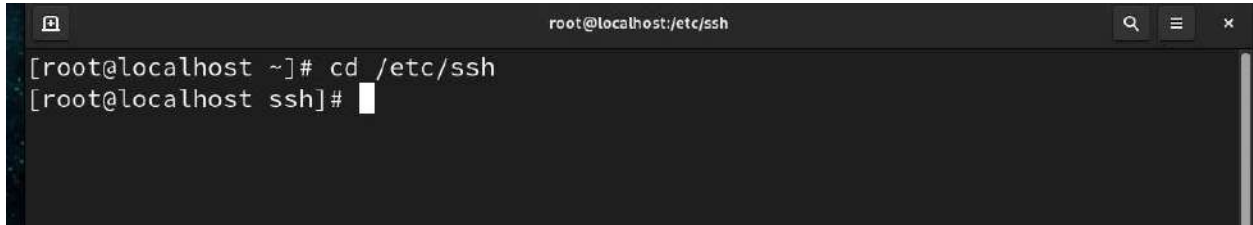


Assignment no.1

1. Go to the ssh folder

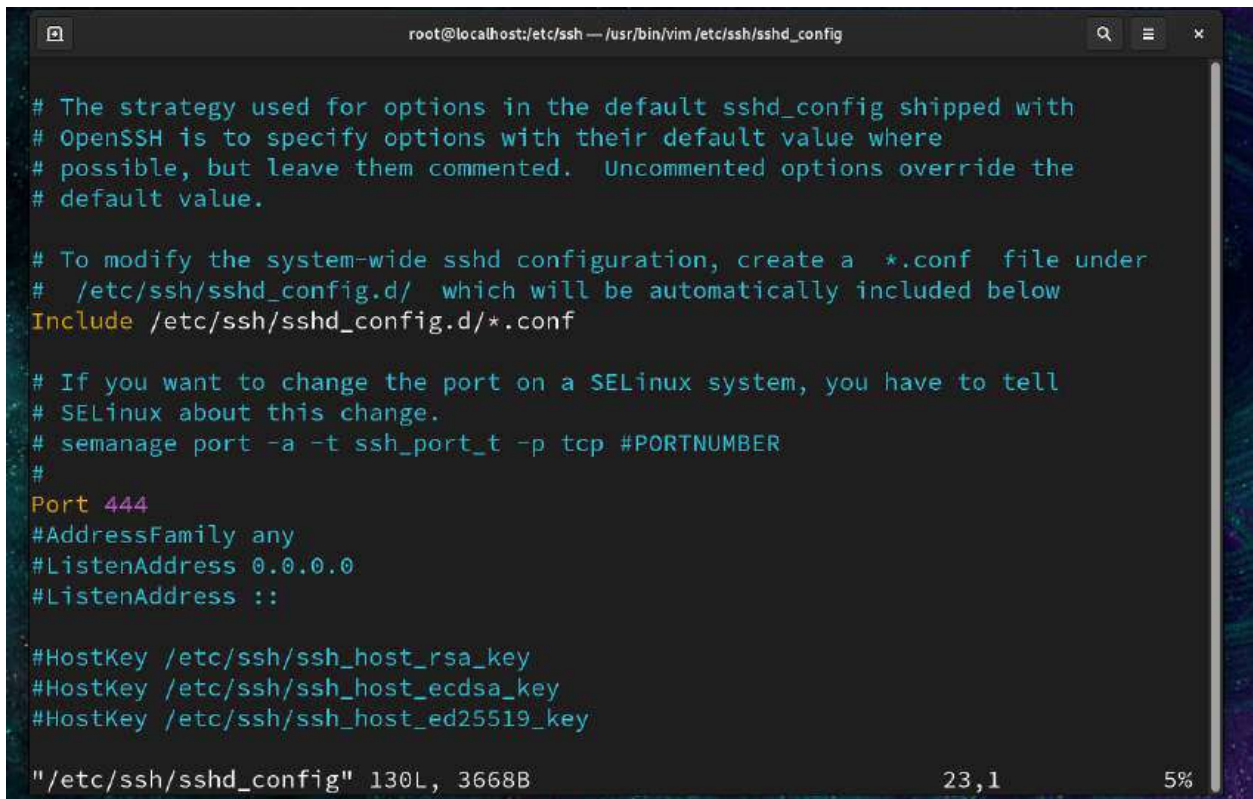


```
root@localhost:/etc/ssh
[root@localhost ~]# cd /etc/ssh
[root@localhost ssh]#
```

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



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

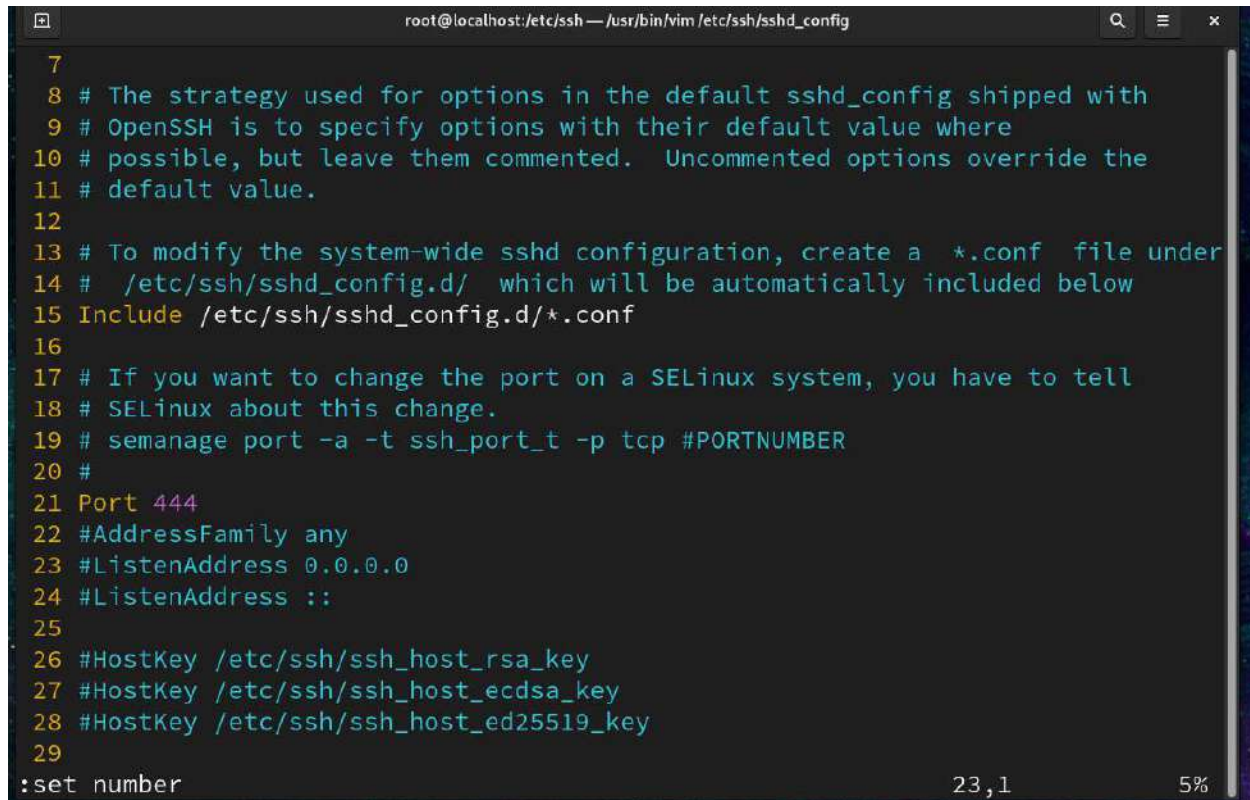
# To modify the system-wide sshd configuration, create a *.conf file under
# /etc/ssh/sshd_config.d/ which will be automatically included below
Include /etc/ssh/sshd_config.d/*.conf

# 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%
```

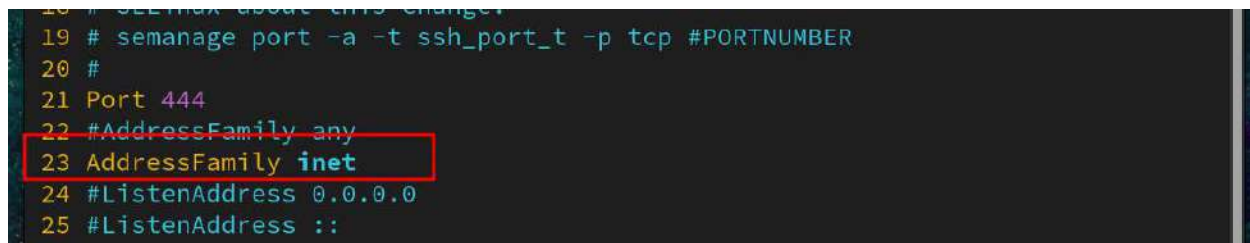
3. Use this command :set number



```
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

```
99
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

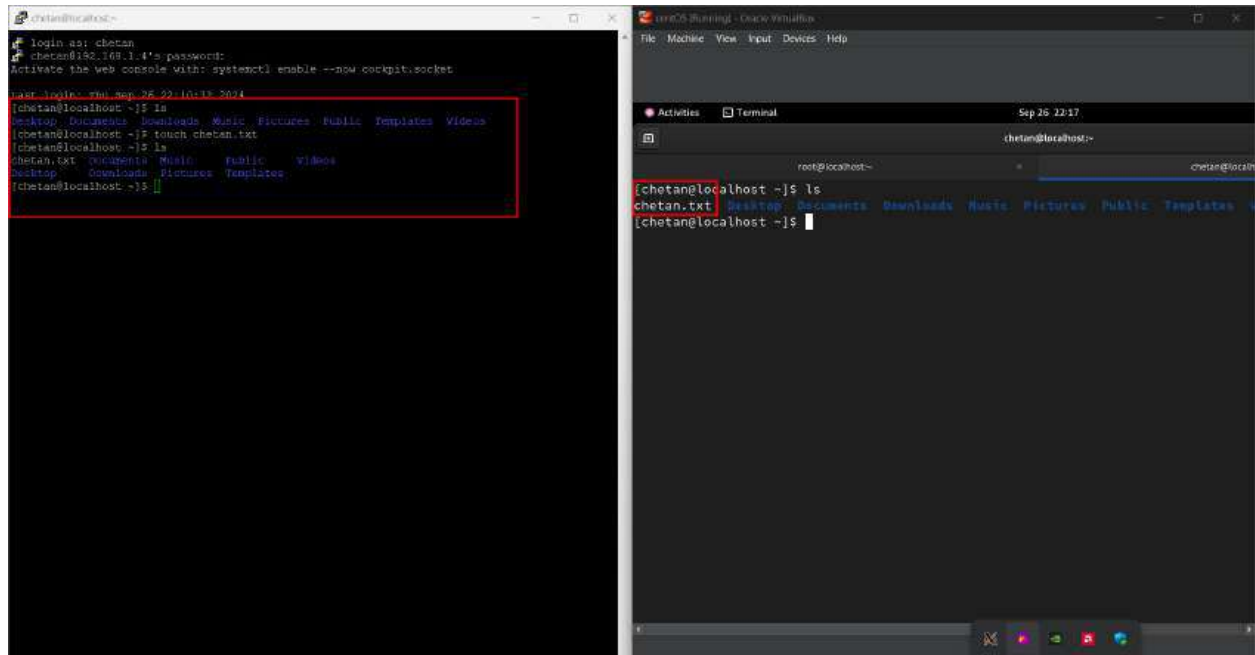
```

[... rules ...]
[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

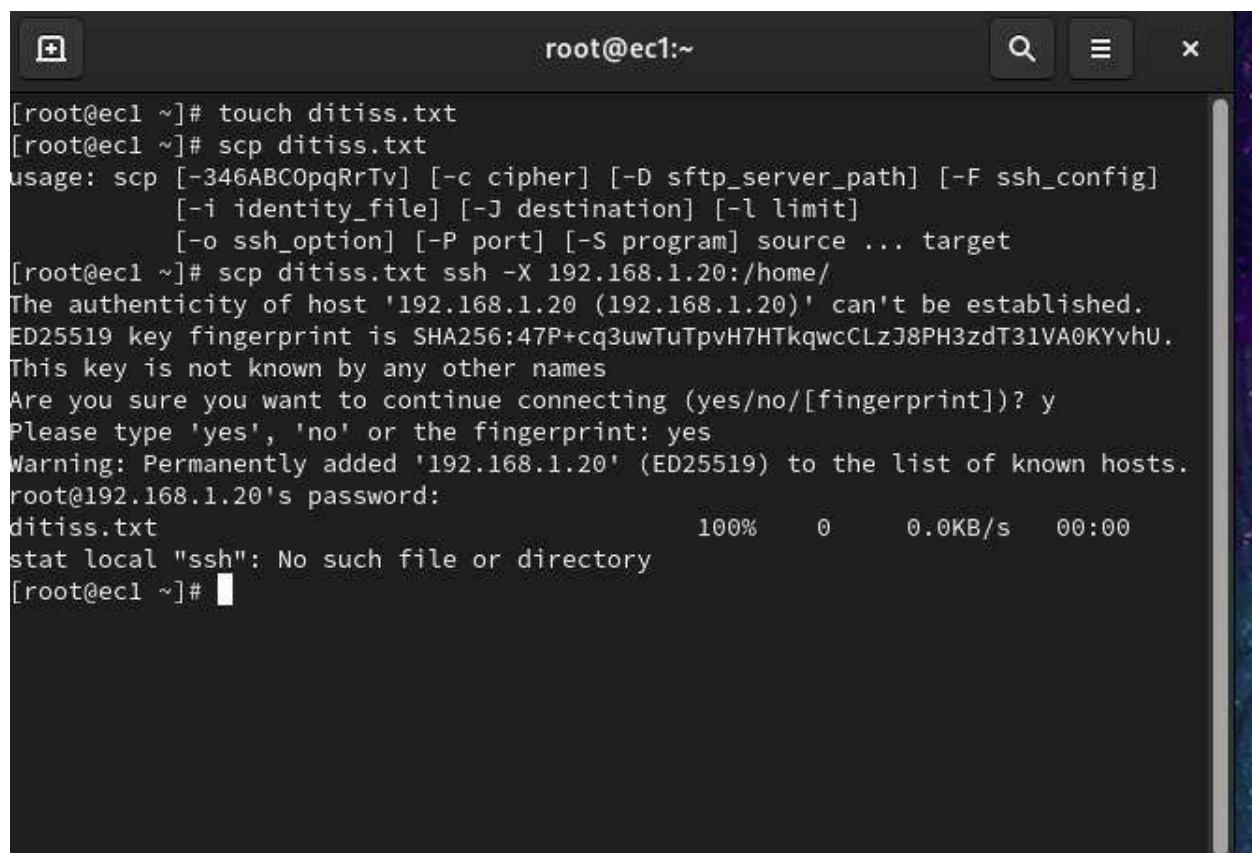
a.

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

/home

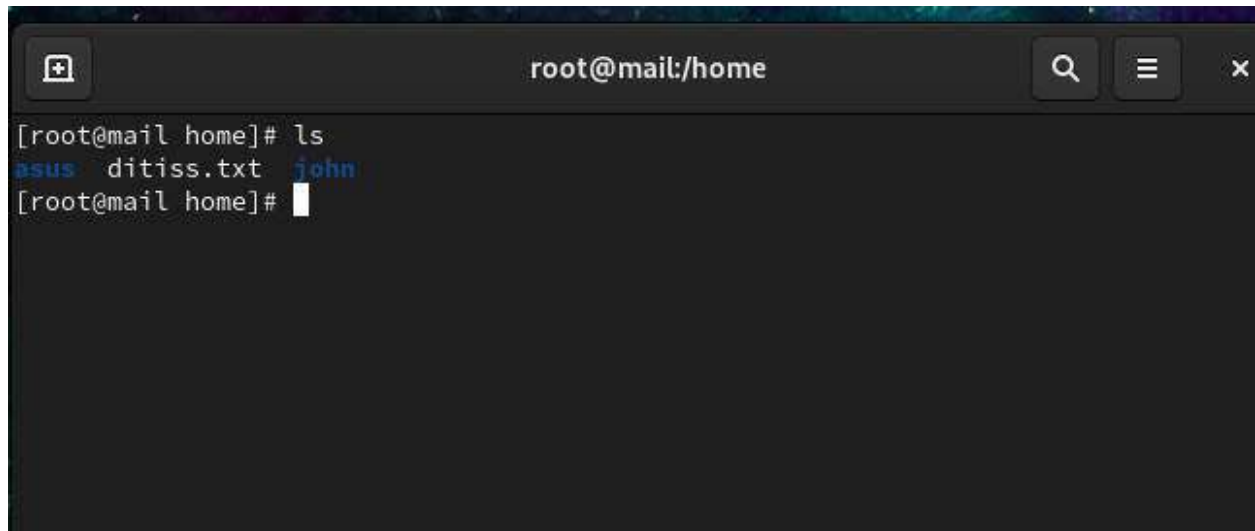
#touch ditiss.txt to create a txt file

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

A terminal window titled 'root@ec1:~' with search, menu, and close icons in the title bar. The terminal shows the following commands and output:

```
[root@ec1 ~]# touch ditiss.txt
[root@ec1 ~]# scp ditiss.txt
usage: scp [-346ABCOpqRrTv] [-c cipher] [-D sftp_server_path] [-F ssh_config]
          [-i identity_file] [-J destination] [-l limit]
          [-o ssh_option] [-P port] [-S program] source ... target
[root@ec1 ~]# scp ditiss.txt ssh -X 192.168.1.20:/home/
The authenticity of host '192.168.1.20 (192.168.1.20)' can't be established.
ED25519 key fingerprint is SHA256:47P+cq3uwTuTpvH7HTkqwcCLzJ8PH3zdT31VA0KYvhU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '192.168.1.20' (ED25519) to the list of known hosts.
root@192.168.1.20's password:
ditiss.txt                                100%    0    0.0KB/s   00:00
stat local "ssh": No such file or directory
[root@ec1 ~]#
```

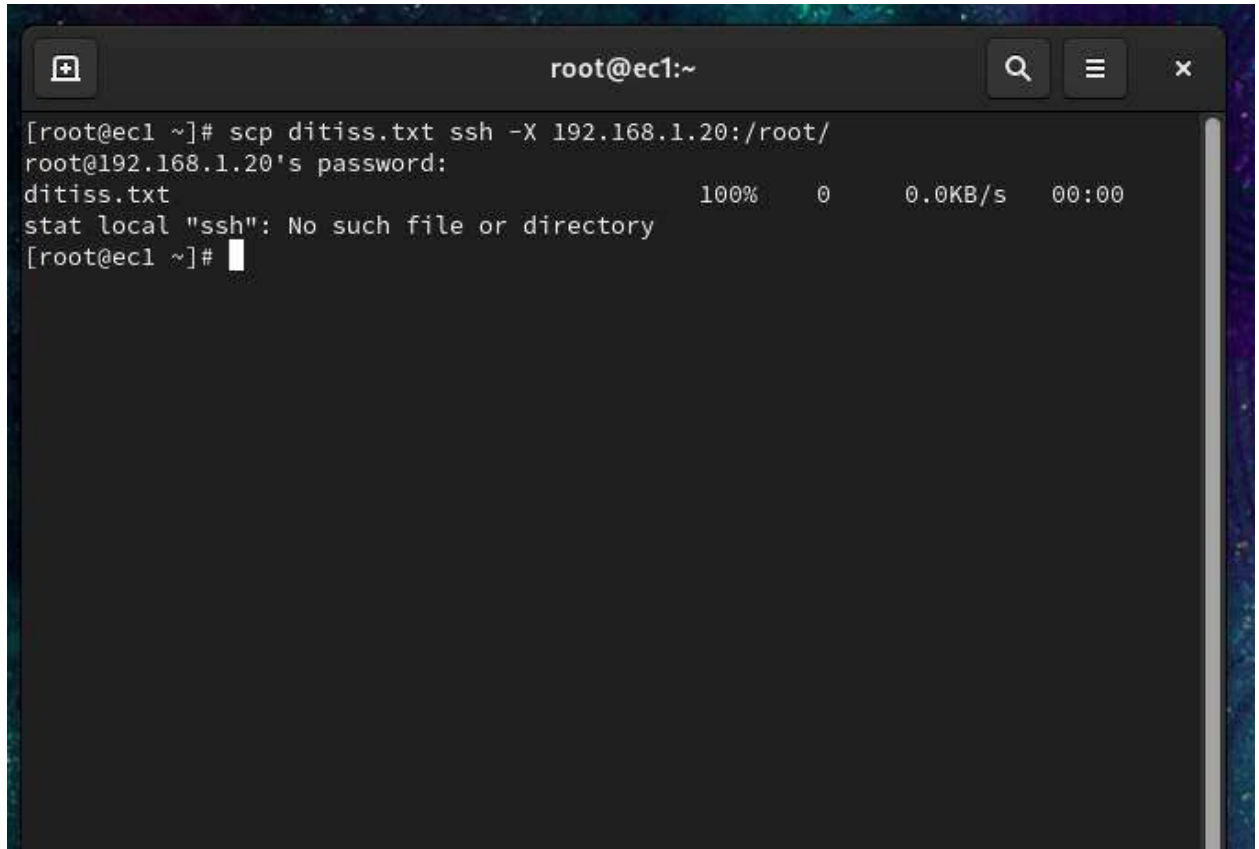
2. Check the file in client in /home folder

A terminal window with a dark background and a title bar. The title bar contains a window icon, the text 'root@mail:/home', and search, menu, and close buttons. The terminal shows the command 'ls' being executed, resulting in the output 'asus ditiss.txt john'. The prompt '[root@mail home]#' is visible at the start and end of the command line.

```
[root@mail home]# ls
asus  ditiss.txt  john
[root@mail home]#
```


B.

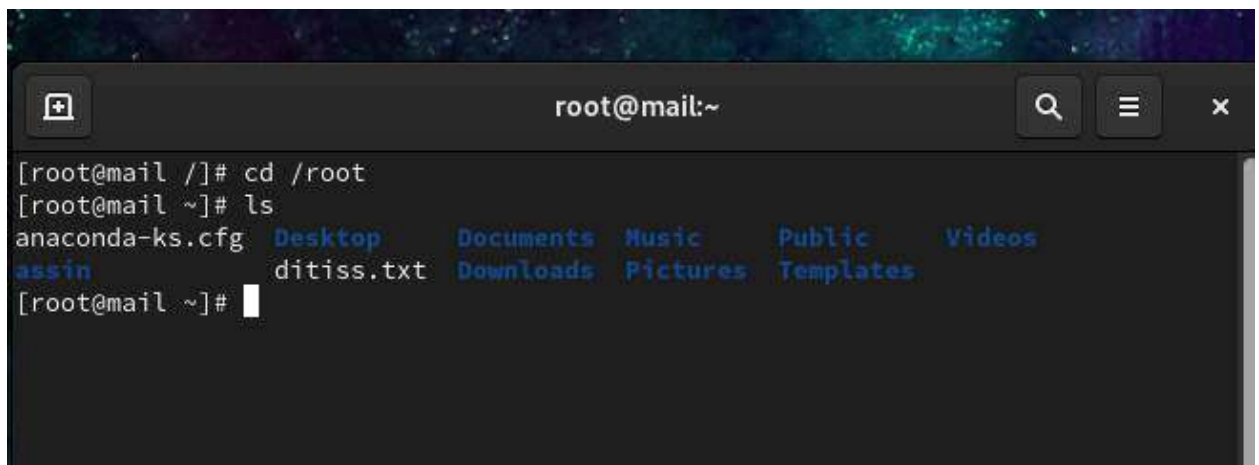
1. Create empty file touch ditiss.txt in server side then copy this file in client-side /root

A terminal window titled 'root@ec1:~' with search, menu, and close icons. It shows the execution of the command 'scp ditiss.txt ssh -X 192.168.1.20:/root/'. The output indicates the file 'ditiss.txt' was copied successfully (100% progress, 0 bytes, 0.0KB/s, 00:00) but then shows an error: 'stat local "ssh": No such file or directory'.

```
[root@ec1 ~]# scp ditiss.txt ssh -X 192.168.1.20:/root/
root@192.168.1.20's password:
ditiss.txt                                100%    0    0.0KB/s   00:00
stat local "ssh": No such file or directory
[root@ec1 ~]#
```

2. Check the file it is copied in client side

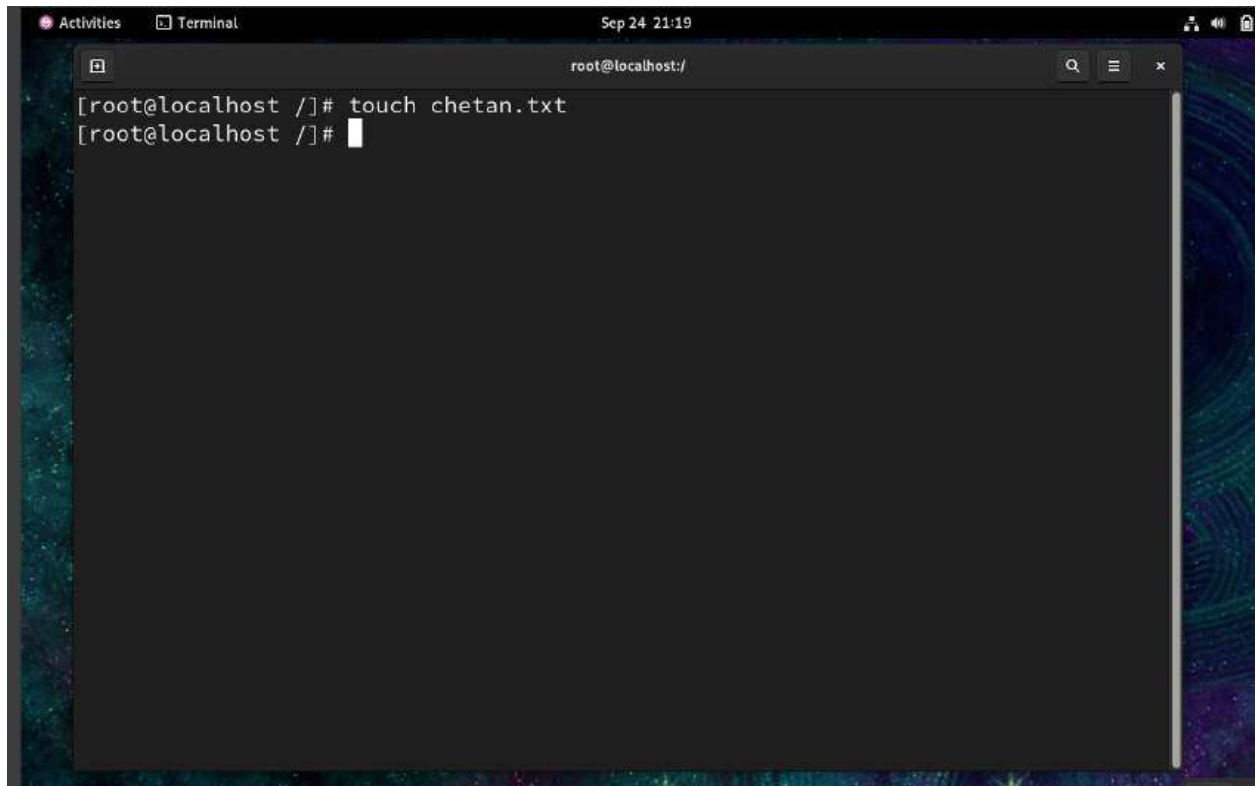
#cd /root to go to root

A terminal window titled 'root@mail:~' with search, menu, and close icons. It shows the user navigating to the root directory with 'cd /root' and then listing files with 'ls'. The output shows various system files and directories, including 'anaconda-ks.cfg', 'Desktop', 'Documents', 'Music', 'Public', 'Videos', 'assin', 'ditiss.txt', 'Downloads', 'Pictures', and 'Templates'.

```
[root@mail /]# cd /root
[root@mail ~]# ls
anaconda-ks.cfg  Desktop      Documents    Music        Public       Videos
assin           ditiss.txt  Downloads    Pictures     Templates
```

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

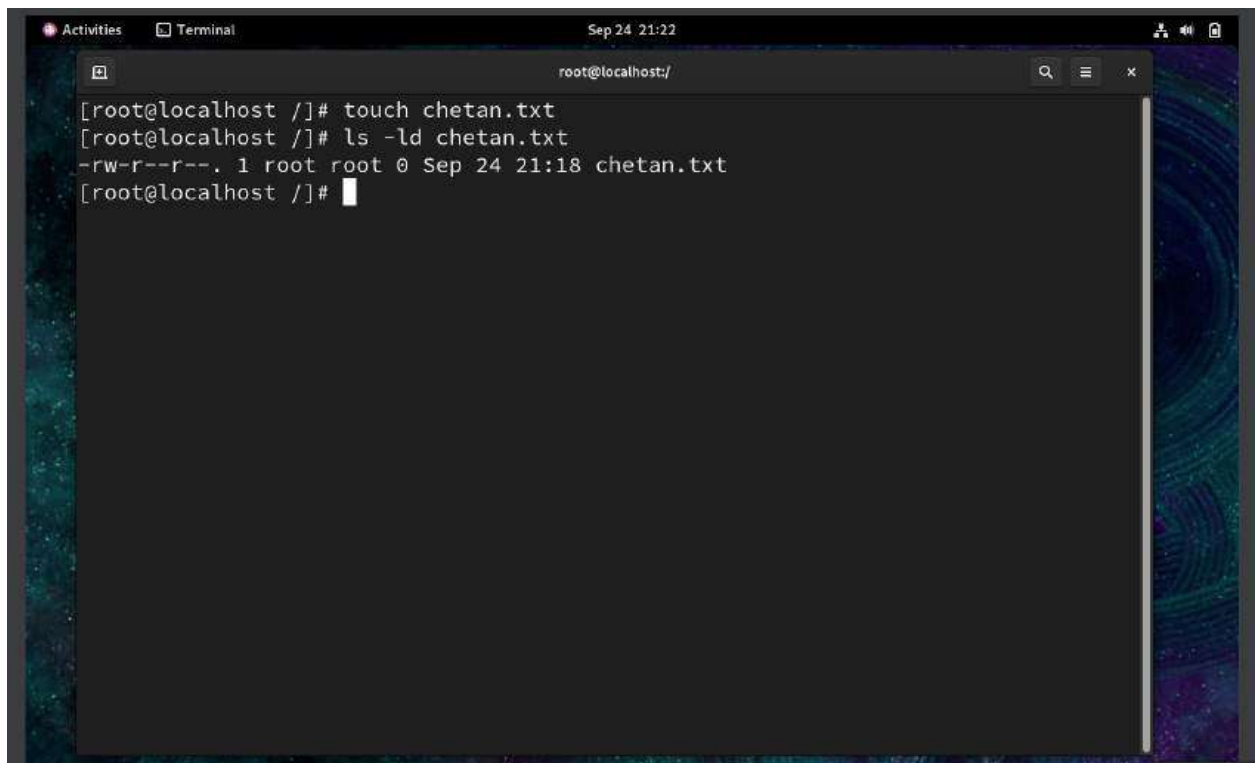
A terminal window titled "Terminal" with a search icon, menu icon, and close button. The window shows the command prompt "root@localhost:/" and the command "touch chetan.txt" being executed. The prompt is now waiting for the next command.

```
root@localhost: /]# touch chetan.txt
root@localhost: /]#
```

3.As you can see the file has been created

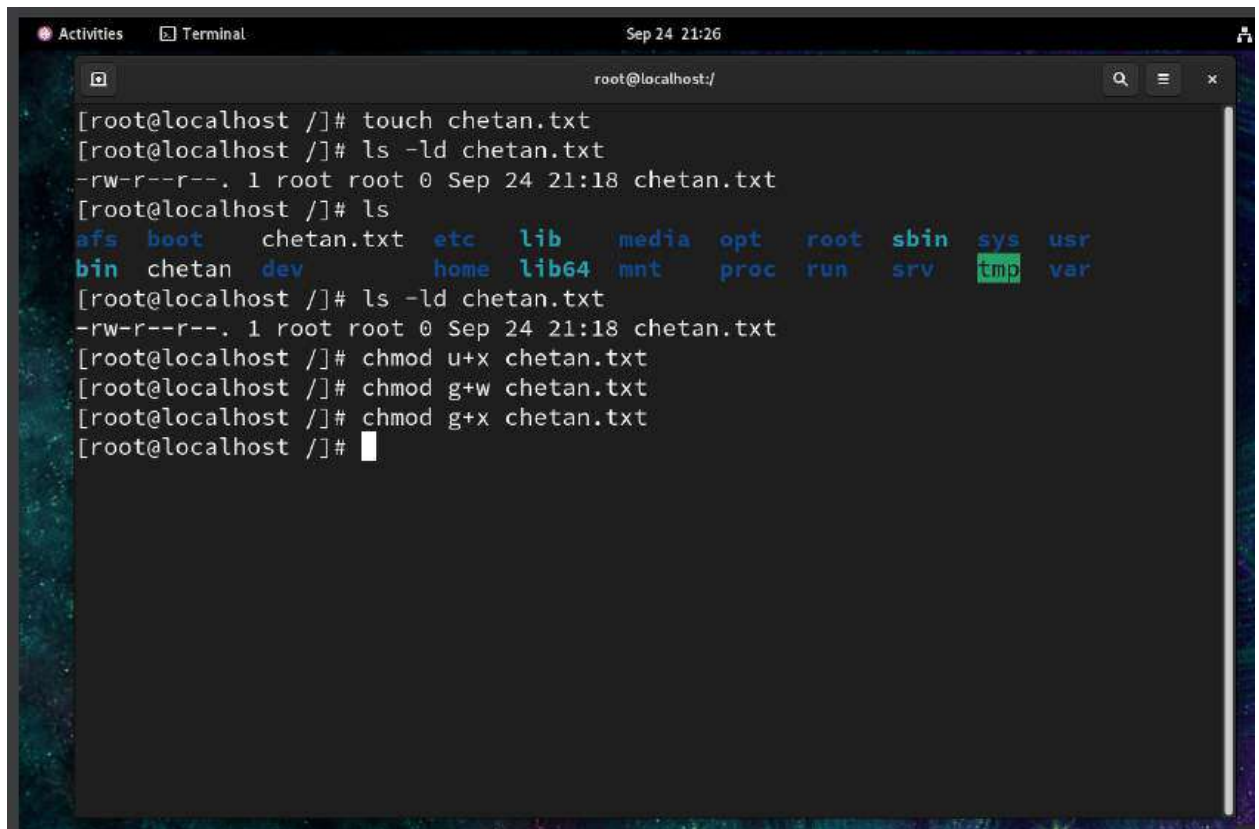
```
[root@localhost /]# ls
afs  boot  chetan.txt  etc  lib  media  opt  root  sbin  sys  usr
bin  chetan  dev  home  lib64  mnt  proc  run  srv  tmp  var
[root@localhost /]#
```

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

A terminal window titled "Terminal" with a search bar and window controls. The prompt is "root@localhost:/". The user enters "touch chetan.txt", then "ls -ld chetan.txt". The output shows the file's permissions as "-rw-r--r--", size as "1", owner as "root", group as "root", and creation time as "Sep 24 21:18".

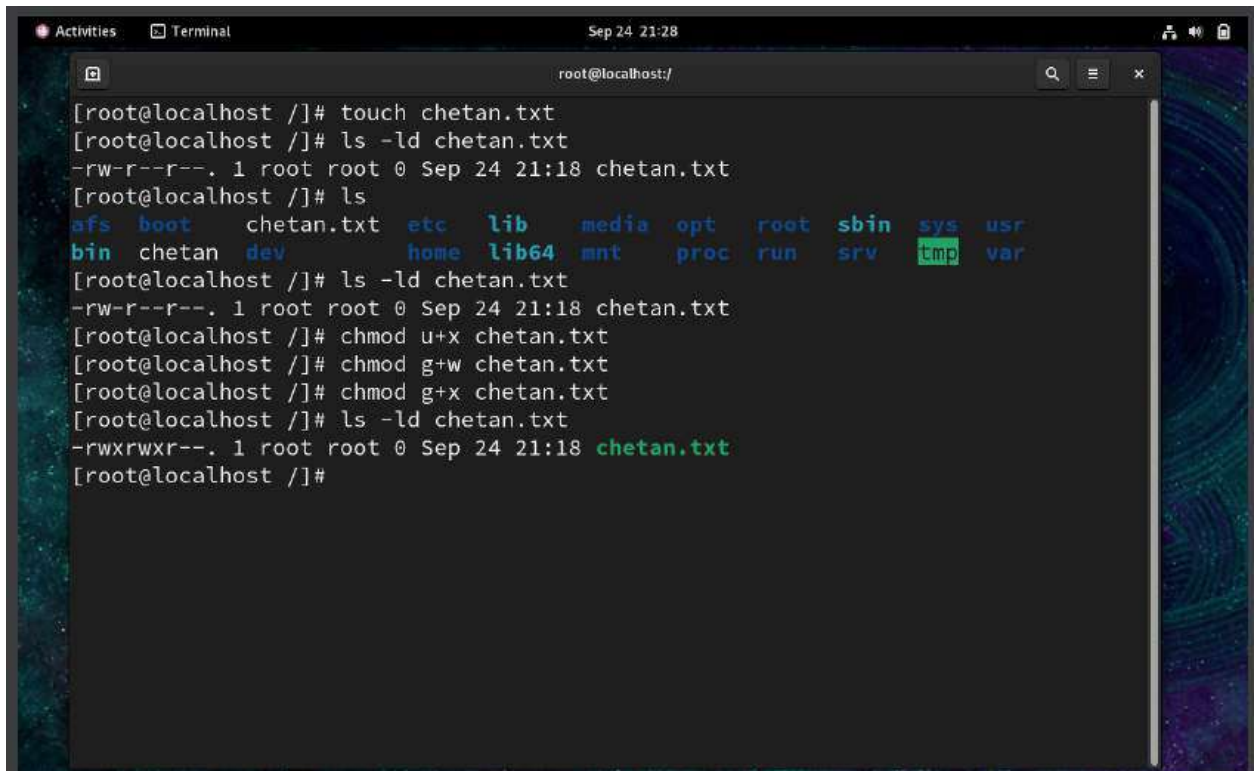
```
Activities Terminal Sep 24 21:22
root@localhost:/
[root@localhost /]# touch chetan.txt
[root@localhost /]# ls -ld chetan.txt
-rw-r--r--. 1 root root 0 Sep 24 21:18 chetan.txt
[root@localhost /]#
```

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



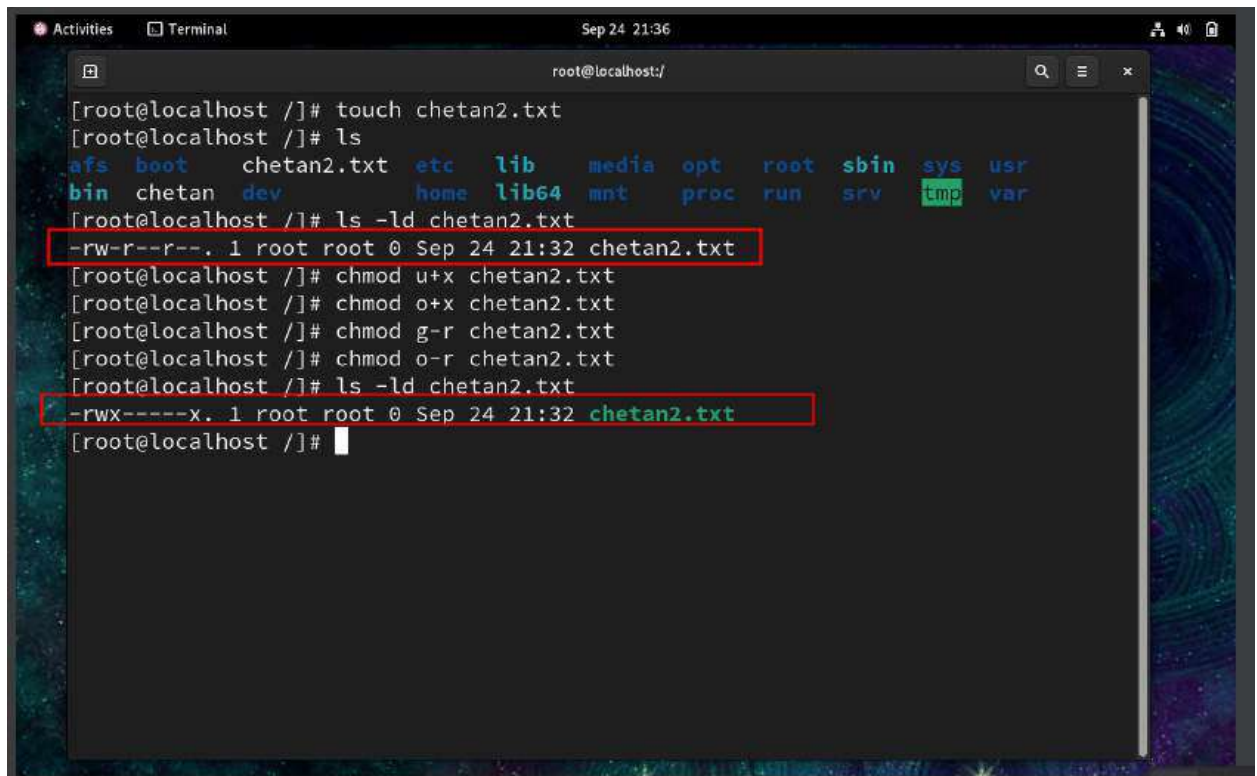
```
[root@localhost ~]# touch chetan.txt
[root@localhost ~]# ls -ld chetan.txt
-rw-r--r--. 1 root root 0 Sep 24 21:18 chetan.txt
[root@localhost ~]# ls
afs  boot  chetan.txt  etc  lib  media  opt  root  sbin  sys  usr
bin  chetan  dev      home  lib64  mnt  proc  run  srv  tmp  var
[root@localhost ~]# ls -ld chetan.txt
-rw-r--r--. 1 root root 0 Sep 24 21:18 chetan.txt
[root@localhost ~]# chmod u+x chetan.txt
[root@localhost ~]# chmod g+w chetan.txt
[root@localhost ~]# chmod g+x chetan.txt
[root@localhost ~]#
```

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

A terminal window titled 'Terminal' with a timestamp of 'Sep 24 21:28'. The prompt is 'root@localhost:/' and the terminal shows a series of commands and their outputs. The commands are: 'touch chetan.txt', 'ls -ld chetan.txt', 'ls', 'ls -ld chetan.txt', 'chmod u+x chetan.txt', 'chmod g+w chetan.txt', 'chmod g+x chetan.txt', 'ls -ld chetan.txt', and a final prompt. The outputs show the file's creation, its initial permissions (-rw-r--r--), and the directory listing. The final 'ls -ld' command shows the permissions changed to -rwxrwxr--. The terminal has a dark background with a colorful, abstract pattern on the right side.

```
[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 /]# 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 /]# ls -ld chetan.txt
-rw-r--r--. 1 root root 0 Sep 24 21:18 chetan.txt
[root@localhost /]# chmod u+x chetan.txt
[root@localhost /]# chmod g+w chetan.txt
[root@localhost /]# chmod g+x chetan.txt
[root@localhost /]# ls -ld chetan.txt
-rwxrwxr--. 1 root root 0 Sep 24 21:18 chetan.txt
[root@localhost /]#
```

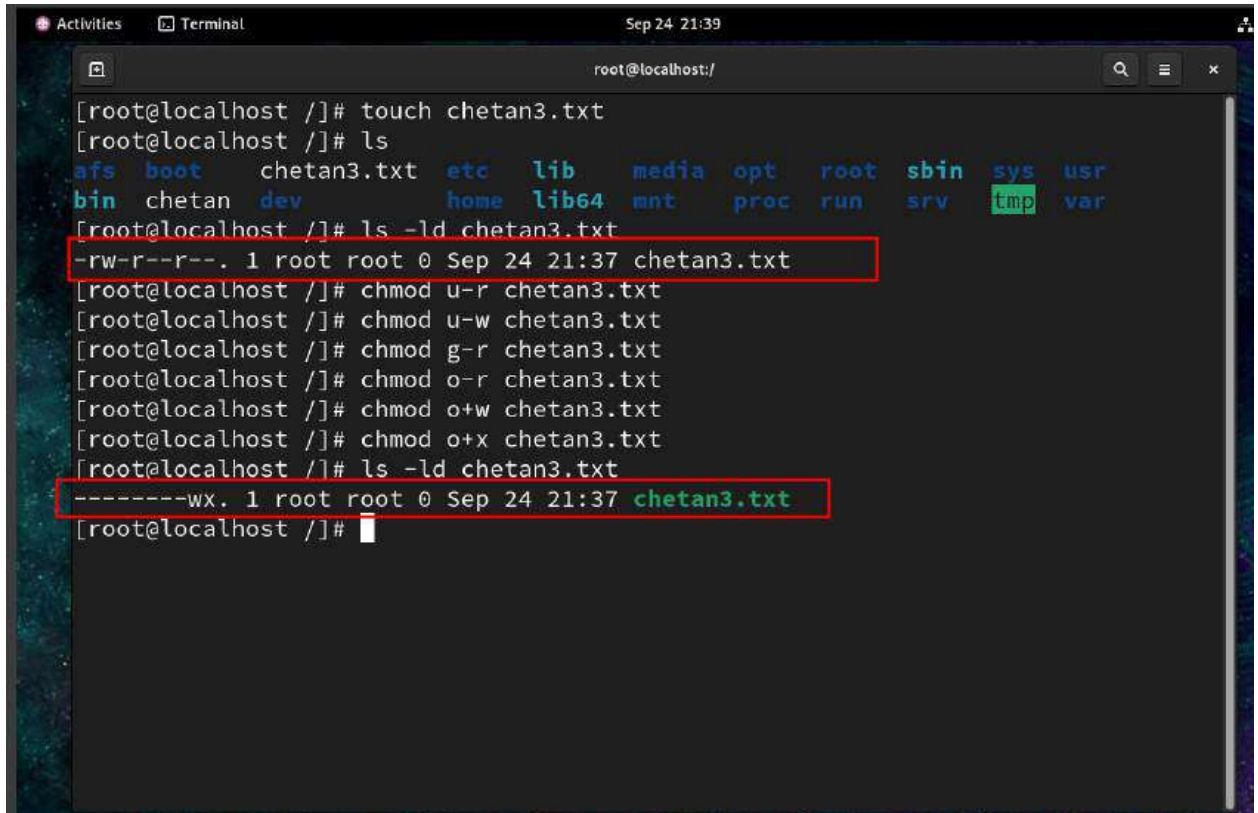

7. Change the permission to rwx____x



```
[root@localhost ~]# touch chetan2.txt
[root@localhost ~]# ls
afs  boot  chetan2.txt  etc  lib  media  opt  root  sbin  sys  usr
bin  chetan  dev  home  lib64  mnt  proc  run  srv  tmp  var
[root@localhost ~]# ls -ld chetan2.txt
-rw-r--r--. 1 root root 0 Sep 24 21:32 chetan2.txt
[root@localhost ~]# chmod u+x chetan2.txt
[root@localhost ~]# chmod o+x chetan2.txt
[root@localhost ~]# chmod g-r chetan2.txt
[root@localhost ~]# chmod o-r chetan2.txt
[root@localhost ~]# ls -ld chetan2.txt
-rwx-----x. 1 root root 0 Sep 24 21:32 chetan2.txt
[root@localhost ~]#
```

The image shows a terminal window with a dark background and a colorful, abstract pattern on the right side. The terminal output shows the creation of a file named 'chetan2.txt' and the subsequent changes to its permissions. The initial permissions are '-rw-r--r--'. The user then runs a series of 'chmod' commands to change the permissions to '-rwx-----x'. The final output shows the file 'chetan2.txt' with permissions '-rwx-----x'.

8. Change the permission to _____wx



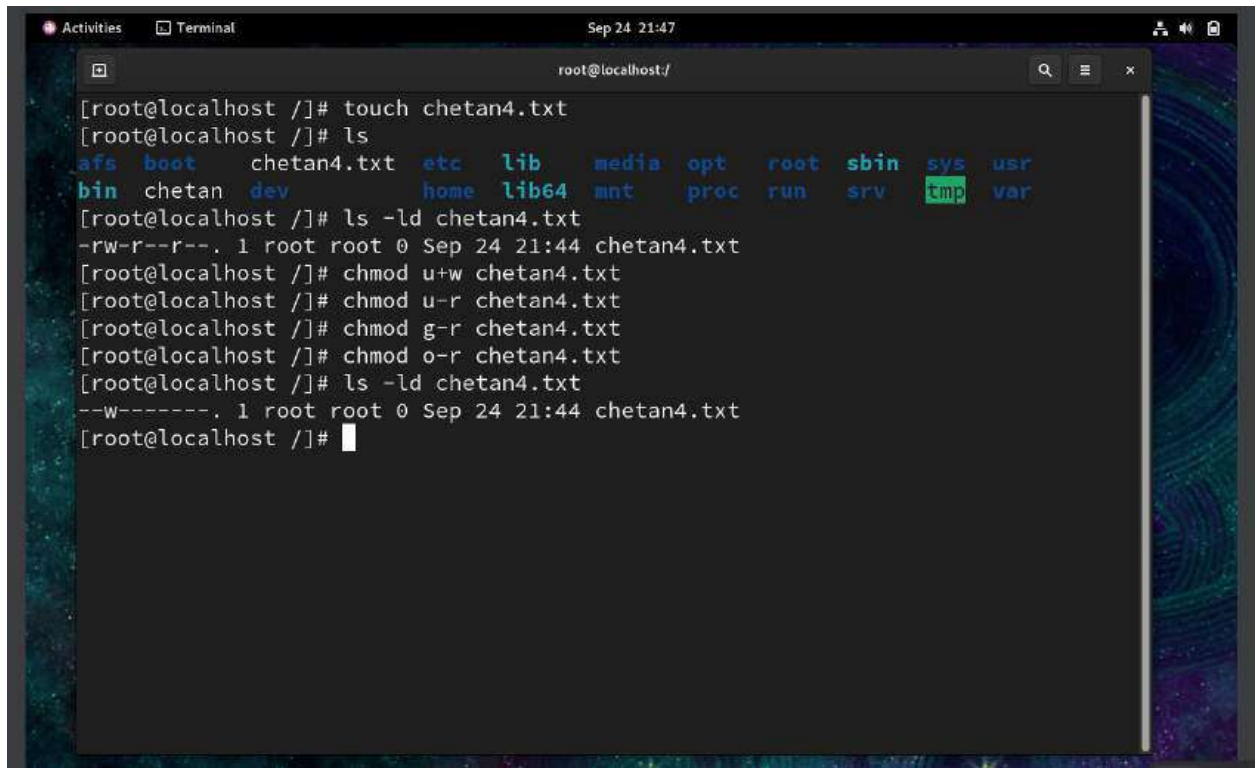
```
root@localhost: /
[root@localhost ~]# touch chetan3.txt
[root@localhost ~]# ls
afs  boot  chetan3.txt  etc  lib  media  opt  root  sbin  sys  usr
bin  chetan  dev  home  lib64  mnt  proc  run  srv  tmp  var
[root@localhost ~]# ls -ld chetan3.txt
-rw-r--r--. 1 root root 0 Sep 24 21:37 chetan3.txt
[root@localhost ~]# chmod u-r chetan3.txt
[root@localhost ~]# chmod u-w chetan3.txt
[root@localhost ~]# chmod g-r chetan3.txt
[root@localhost ~]# chmod o-r chetan3.txt
[root@localhost ~]# chmod o+w chetan3.txt
[root@localhost ~]# chmod o+x chetan3.txt
[root@localhost ~]# ls -ld chetan3.txt
-----wx. 1 root root 0 Sep 24 21:37 chetan3.txt
[root@localhost ~]#
```

The image shows a terminal window with the following commands and output:

- `touch chetan3.txt`
- `ls` (output: `afs boot chetan3.txt etc lib media opt root sbin sys usr bin chetan dev home lib64 mnt proc run srv tmp var`)
- `ls -ld chetan3.txt` (output: `-rw-r--r--. 1 root root 0 Sep 24 21:37 chetan3.txt`)
- `chmod u-r chetan3.txt`
- `chmod u-w chetan3.txt`
- `chmod g-r chetan3.txt`
- `chmod o-r chetan3.txt`
- `chmod o+w chetan3.txt`
- `chmod o+x chetan3.txt`
- `ls -ld chetan3.txt` (output: `-----wx. 1 root root 0 Sep 24 21:37 chetan3.txt`)

Red boxes highlight the initial permissions and the final permissions after the `chmod` commands.

9. Give write permission to the owner

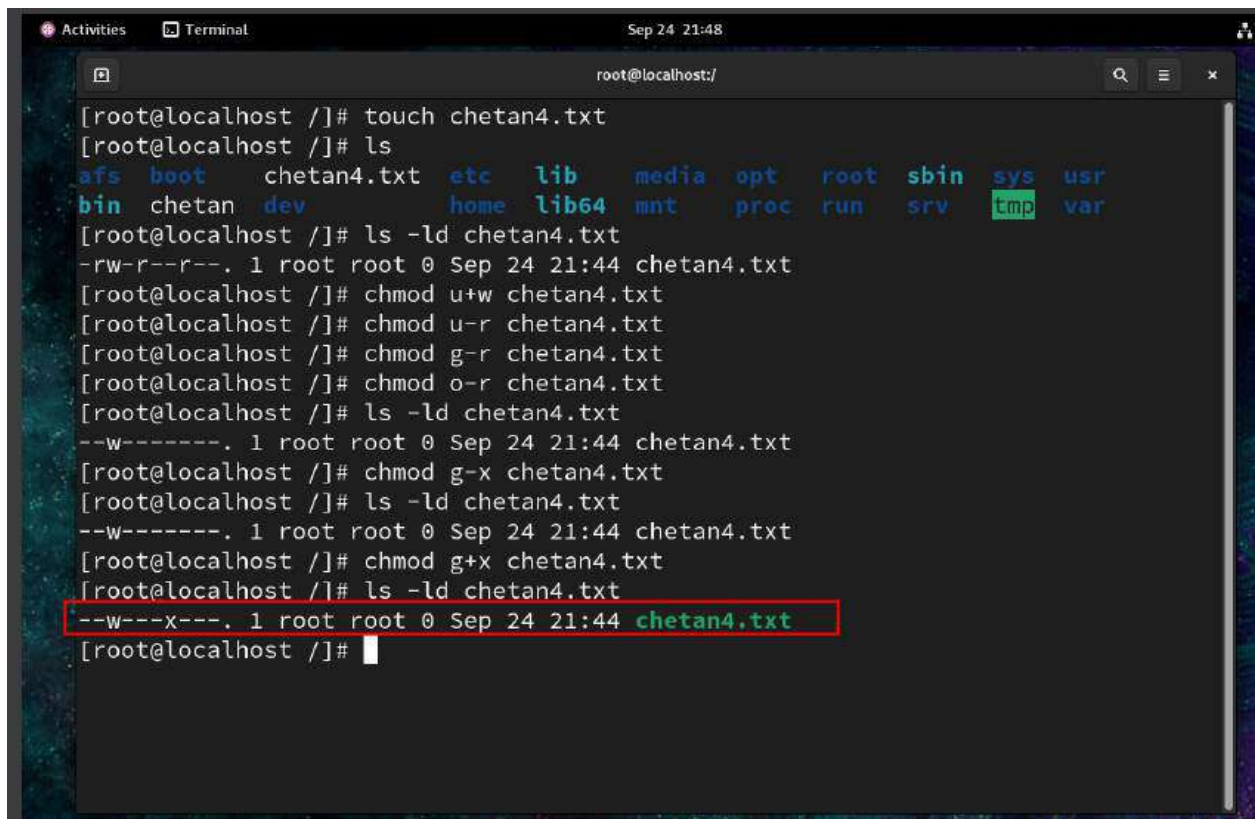


```
[root@localhost ~]# touch chetan4.txt
[root@localhost ~]# ls
afs  boot  chetan4.txt  etc  lib  media  opt  root  sbin  sys  usr
bin  chetan  dev  home  lib64  mnt  proc  run  srv  tmp  var
[root@localhost ~]# ls -ld chetan4.txt
-rw-r--r--. 1 root root 0 Sep 24 21:44 chetan4.txt
[root@localhost ~]# 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 ~]#
```

The image shows a terminal window titled "Terminal" with the date and time "Sep 24 21:47". The user is logged in as root at localhost. The terminal shows the following sequence of commands and output:

- `touch chetan4.txt`: Creates the file `chetan4.txt`.
- `ls`: Lists the contents of the current directory, showing various system directories and the newly created `chetan4.txt`.
- `ls -ld chetan4.txt`: Shows the permissions of `chetan4.txt` as `-rw-r--r--`.
- `chmod u+w chetan4.txt`: Adds write permission for the owner.
- `chmod u-r chetan4.txt`: Removes read permission for the owner.
- `chmod g-r chetan4.txt`: Removes read permission for the group.
- `chmod o-r chetan4.txt`: Removes read permission for others.
- `ls -ld chetan4.txt`: Shows the updated permissions of `chetan4.txt` as `--w-----`.

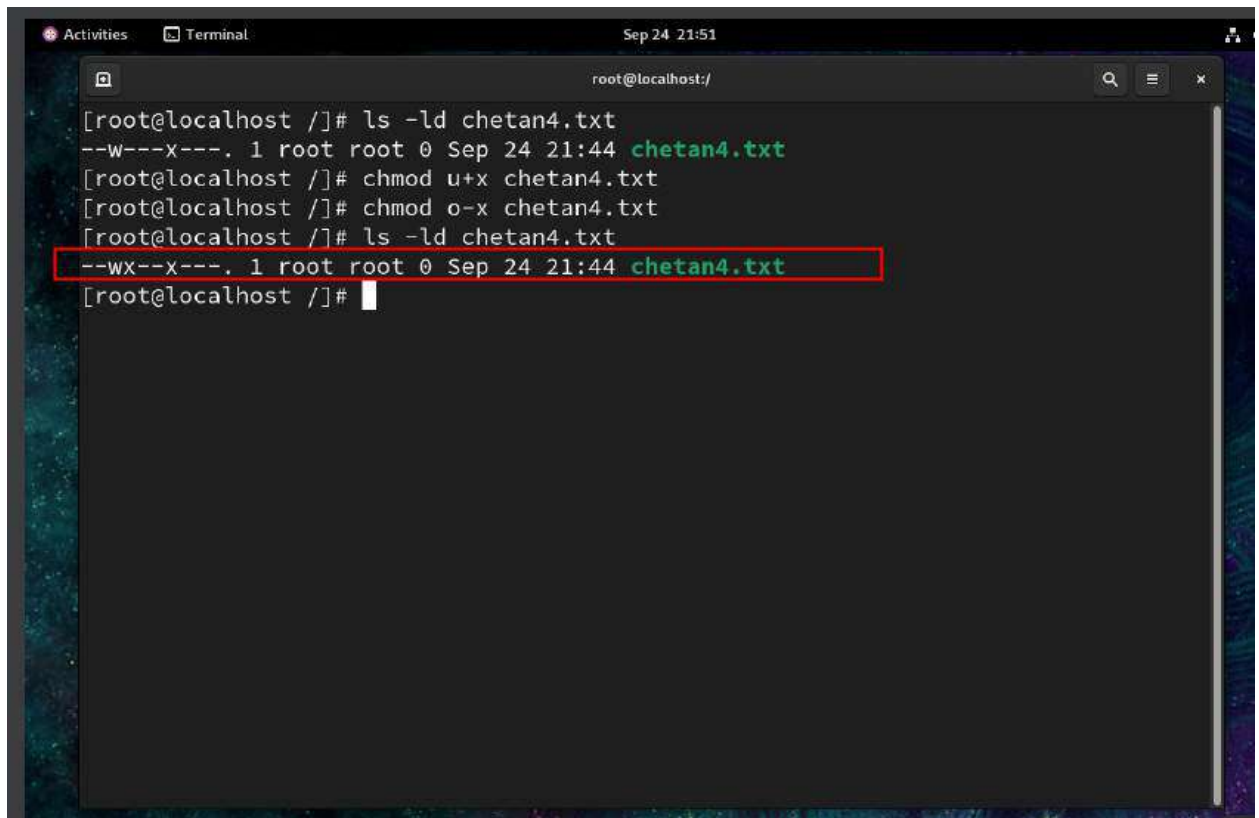
10. Give execute permission to the group owner



```
[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 ~]#
```

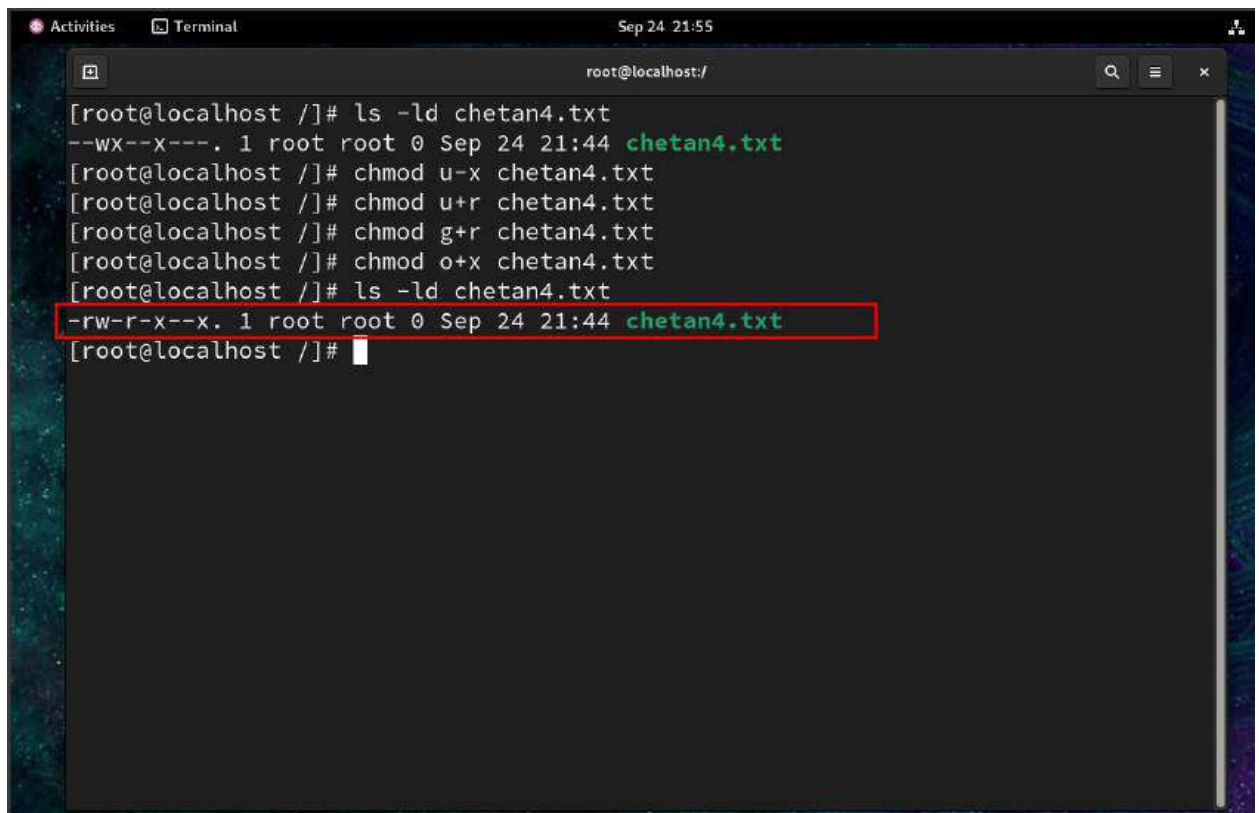
The terminal window shows the user root at localhost. The user creates a file named chetan4.txt using the touch command. Then, they use the ls command to list the files in the current directory. Next, they use the ls -ld command to show the permissions of chetan4.txt, which are -rw-r--r--. They then use the chmod command to add write permission for the user (u+w), remove read permission for the user (u-r), remove read permission for the group (g-r), and remove read permission for others (o-r). After these changes, the permissions are --w-----. They then use the chmod command to add execute permission for the group (g-x), and finally, they use the chmod command to add execute permission for the group (g+x). The final permissions are --w---x---, which is highlighted with a red box in the original image.

11. Remove execute permission to others

A terminal window titled 'Terminal' with a date and time of 'Sep 24 21:51'. The prompt is 'root@localhost:/' and the current directory is '/'. The user runs 'ls -ld chetan4.txt' showing permissions '--w---x---. 1 root root 0 Sep 24 21:44 chetan4.txt'. Then they run 'chmod u+x chetan4.txt' and 'chmod o-x chetan4.txt'. Finally, they run 'ls -ld chetan4.txt' again, and the output '--wx--x---. 1 root root 0 Sep 24 21:44 chetan4.txt' is highlighted with a red rectangle.

```
[root@localhost /]# ls -ld chetan4.txt
--w---x---. 1 root root 0 Sep 24 21:44 chetan4.txt
[root@localhost /]# chmod u+x chetan4.txt
[root@localhost /]# chmod o-x chetan4.txt
[root@localhost /]# ls -ld chetan4.txt
--wx--x---. 1 root root 0 Sep 24 21:44 chetan4.txt
[root@localhost /]#
```


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

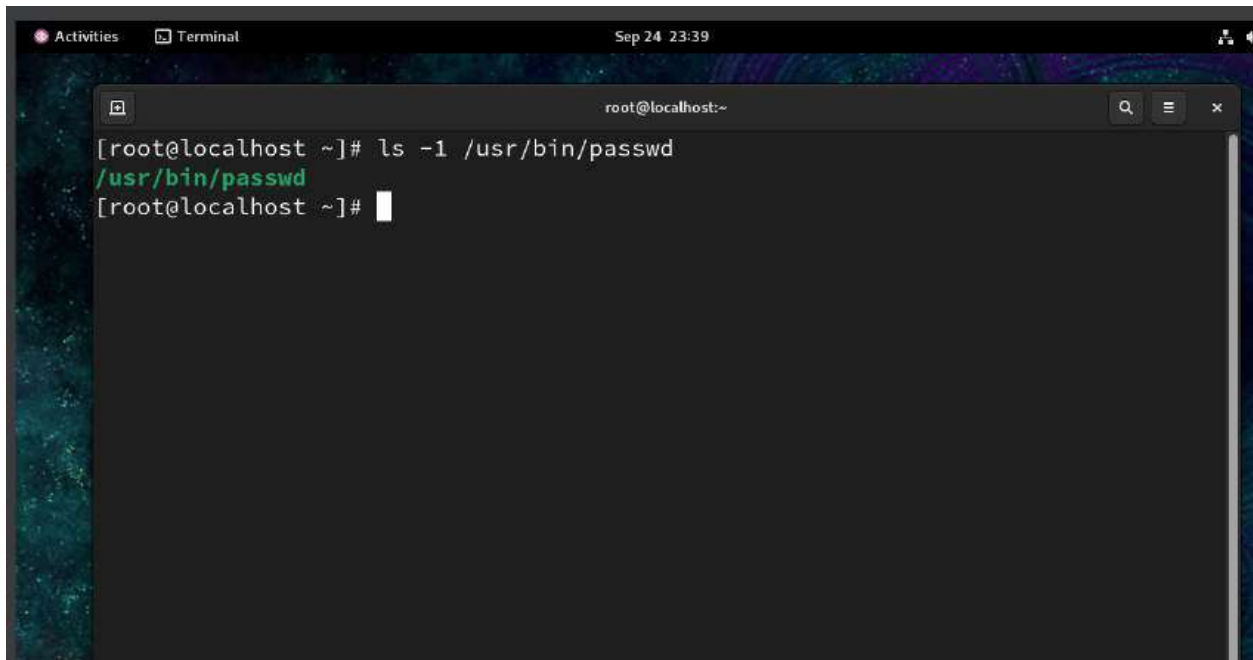


A terminal window titled "Terminal" with a timestamp of "Sep 24 21:55". The prompt is "root@localhost:/". The user enters the command "ls -ld chetan4.txt", which returns "--wx--x---. 1 root root 0 Sep 24 21:44 chetan4.txt". The user then enters a series of "chmod" commands: "chmod u-x chetan4.txt", "chmod u+r chetan4.txt", "chmod g+r chetan4.txt", and "chmod o+x chetan4.txt". Finally, the user enters "ls -ld chetan4.txt" again, and the output "-rw-r-x--x. 1 root root 0 Sep 24 21:44 chetan4.txt" is highlighted with a red rectangular box.

```
[root@localhost /]# ls -ld chetan4.txt
--wx--x---. 1 root root 0 Sep 24 21:44 chetan4.txt
[root@localhost /]# chmod u-x chetan4.txt
[root@localhost /]# chmod u+r chetan4.txt
[root@localhost /]# chmod g+r chetan4.txt
[root@localhost /]# chmod o+x chetan4.txt
[root@localhost /]# ls -ld chetan4.txt
-rw-r-x--x. 1 root root 0 Sep 24 21:44 chetan4.txt
[root@localhost /]#
```

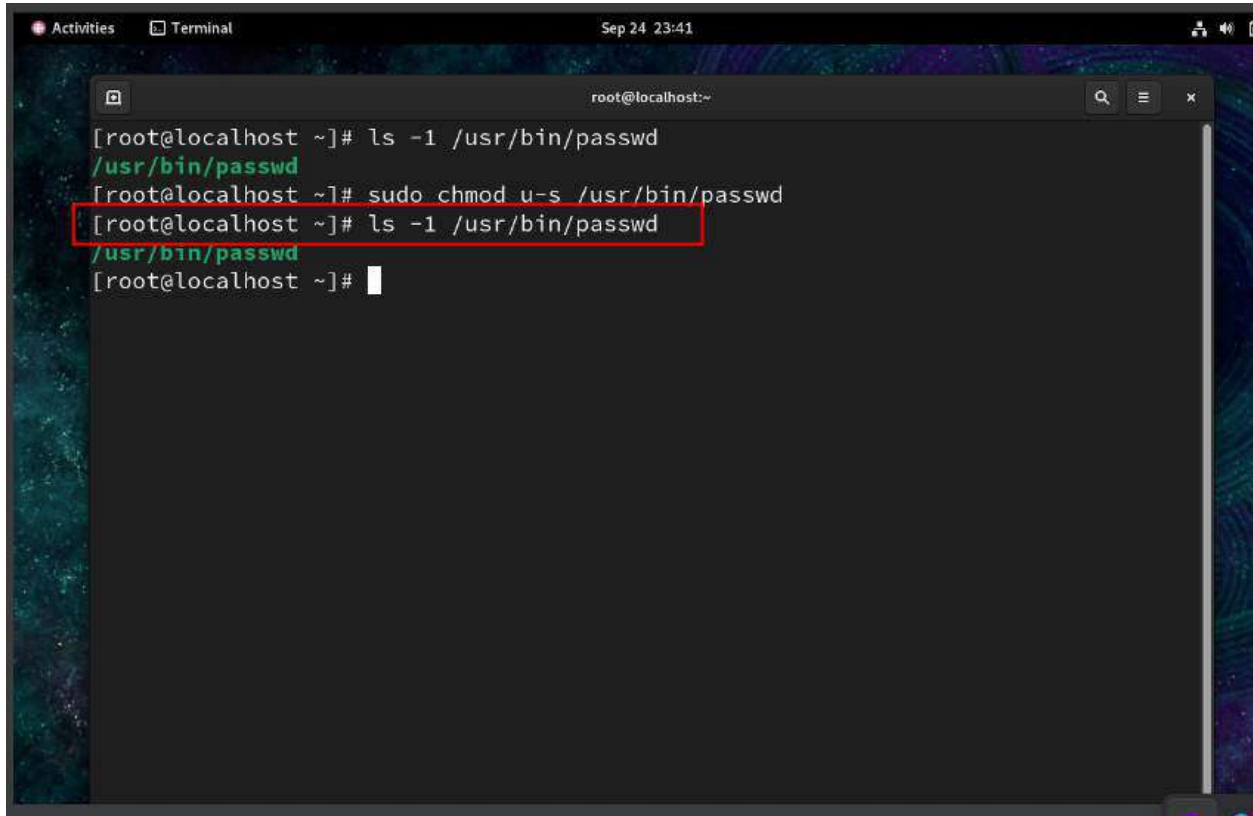
Q. 5. Remove suid permission from passwd command on your Linux system and try changing your password.
Give back the suid permission to passwd command and try changing your password.

1. Checking the current permission of the passwd command

A screenshot of a Linux terminal window. The window title bar shows 'Activities', 'Terminal', and the date 'Sep 24 23:39'. The terminal prompt is '[root@localhost ~]#'. The user has entered the command 'ls -l /usr/bin/passwd'. The output is displayed in green text: '-rwsr-xr-x 1 root root 1460000 Sep 24 23:39 /usr/bin/passwd'. The prompt is now '[root@localhost ~]#'.

```
[root@localhost ~]# ls -l /usr/bin/passwd
-rwsr-xr-x 1 root root 1460000 Sep 24 23:39 /usr/bin/passwd
[root@localhost ~]#
```

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

A terminal window titled "root@localhost:~" with a search icon, menu icon, and close button in the top right. The terminal shows the following commands and output:

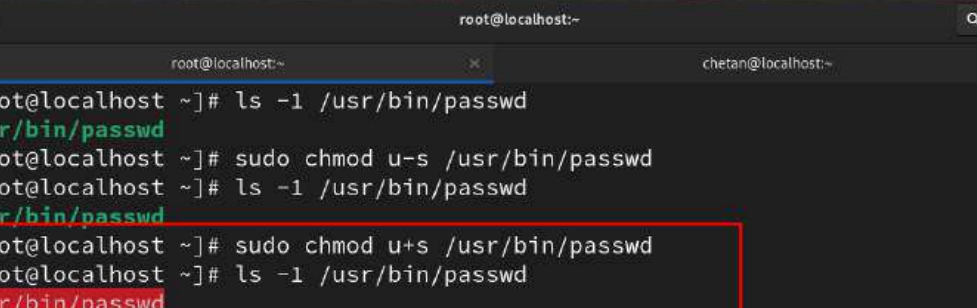
```
[root@localhost ~]# ls -l /usr/bin/passwd
-rwsr-xr-x 1 root root 144400 Sep 24 23:41 /usr/bin/passwd
[root@localhost ~]# sudo chmod u-s /usr/bin/passwd
[root@localhost ~]# ls -l /usr/bin/passwd
-rwxr-xr-x 1 root root 144400 Sep 24 23:41 /usr/bin/passwd
[root@localhost ~]#
```

The third line, showing the file permissions after the command, is highlighted with a red rectangle. The window's top bar includes "Activities", "Terminal", and a timestamp "Sep 24 23:41". The desktop background is a dark, abstract pattern.

3. Try to change the passwd but it show error

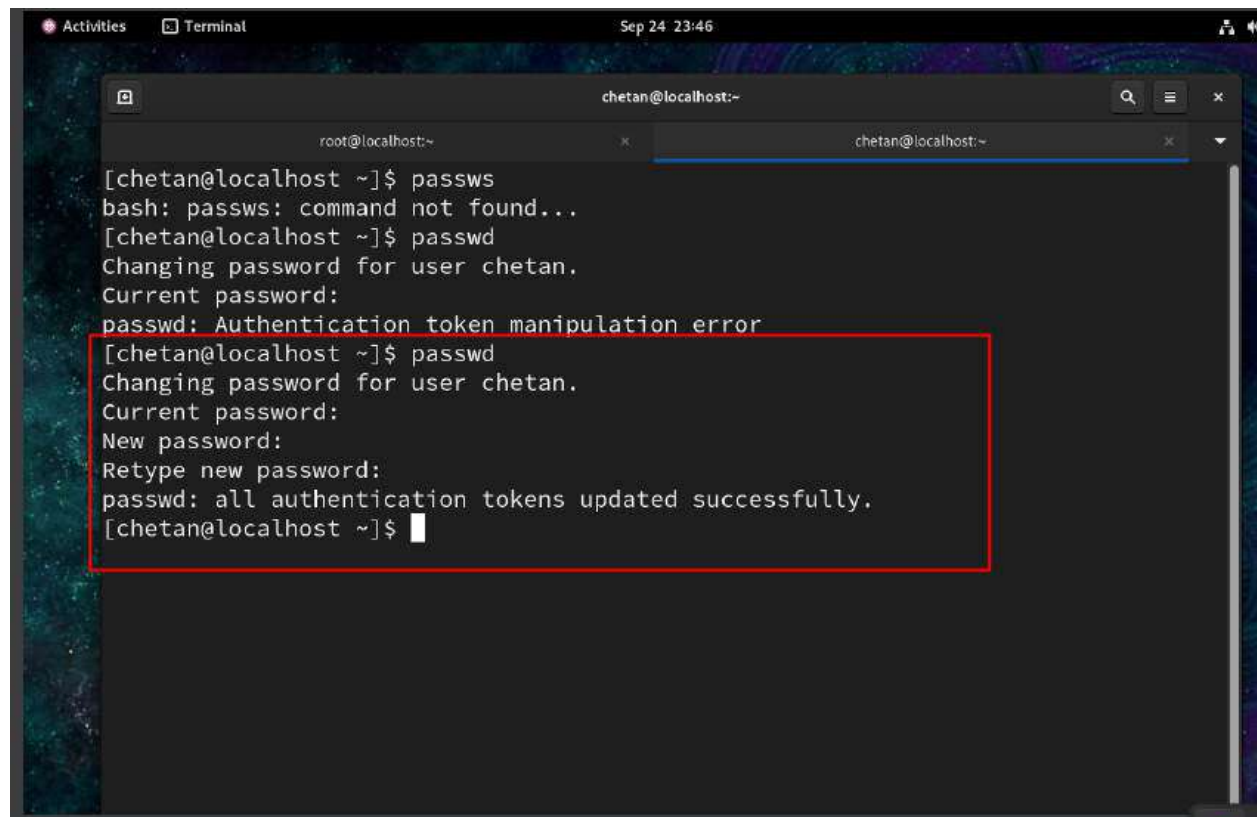
```
[chetan@localhost ~]$ passwd
Changing password for user chetan.
Current password:
passwd: Authentication token manipulation error
[chetan@localhost ~]$
```

4. Change to default setting



```
root@localhost:~  
[root@localhost ~]# ls -l /usr/bin/passwd  
-rwxr-xr-x 1 root root 1444000000  
[root@localhost ~]# sudo chmod u-s /usr/bin/passwd  
[root@localhost ~]# ls -l /usr/bin/passwd  
-rwxr-xr-x 1 root root 1444000000  
[root@localhost ~]# sudo chmod u+s /usr/bin/passwd  
[root@localhost ~]# ls -l /usr/bin/passwd  
-rwxr-xr-x 1 root root 1444000000  
[root@localhost ~]#
```

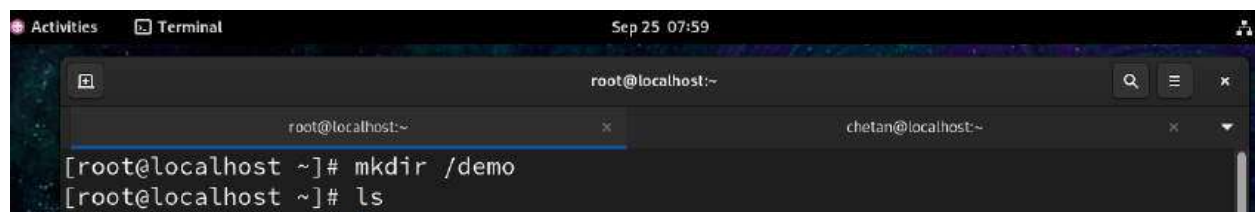

5. Now we can change the passwd by the user

A screenshot of a Linux terminal window. The window title is "chetan@localhost:~". The terminal shows the user "chetan" attempting to change their password. They first run "passwd", which fails with "Authentication token manipulation error". They then run "passwd" again, which succeeds. The successful sequence is highlighted with a red rectangle.

```
chetan@localhost:~  
[chetan@localhost ~]$ passws  
bash: passws: command not found...  
[chetan@localhost ~]$ passwd  
Changing password for user chetan.  
Current password:  
passwd: Authentication token manipulation error  
[chetan@localhost ~]$ passwd  
Changing password for user chetan.  
Current password:  
New password:  
Retype new password:  
passwd: all authentication tokens updated successfully.  
[chetan@localhost ~]$
```

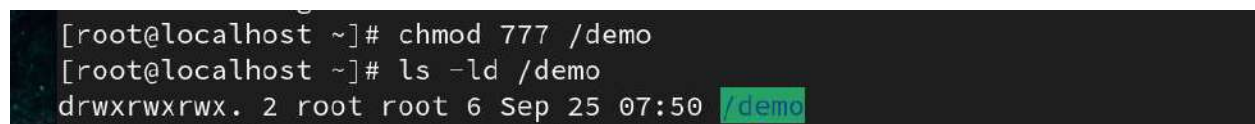
6. Create /demo folder as root, give 777 permission to /demo folder and login as normal user and try creating files and folder in /demo folder. Now add sgid permission to the /demo folder using root. Again try creating files and folder as normal user in /demo folder. Write the observations.

1. Login as root user, make directory named demo



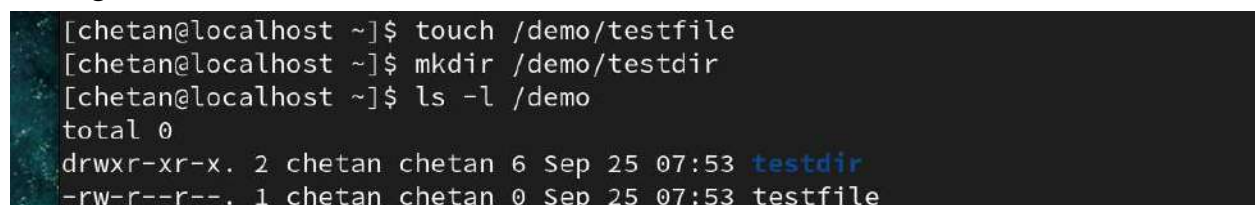
```
Activities Terminal Sep 25 07:59
root@localhost:~
root@localhost:~
[root@localhost ~]# mkdir /demo
[root@localhost ~]# ls
```

2. Given 777 permission to demo folder



```
[root@localhost ~]# chmod 777 /demo
[root@localhost ~]# ls -ld /demo
drwxrwxrwx. 2 root root 6 Sep 25 07:50 /demo
```

3. Login as normal user and created a file



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

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 bit 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

```
john@localhost:~  
[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.