssh — /usr/bin/vim /etc/ssh/sshd_config

 7
 8 # The strategy used for options in the default sshd_config shipped with
 9 # OpenSSH is to specify options with their default value where
10 # possible, but leave them commented.  Uncommented options override the
11 # default value.
12
13 # To modify the system-wide sshd configuration, create a  *.conf  file under
14 #  /etc/ssh/sshd_config.d/  which will be automatically included below
15 Include /etc/ssh/sshd_config.d/*.conf
16
17 # If you want to change the port on a SELinux system, you have to tell
18 # SELinux about this change.
19 # semanage port -a -t ssh_port_t -p tcp #PORTNUMBER
20 #
21 Port 444
22 #AddressFamily any
23 #ListenAddress 0.0.0.0
24 #ListenAddress ::
25
26 #HostKey /etc/ssh/ssh_host_rsa_key
27 #HostKey /etc/ssh/ssh_host_ecdsa_key
28 #HostKey /etc/ssh/ssh_host_ed25519_key
29
:set number                                                    23,1          5%
```

5.Add this line on 23 number

```
19 # semanage port -a -t ssh_port_t -p tcp #PORTNUMBER
20 #
21 Port 444
22 #AddressFamily any
23 AddressFamily inet
24 #ListenAddress 0.0.0.0
25 #ListenAddress ::
```

6. Add this in 41 line

```
37
38 # Authentication:
39
40 #LoginGraceTime 2m
41 PermitRootLogin yes
42 #PermitRootLogin prohibit-password
43 #StrictModes yes
44 #MaxAuthTries 6
45 #MaxSessions 10
46
47 #PubkeyAuthentication yes
48
```

7. Add this line in 104

```
100 #AllowAgentForwarding yes
101 #AllowTcpForwarding yes
102 #GatewayPorts no
103 #X11Forwarding no
104 X11Forwarding yes
105 #X11DisplayOffset 10
106 #X11UseLocalhost yes
107 #PermitTTY yes
108 #PrintMotd yes
109 #PrintLastLog yes
110 #TCPKeepAlive yes
111 #PermitUserEnvironment no
```

8. Add this :wq! to save the file

```
101 #AllowTcpForwarding yes
102 #GatewayPorts no
103 #X11Forwarding no
104 X11Forwarding yes
105 #X11DisplayOffset 10
106 #X11UseLocalhost yes
107 #PermitTTY yes
108 #PrintMotd yes
109 #PrintLastLog yes
110 #TCPKeepAlive yes
111 #PermitUserEnvironment no
112 #Compression delayed
113 #ClientAliveInterval 0
114 #ClientAliveCountMax 3
115 #UseDNS no
:wq!
```

9. Add this line in 23 and save it

```
12 # Any configuration value is only changed the first time it is set.
13 # Thus, host-specific definitions should be at the beginning of the
14 # configuration file, and defaults at the end.
15
16 # Site-wide defaults for some commonly used options.  For a comprehensive
17 # list of available options, their meanings and defaults, please see the
18 # ssh_config(5) man page.
19
20 # Host *
21 #   ForwardAgent no
22 #   ForwardX11 no
23 Forward11 yes
24 #   PasswordAuthentication yes
25 #   HostbasedAuthentication no
26 #   GSSAPIAuthentication no
27 #   GSSAPIDelegateCredentials no
28 #   GSSAPIKeyExchange no
:wq!
```

10. After writing this command you can see it is already enabled and success

```
[root@localhost ssh]# cd
[root@localhost ~]#
[root@localhost ~]# firewall-cmd --permanent --zone=public --add-service=ssh
Warning: ALREADY_ENABLED: ssh
success
[root@localhost ~]#
```

11. After this command we see the success means it has been done

```
[root@localhost ~]# firewall-cmd --reload
success
```

12. List is also active and success

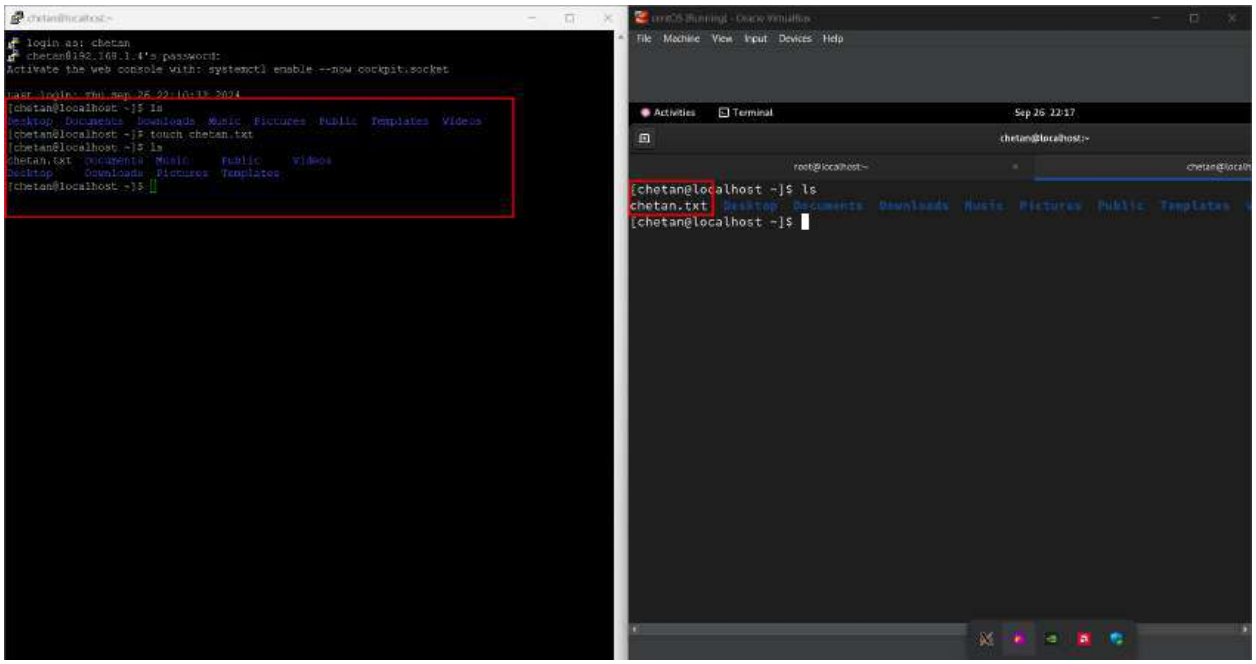
```
[root@localhost ~]# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp0s3
  sources:
  services: cockpit dhcpv6-client ssh
  ports:
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
[root@localhost ~]#
```

13. We can see the active status here

```
[root@localhost ~]# systemctl restart sshd
[root@localhost ~]# systemctl status sshd
● sshd.service - OpenSSH server daemon
     Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; preset: enabled)
     Active: active (running) since Thu 2024-09-26 22:14:40 IST; 7s ago
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 5516 (sshd)
      Tasks: 1 (limit: 23020)
     Memory: 1.4M
        CPU: 21ms
     CGroup: /system.slice/sshd.service
             └─5516 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Sep 26 22:14:40 localhost.localdomain systemd[1]: Starting OpenSSH server daemon...
Sep 26 22:14:40 localhost.localdomain sshd[5516]: Server listening on 0.0.0.0 port 22.
Sep 26 22:14:40 localhost.localdomain systemd[1]: Started OpenSSH server daemon.
[root@localhost ~]#
```

14. After downloading the Xming Xserver and putty
In putty i had connected the ip address of centos and you can it get access properly and created the file there and it is showing in centos so in these we can access the remote system



2. Create a file on your Linux system and try to copy from your system to neighbors Linux system using

a. Destination folder is /home

b. Destination folder is /root

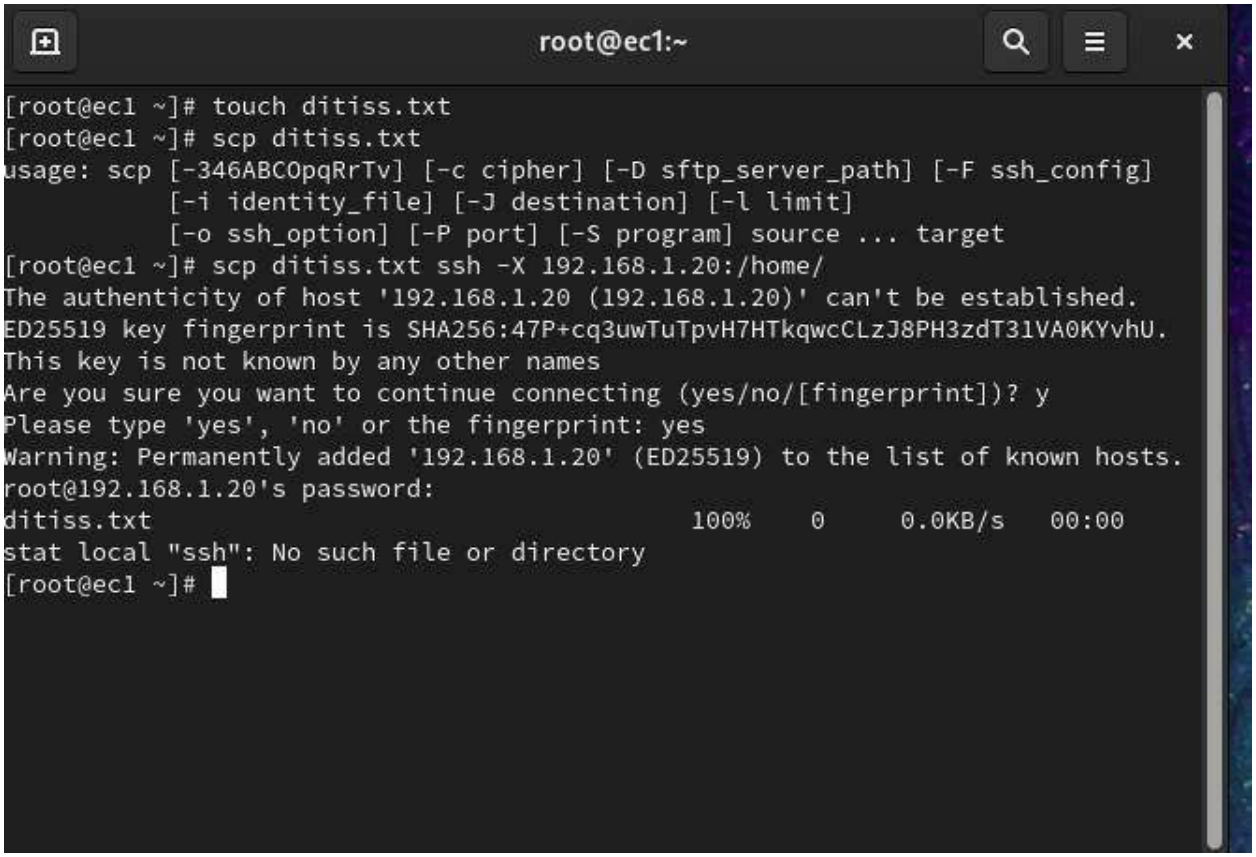
a.

1.Now create empty file touch ditiss.txt in server side then copy this file in client side

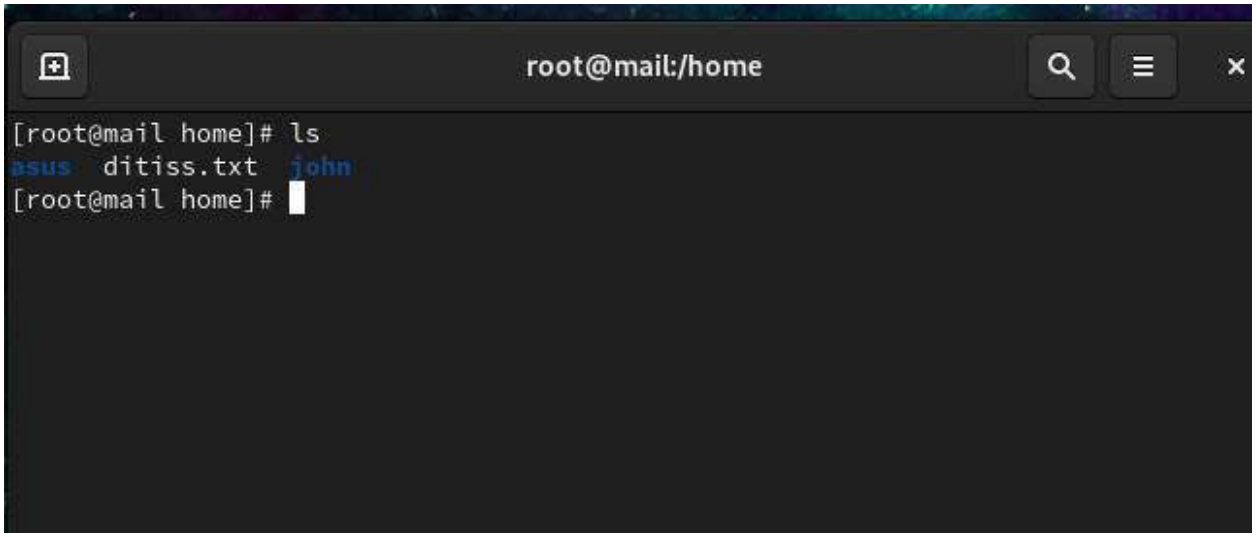
/home

#touch ditiss.txt to create a txt file

scp ditiss.txt ssh-X 192.168.1.20:/home/

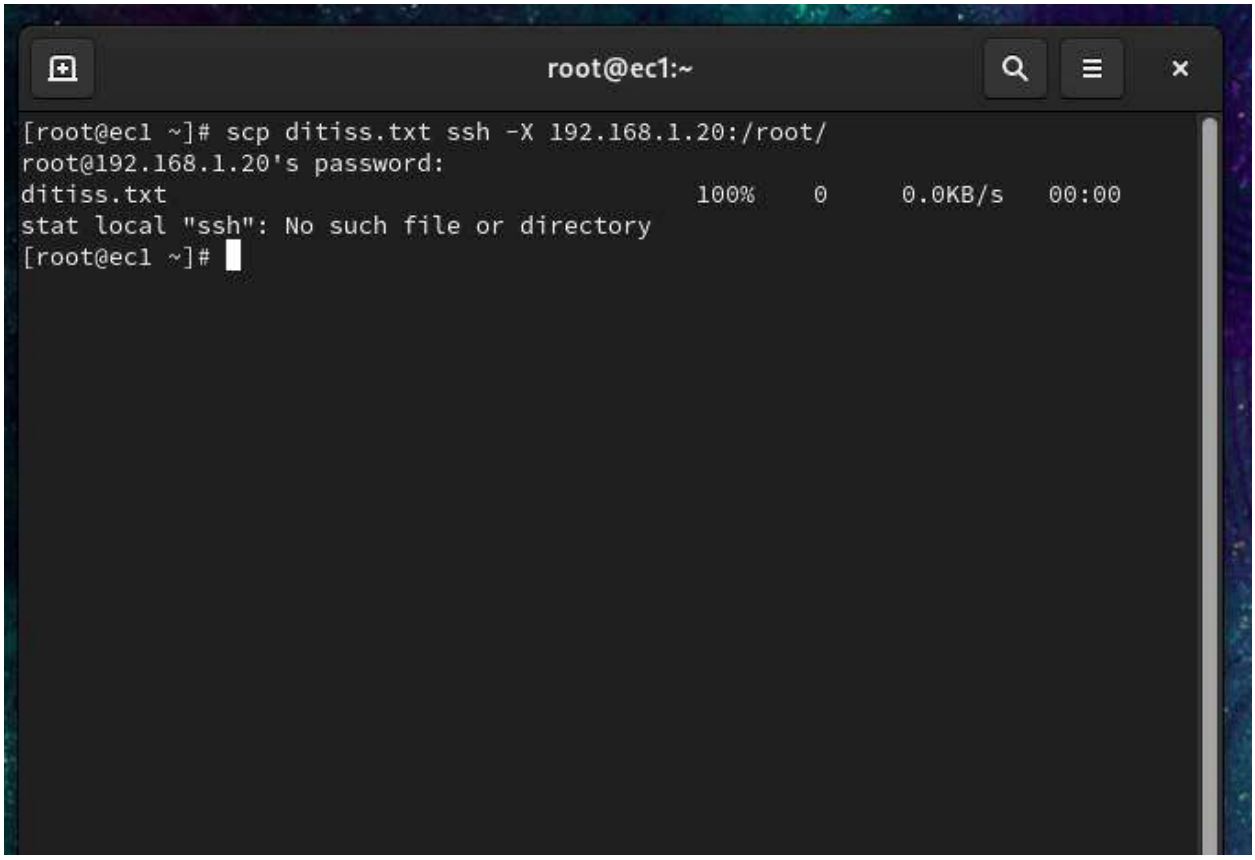


2. Check the file in client in /home folder

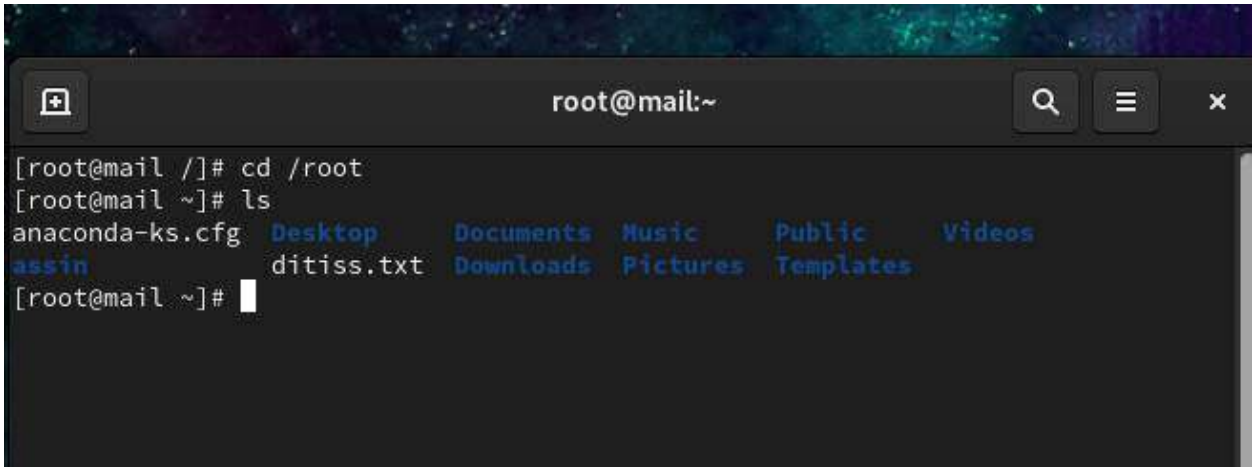


B.

1. Create empty file touch ditiss.txt in server side then copy this file in client-side /root

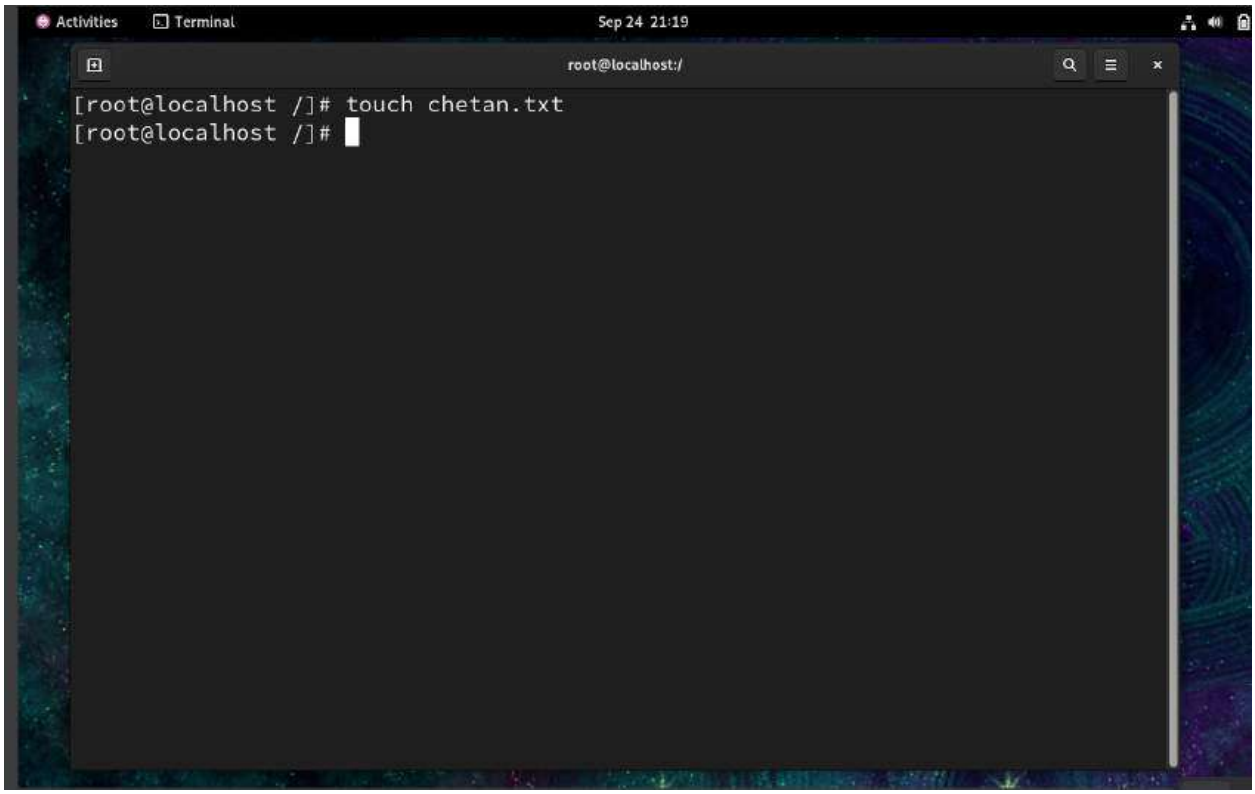


2. Check the file it is copied in client side
   #cd /root to go to root



Q.4 Use chmod command to change the permission of newly created file

1. Change the permission to rwxrwxr__
2. First create the file by using touch command



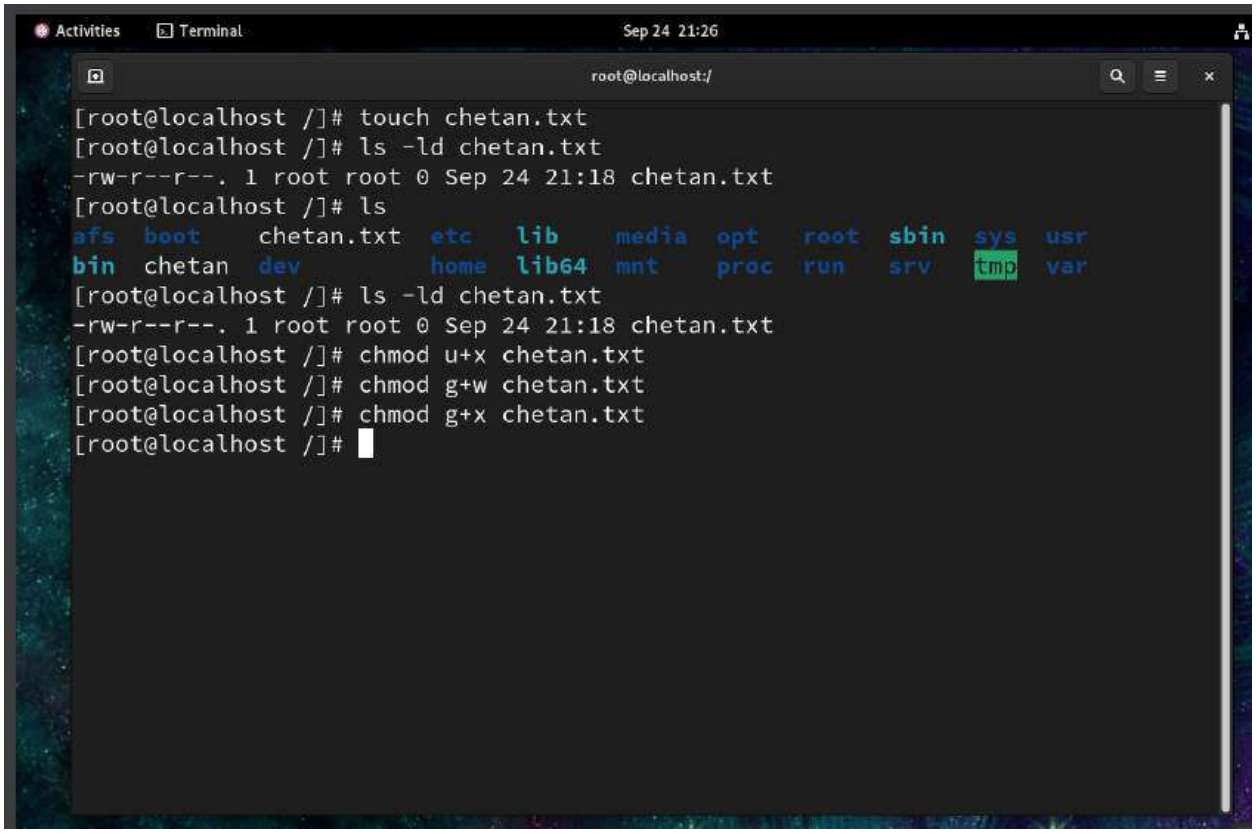
3.As you can see the file has been created

```
[root@localhost /]# ls
afs  boot    chetan.txt  etc   lib    media  opt   root  sbin  sys  usr
bin  chetan  dev         home  lib64  mnt    proc  run   srv   tmp  var
[root@localhost /]#
```

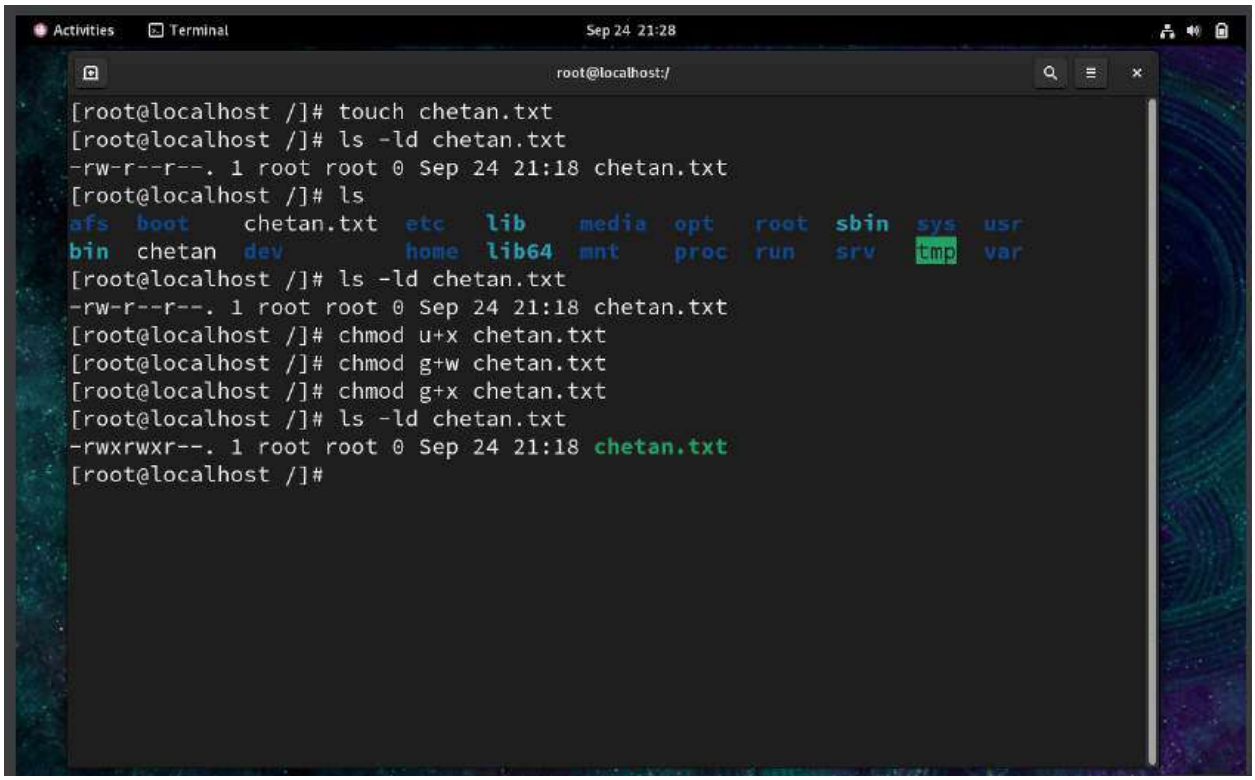
4.Use ls -ld command to check the file default permission

```
Activities   Terminal                    Sep 24 21:22
root@localhost:/
[root@localhost /]# touch chetan.txt
[root@localhost /]# ls -ld chetan.txt
-rw-r--r--. 1 root root 0 Sep 24 21:18 chetan.txt
[root@localhost /]#
```

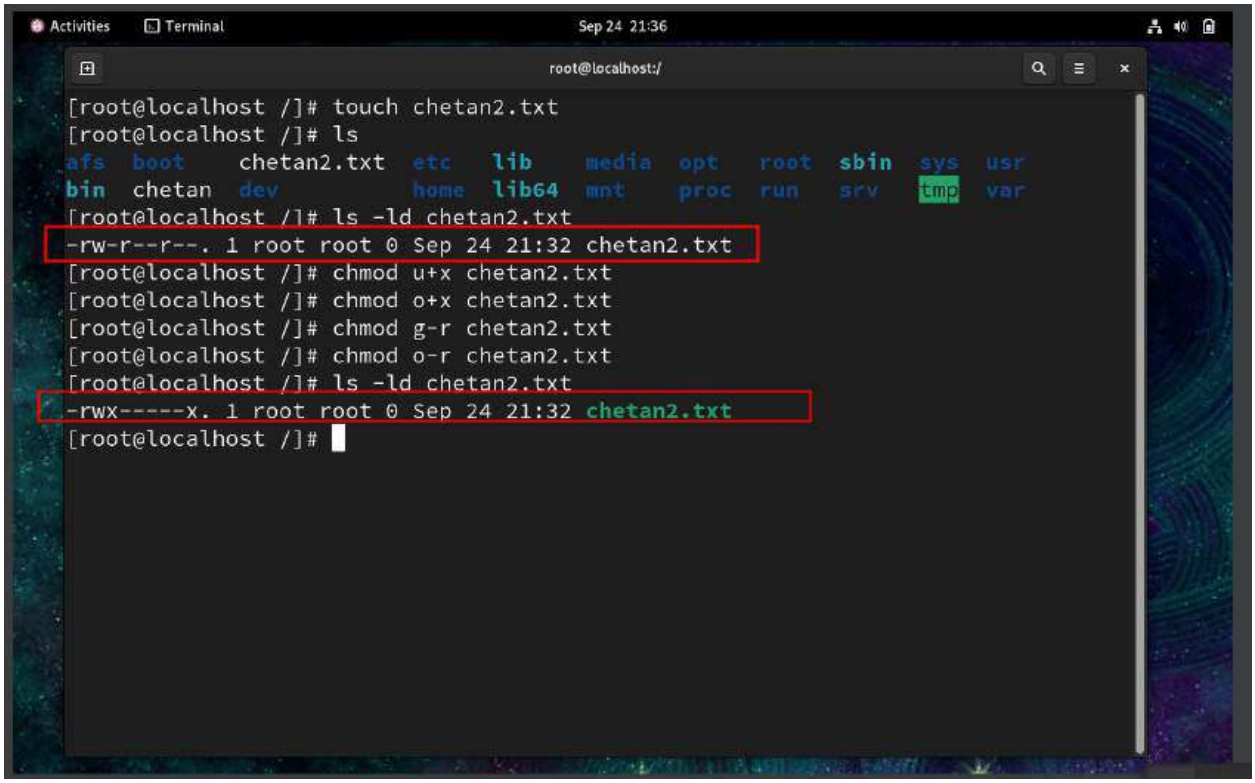
5. We have to use chmod command to change the file permission



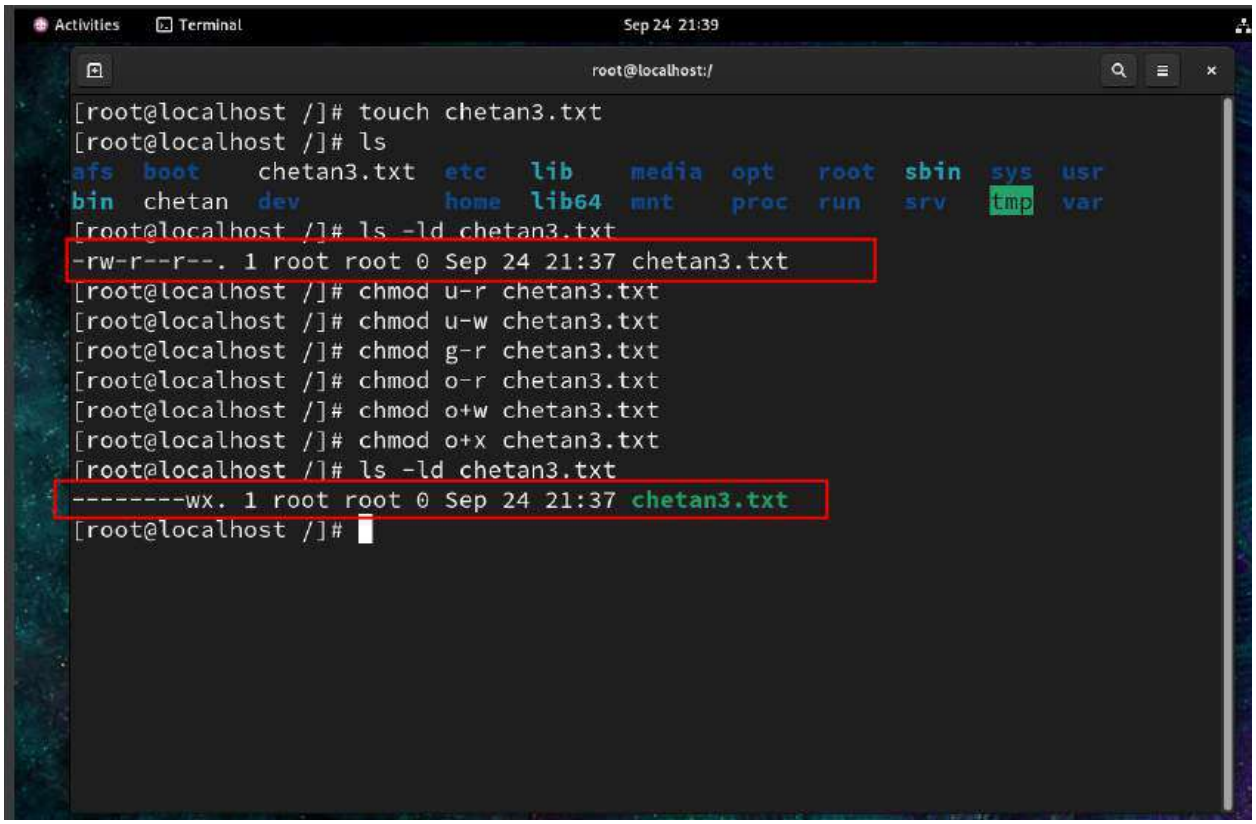
6. As you can see the file permission has been changed



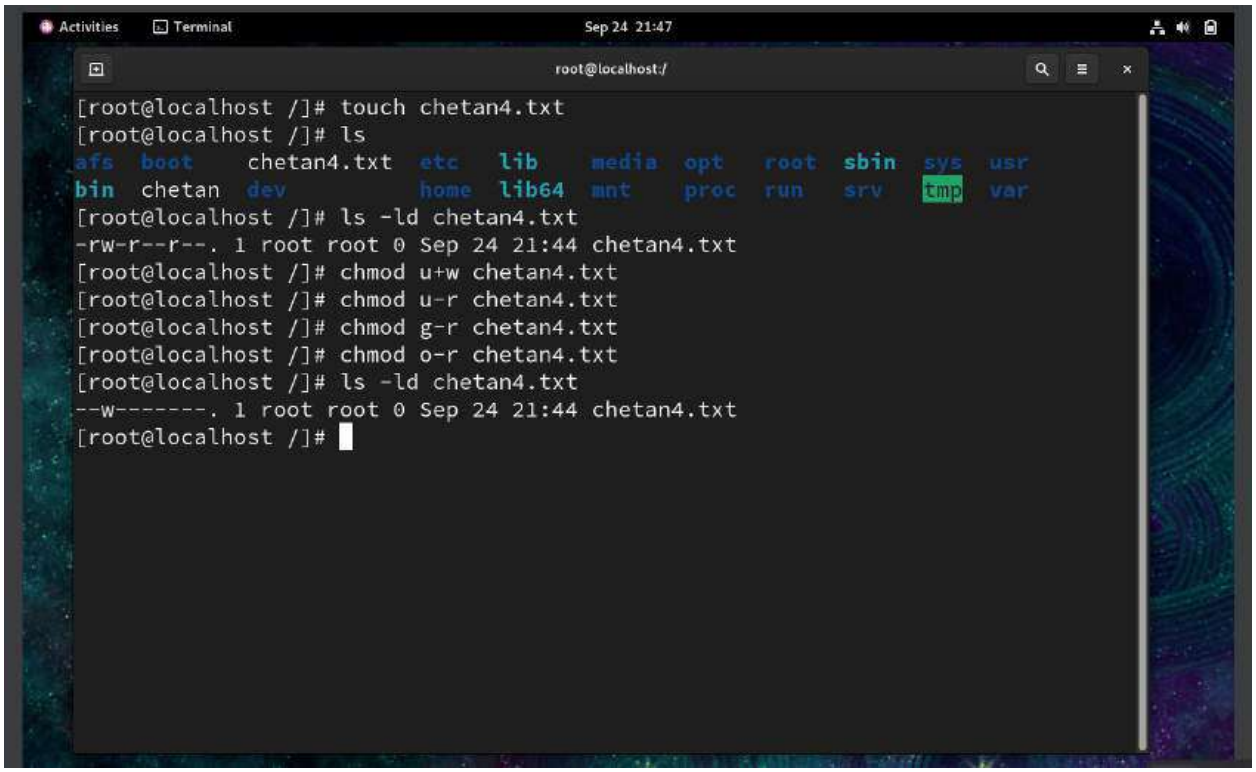
7. Change the permission to rwx_____x



8. Change the permission to _____wx



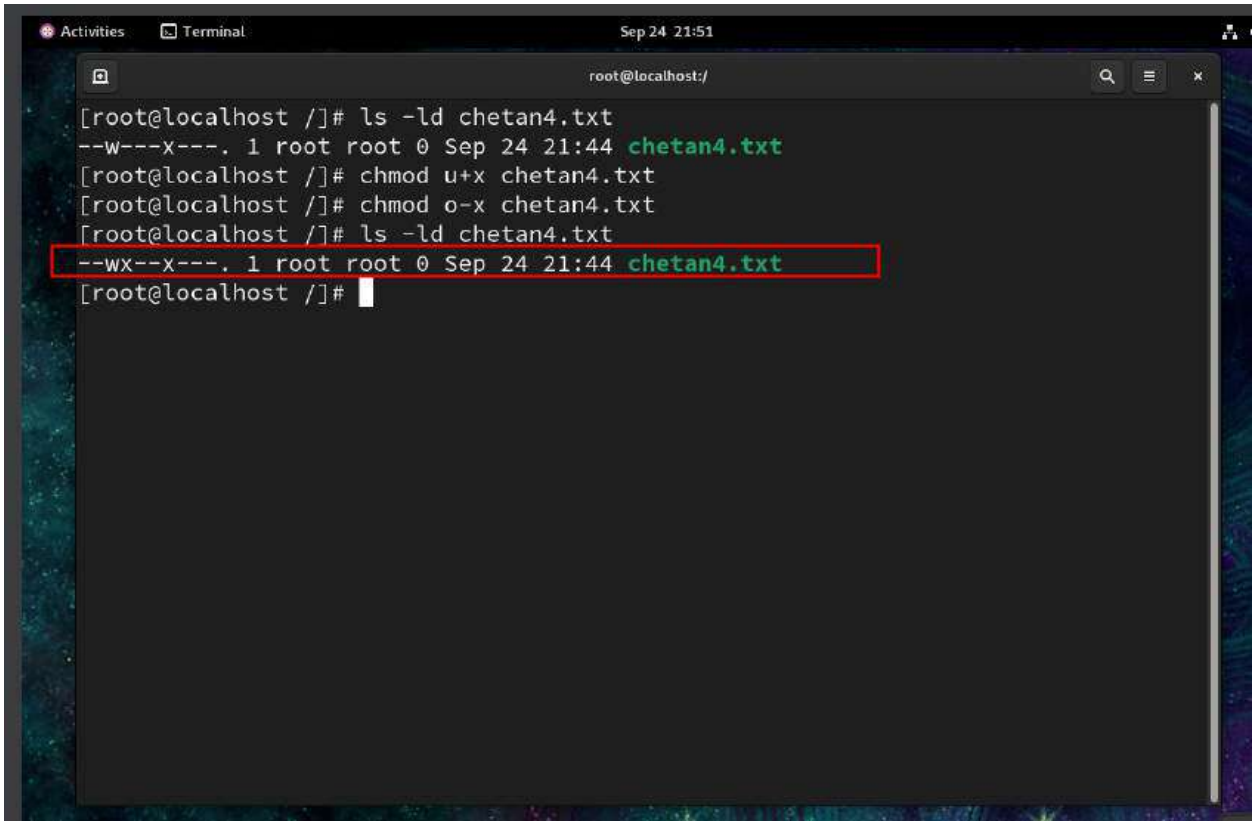
9. Give write permission to the owner



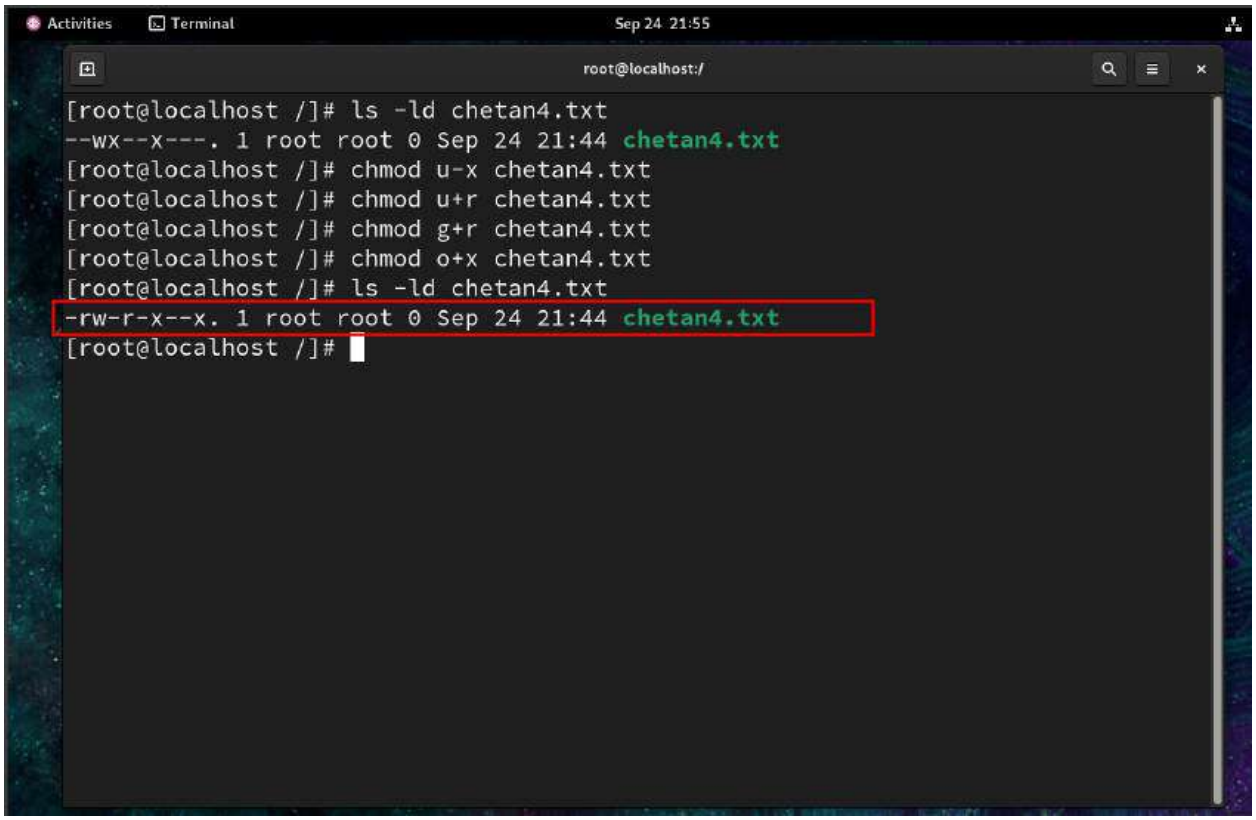
10. Give execute permission to the group owner

```
Activities   Terminal                    Sep 24 21:48
                          root@localhost:/
[root@localhost /]# touch chetan4.txt
[root@localhost /]# ls
afs  boot    chetan4.txt  etc   lib    media  opt   root  sbin  sys  usr
bin  chetan  dev          home  lib64  mnt    proc  run   srv   tmp  var
[root@localhost /]# ls -ld chetan4.txt
-rw-r--r--. 1 root root 0 Sep 24 21:44 chetan4.txt
[root@localhost /]# chmod u+w chetan4.txt
[root@localhost /]# chmod u-r chetan4.txt
[root@localhost /]# chmod g-r chetan4.txt
[root@localhost /]# chmod o-r chetan4.txt
[root@localhost /]# ls -ld chetan4.txt
--w-------. 1 root root 0 Sep 24 21:44 chetan4.txt
[root@localhost /]# chmod g-x chetan4.txt
[root@localhost /]# ls -ld chetan4.txt
--w-------. 1 root root 0 Sep 24 21:44 chetan4.txt
[root@localhost /]# chmod g+x chetan4.txt
[root@localhost /]# ls -ld chetan4.txt
--w---x---. 1 root root 0 Sep 24 21:44 chetan4.txt
[root@localhost /]#
```

11. Remove execute permission to others

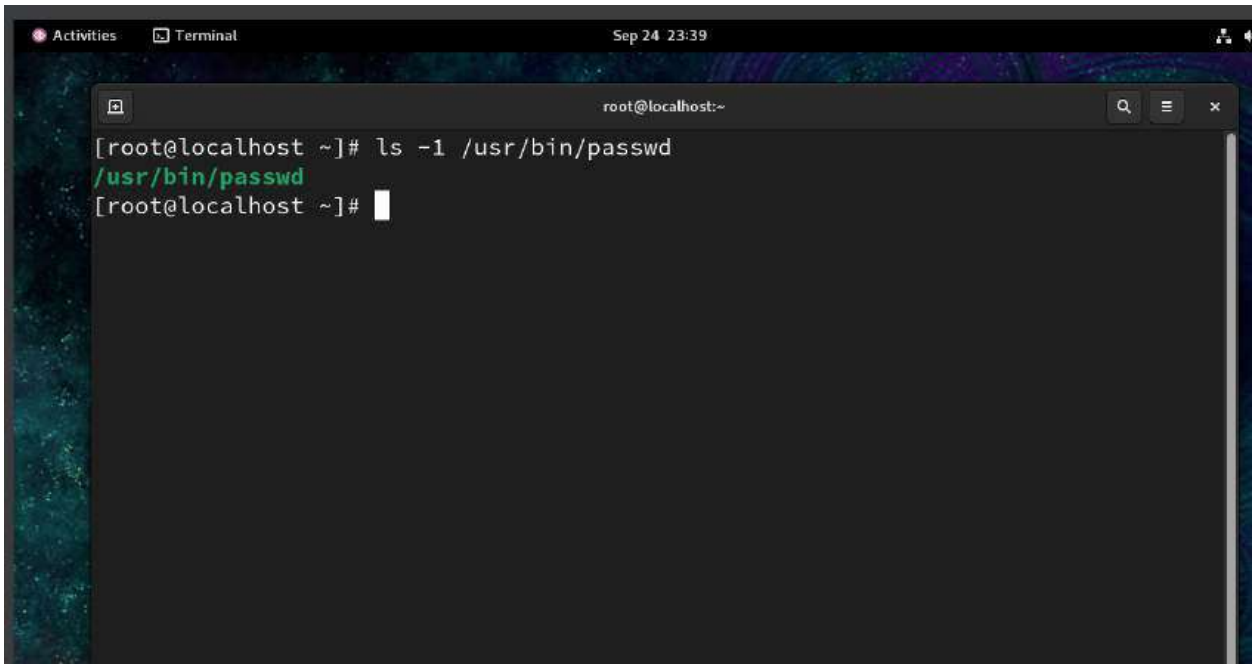


12. Set permission as owner =rw, group owner=rx, others=x

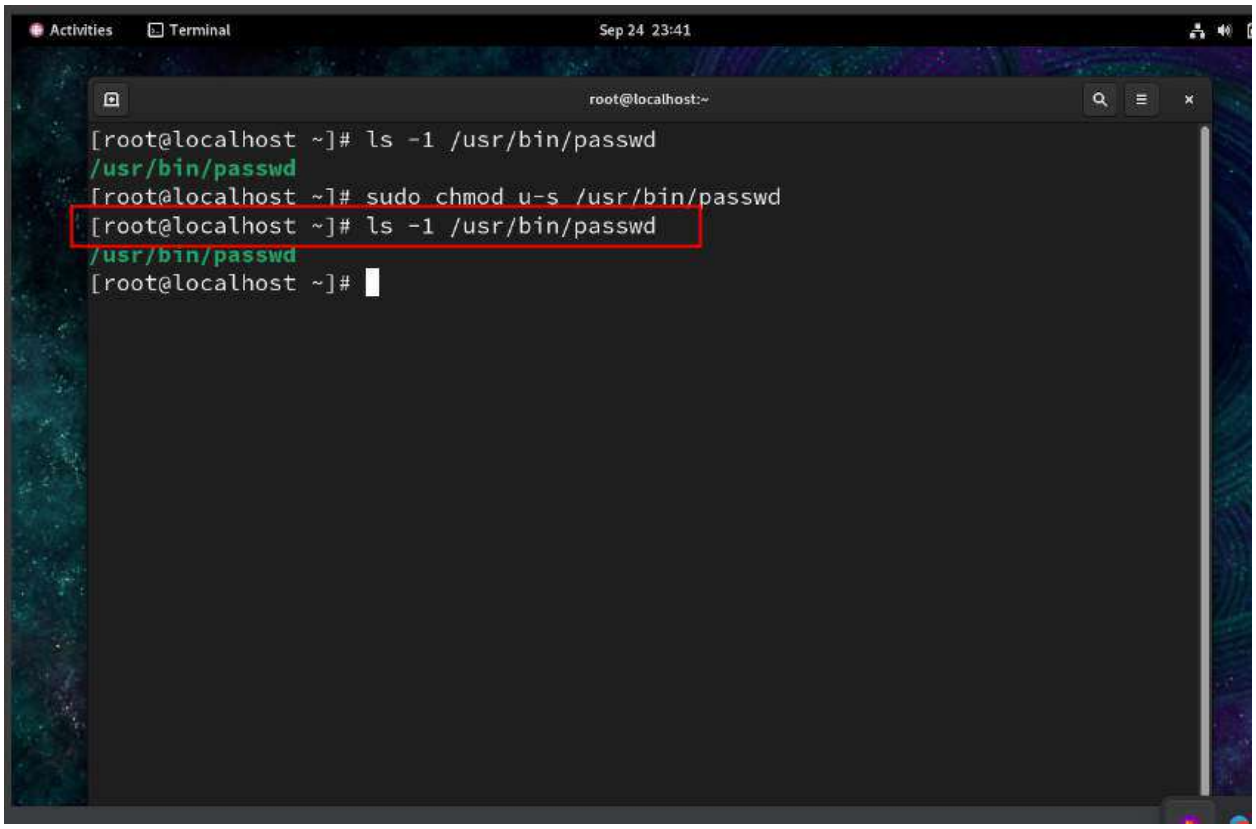


Q. 5. Remove suid permission from passwd command on your Linux system and try changing your password.
Give back the suid permission to passwd command and try changing your password.

1. Checking the current permission of the passwd command



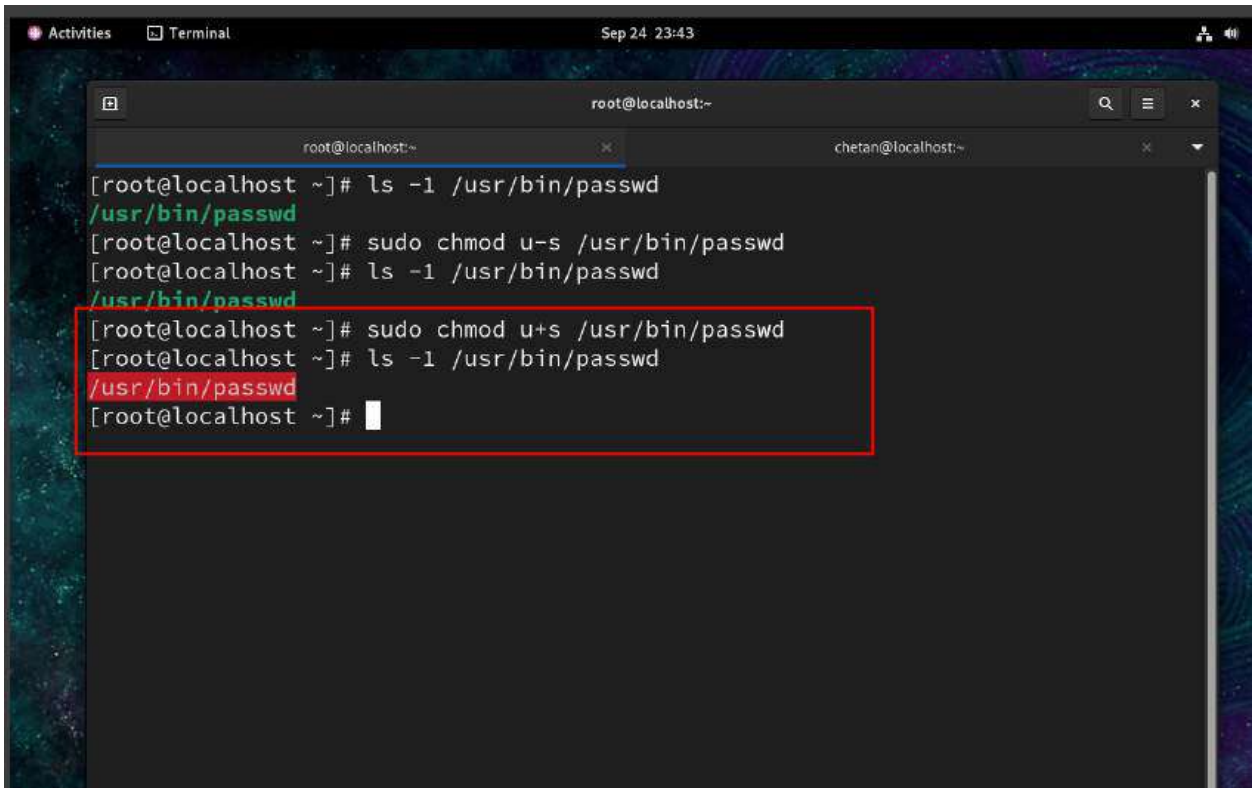
2. Remove the SUID from the passwd command and verify the suid has been removed



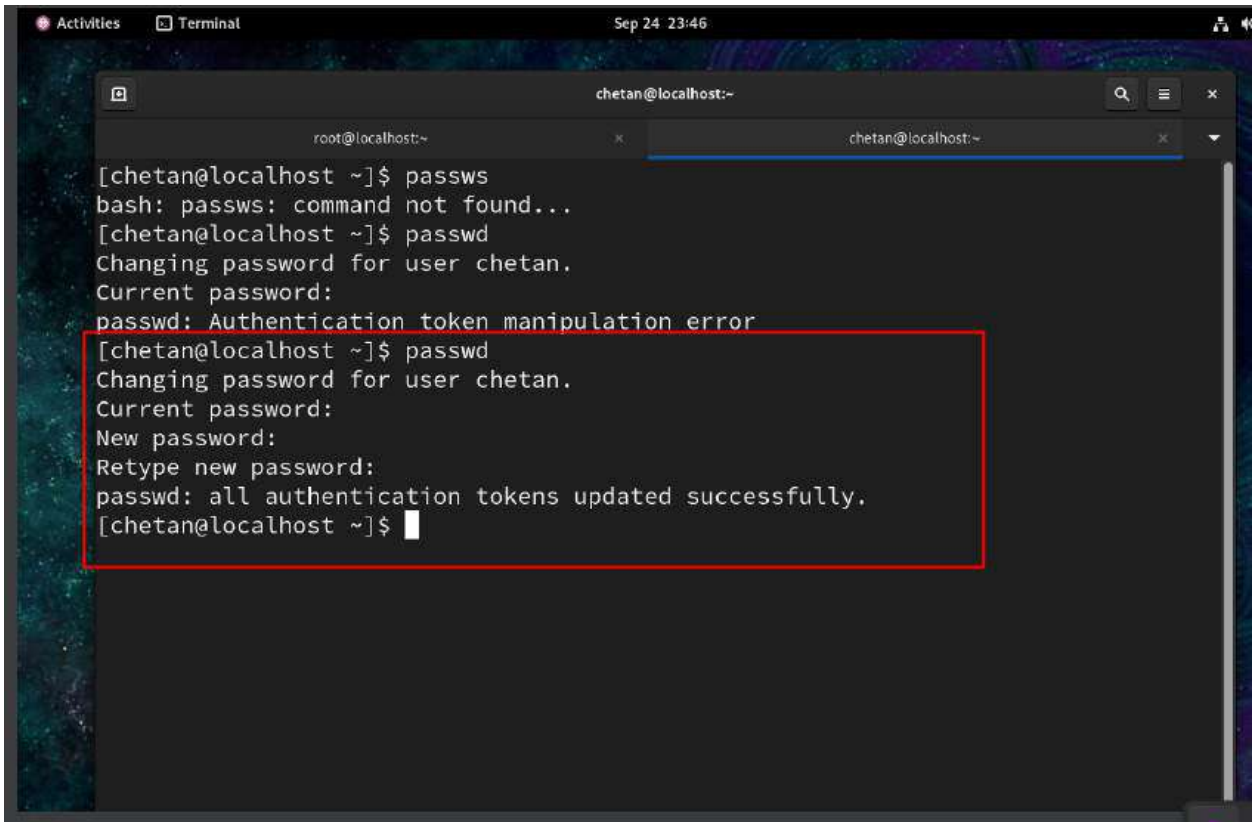
3. Try to change the passwd but it show error

```
[chetan@localhost ~]$ passwd
Changing password for user chetan.
Current password:
passwd: Authentication token manipulation error
[chetan@localhost ~]$
```

4. Change to default setting

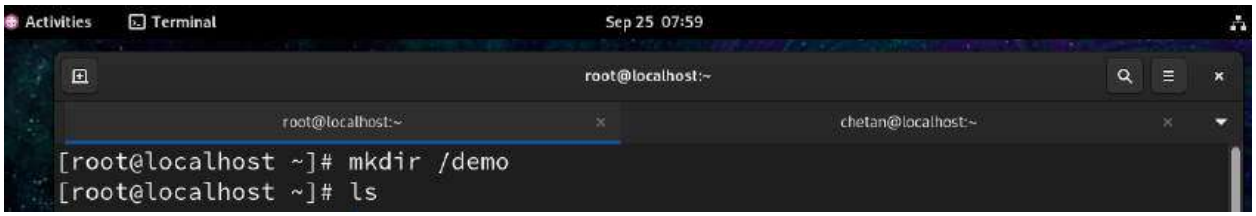


5. Now we can change the passwd by the user

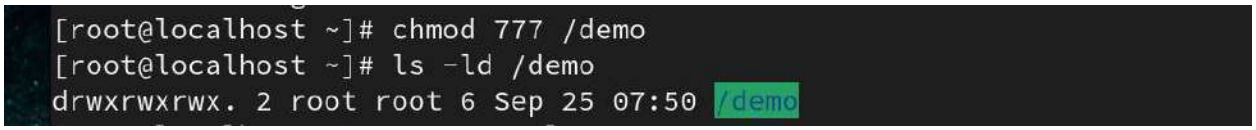


6. Create /demo folder as root, give 777 permission to /demo folder and login as normal user and try creating files and folder in /demo folder. Now add sgid permission to the /demo folder using root. Again try creating files and folder as normal user in /demo folder. Write the observations.

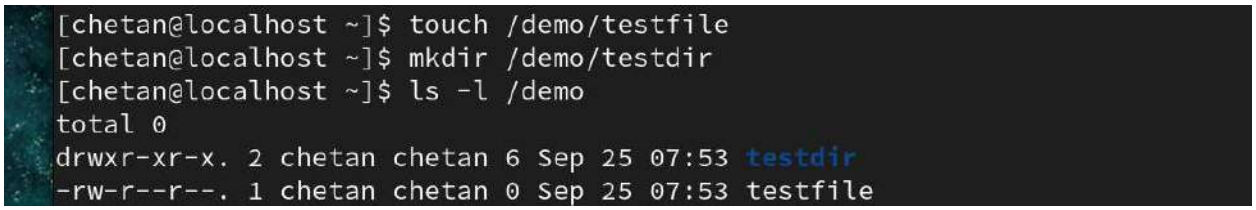
1. Login as root user, make directory named demo



2.Given 777 permission to demo folder



3. Login as normal user and created a file



4. Using root we added SGID permission to demo folder

```
[root@localhost ~]# chmod g+s /demo
[root@localhost ~]# ls -ld /demo
drwxrwsrwx. 3 root root 37 Sep 25 07:53 /demo
```

5.Created file as normal user

```
[chetan@localhost ~]$ touch /demo/testfile_sgid
[chetan@localhost ~]$ mkdir /demo/testdir_sgid
[chetan@localhost ~]$ ls -l /demo
total 0
drwxr-xr-x. 2 chetan chetan 6 Sep 25 07:53 testdir
drwxr-sr-x. 2 chetan root   6 Sep 25 07:56 testdir_sgid
-rw-r--r--. 1 chetan chetan 0 Sep 25 07:53 testfile
-rw-r--r--. 1 chetan root   0 Sep 25 07:55 testfile_sgid
```

Before Setting SGID:

- The normal user can create files and directories because of `777` permissions.
- The ownership of the files and directories is set to the user and the user's group.

After Setting SGID:

- The normal user can still create files and directories.
- The ownership of the files remains with the normal user, but the **group ownership** is inherited from the `/demo` folder (which is `root`).
- This behavior ensures consistent group ownership for files and directories created inside `/demo`, which can be useful for shared folders and files.

7. Login as root and create a folder called "/assin", give 777 permission to /assin, now login as one user (user1) and try creating files and folders in /assin. Now login as other user (user2) and try deleting the file created by user1. Write the observations. Now assign sticky bot to /assin folder and do the same mentioned above and write your observation.

1. Created a folder name assin

```
[root@localhost ~]# mkdir assin
[root@localhost ~]# ls
anaconda-ks.cfg  Desktop    Downloads  Pictures  Templates
assin            Documents  Music      Public    Videos
[root@localhost ~]# chmod 777 /assin
```

2. Given permission to file

```
[root@localhost ~]# mkdir /assin
[root@localhost ~]# ls
anaconda-ks.cfg  Documents  Music     Public     Videos
Desktop          Downloads  Pictures  Templates
[root@localhost ~]# ls
anaconda-ks.cfg  Documents  Music     Public     Videos
Desktop          Downloads  Pictures  Templates
[root@localhost ~]# mkdir assin
[root@localhost ~]# ls
anaconda-ks.cfg  Desktop    Downloads  Pictures  Templates
assin            Documents  Music      Public    Videos
[root@localhost ~]# chmod 777 /assin
[root@localhost ~]# ls -ld /assin
drwxrwxrwx. 2 root root 6 Oct 20 23:31 /assin
[root@localhost ~]#
```

3.Now login as user 1 and created file and directories

```
[root@localhost ~]# su asus
[asus@localhost root]$ cd
[asus@localhost ~]$ touch /assin/file_user1.txt
[asus@localhost ~]$ mkdir /assin/dir_user1
[asus@localhost ~]$ ls -l /assin
total 0
drwxr-xr-x. 2 asus asus 6 Oct 20 23:35 dir_user1
-rw-r--r--. 1 asus asus 0 Oct 20 23:34 file_user1.txt
[asus@localhost ~]$
```

4. Now login as user 2 and deleted all the files and directory

```
[root@localhost ~]# su john
[john@localhost root]$ cd
[john@localhost ~]$ rm /assin/file_user1.txt
rm: remove write-protected regular empty file '/assin/file_user1.txt'? y
[john@localhost ~]$ rmdir /assin/dir_user1
[john@localhost ~]$ ls -l /assin
total 0
[john@localhost ~]$
```

5.With `777` permissions, both `user1` and `user2` have full access to `/assin`, which includes creating, modifying, and deleting files. Therefore, `user2` **can delete** the files and folders created by `user1`.

6. Now i had applied sticky bot to /assin folder

```
[root@localhost ~]# chmod o+t /assin
[root@localhost ~]# ls -ld /assin
drwxrwxrwt. 2 root root 6 Oct 20 23:37 /assin
[root@localhost ~]#
```

7. Now user1 has created the file

```
[john@localhost root]$ cd
[john@localhost ~]$ touch /assin/file_user1_sticky.txt
[john@localhost ~]$ mkdir /assin/dir_user1_sticky
[john@localhost ~]$
```

8. Now login as user2 trying to delete the files

```
[john@localhost ~]$ su asus
Password:
[asus@localhost john]$ cd
[asus@localhost ~]$ rm /assin/file_user1_sticky.txt
rm: remove write-protected regular empty file '/assin/file_user1_sticky.txt'? y
rm: cannot remove '/assin/file_user1_sticky.txt': Operation not permitted
[asus@localhost ~]$ rmdir /assin/dir_user1_sticky
rmdir: failed to remove '/assin/dir_user1_sticky': Operation not permitted
[asus@localhost ~]$
```

After the sticky bit is applied, **only the owner of a file or directory can delete** it. Therefore, `user2` **cannot delete** the files and folders created by `user1` in `/assin`.

Without the sticky bit: Any user with write permission can delete files and directories created by other users.
With the sticky bit: Only the owner of a file or directory can delete it, even if other users have write permissions to the directory.