Working with a Vi Editor:

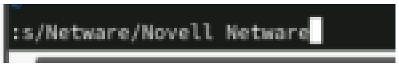
1: Create a file using vi. Enter the following text:

A network is a group of computers that can communicate with each other, share resources, and access remote hosts or other networks. Netware is a computer network operating system designed to connect, manage, and maintain a network and its services. Some of the network services are Netware Directory Services (NDS), file system, printing and security.

a. Change the word "Netware" in the second line to "Novell Netware".

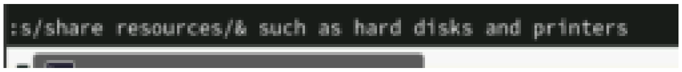
To the end of vim file in command mode:

:s/Netware/Novell Netware/



```
:s/Netware/Novell Netware
```

b. Insert the text "(such as hard disks and printers)" after "share resources" in the first line.



```
:s/share resources/& such as hard disks and printers
```

c. Append the following text to the file:

"Managing NDS is a fundamental administrator role because NDS provides a single point for accessing and managing most network resources."

Ans:

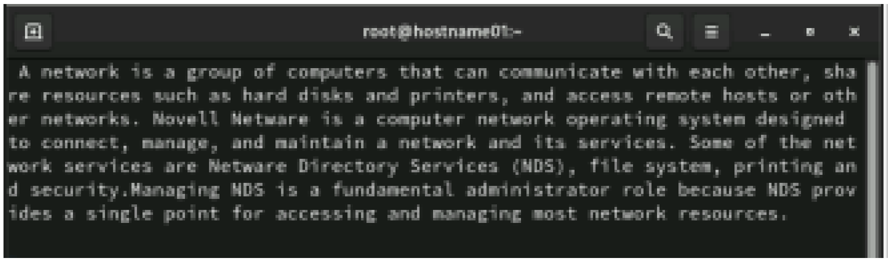
1.Open the vim file

2.Move to the end of file using Shift+G.

3.Press A to append to the end of current line or o to open new line below current line.

4.Add the text or copy paste it.

5.Press esc to return and then save the file using :wq



```
 A network is a group of computers that can communicate with each other, sha
re resources such as hard disks and printers, and access remote hosts or oth
er networks. Novell Netware is a computer network operating system designed
to connect, manage, and maintain a network and its services. Some of the net
work services are Netware Directory Services (NDS), file system, printing an
d security.Managing NDS is a fundamental administrator role because NDS prov
ides a single point for accessing and managing most network resources.
```

Working shell

1. Type some text on the shell separated by space

1: Move cursor one word back

Ans: Ctrl+Left Arrow or b in vim

2: Move cursor one word forward

Ans: Ctrl+Right Arrow or w in vim

3: Move cursor to the first character

Ans: Ctrl+a

4: Move cursor to the end

Ans: **Ctrl+e**

5: Delete test from second word to last character

Ans: To delete text from the second word to the last character in a line in Vim, you can use the following command in normal mode:

1. **Place the cursor on the first word of the line.**
2. **Enter normal mode by pressing Esc.**
3. **Use the command dw$**

6: Delete the current line

Ans: **Press dd**

2: In lab 4 we have created a file errorlog.txt. Display it using cat command using command completion.

Ans:

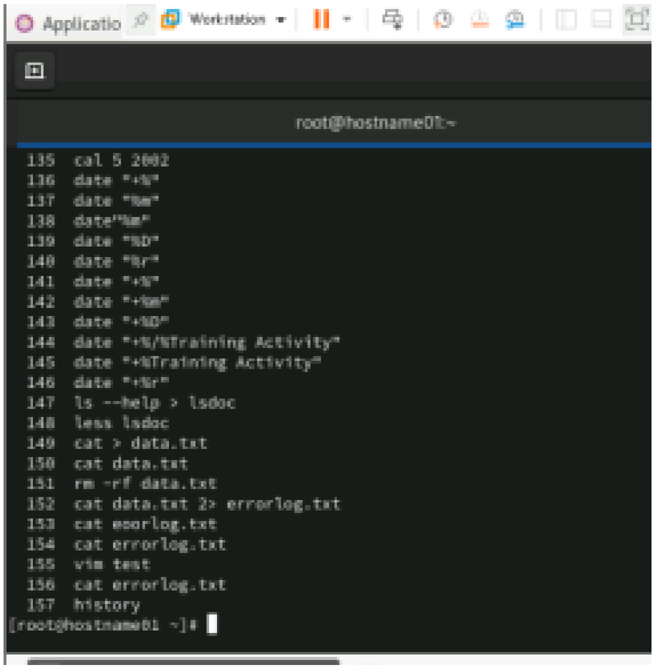
To display the contents of the errorlog.txt file using the cat command with command completion, follow these steps:

1. Open your terminal.
2. Start typing the command:
3. **cat err**
4. Press the Tab key to auto-complete the filename. If there are multiple files starting with "err," press Tab again to cycle through the options until errorlog.txt is selected.
5. Press Enter.

The full command should look like this:

cat errorlog.txt

3: Display history of command used so far.



root@hostname01:~

```
  135  cal 5 2002
  136  date "+%"
  137  date "%m"
  138  date"%m"
  139  date "%D"
  140  date "%r"
  141  date "+%"
  142  date "+%m"
  143  date "+%D"
  144  date "+%/%Training Activity"
  145  date "+%Training Activity"
  146  date "+%r"
  147  ls --help > lsdoc
  148  less lsdoc
  149  cat > data.txt
  150  cat data.txt
  151  rm -rf data.txt
  152  cat data.txt 2> errorlog.txt
  153  cat eoorlog.txt
  154  cat errorlog.txt
  155  vim test
  156  cat errorlog.txt
  157  history
[root@hostname01 ~]#
```

4: Search ls command in history file

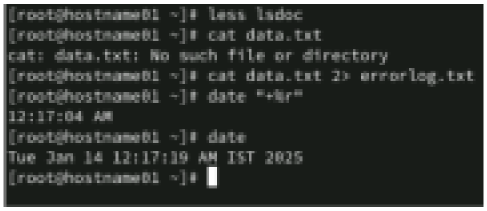
Ans: To search for the ls command in your history file, you can use the history command along with grep. Here's how you can do it:

1. **Open your terminal.**
2. **Type the following command and press Enter:**

history | grep ls

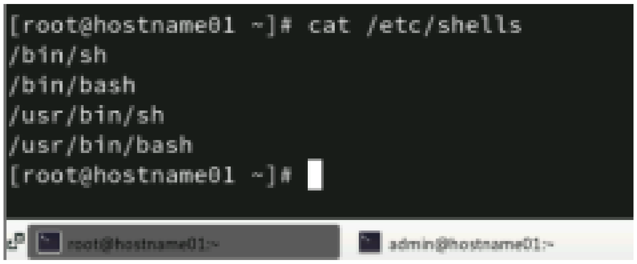
5: Repeat the last command rd

6: Execute 3   command from history file.



```
[root@hostname01 ~]# less lsdoc
[root@hostname01 ~]# cat data.txt
cat: data.txt: No such file or directory
[root@hostname01 ~]# cat data.txt 2> errorlog.txt
[root@hostname01 ~]# date "+%r"
12:17:04 AM
[root@hostname01 ~]# date
Tue Jan 14 12:17:19 AM IST 2025
[root@hostname01 ~]#
```

7: What are the different shells available.



```
[root@hostname01 ~]# cat /etc/shells
/bin/sh
/bin/bash
/usr/bin/sh
/usr/bin/bash
[root@hostname01 ~]#
```

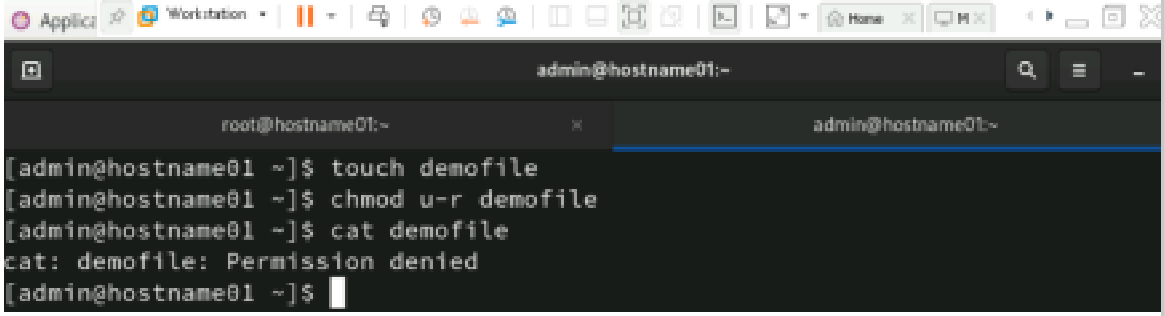
Linux offers several different types of shells, each with unique features and functionalities. Here are some of the most commonly used shells:

1. **Bourne Shell (sh)**: The original Unix shell developed at AT&T Bell Labs. It's known for its simplicity and speed but lacks some advanced features like command history and arithmetic operations[1].
2. **Bourne-Again Shell (bash)**: An enhanced version of the Bourne Shell, bash is the default shell on many Linux distributions. It includes features like command history, job control, and shell functions[1].
3. **C Shell (csh)**: Designed to improve interactive use, the C Shell incorporates syntax similar to the C programming language. It includes features like command history and aliases[2].
4. **Korn Shell (ksh)**: Combines features of the Bourne Shell and the C Shell. It offers advanced scripting capabilities and is compatible with many Unix systems[2].
5. **Z Shell (zsh)**: An extended Bourne Shell with many improvements, including better scripting capabilities, spell checking, and theme support[2].
6. **Fish Shell (fish)**: Known for its user-friendly features, such as syntax highlighting, autosuggestions, and an easy-to-use scripting language[3].

Understanding access permissions

7.1: Create an empty file "demofile" and perform following instruction

1. Revoke read permission from owner and use cat command.

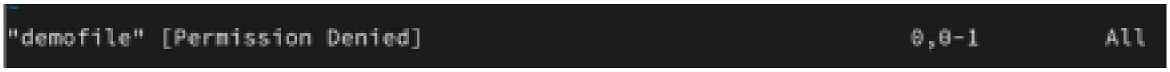


```
[admin@hostname01 ~]$ touch demofile
[admin@hostname01 ~]$ chmod u-r demofile
[admin@hostname01 ~]$ cat demofile
cat: demofile: Permission denied
[admin@hostname01 ~]$ 
```

2. Revoke write permission from owner and open using vi

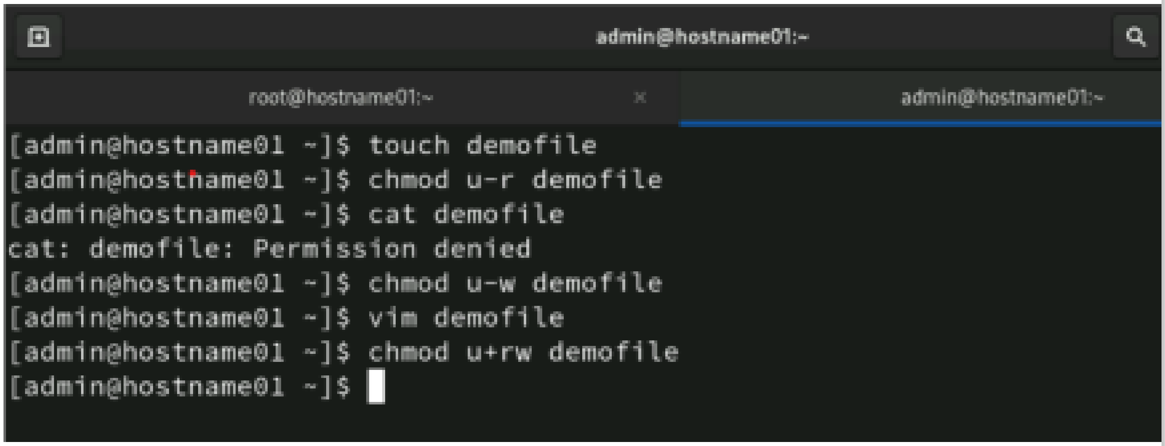
editor and add some contain in it.

Ans: chmod u-w demofile



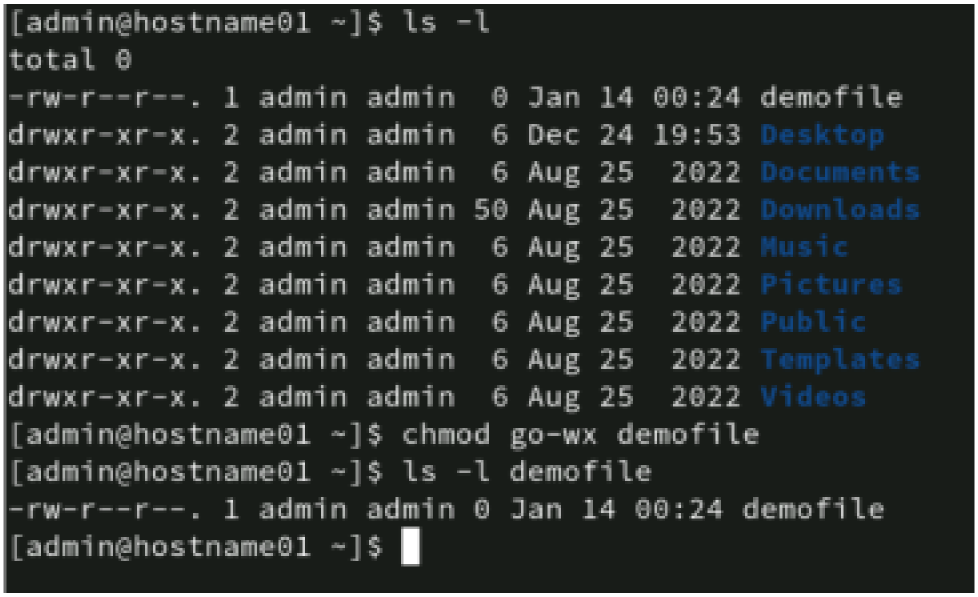
```
"demofile" [Permission Denied]                          0,0-1         All
```

3.Add read and write permission to owner.



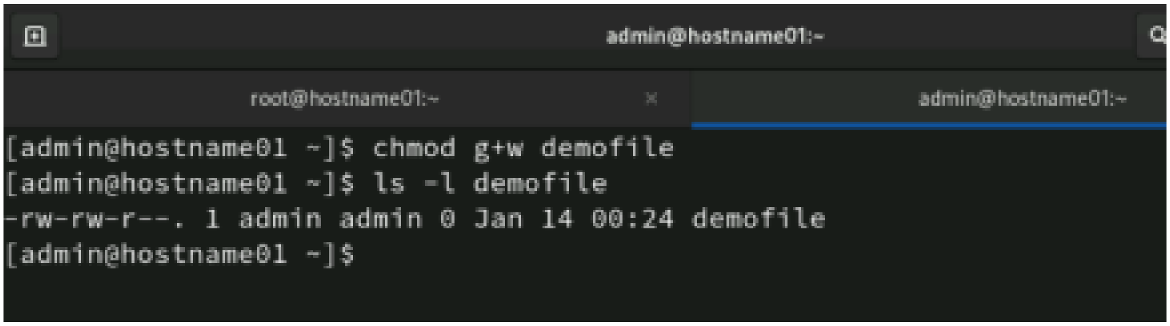
```
[admin@hostname01 ~]$ touch demofile
[admin@hostname01 ~]$ chmod u-r demofile
[admin@hostname01 ~]$ cat demofile
cat: demofile: Permission denied
[admin@hostname01 ~]$ chmod u-w demofile
[admin@hostname01 ~]$ vim demofile
[admin@hostname01 ~]$ chmod u+rw demofile
[admin@hostname01 ~]$ 
```

4. Revoke write and execute from other and group



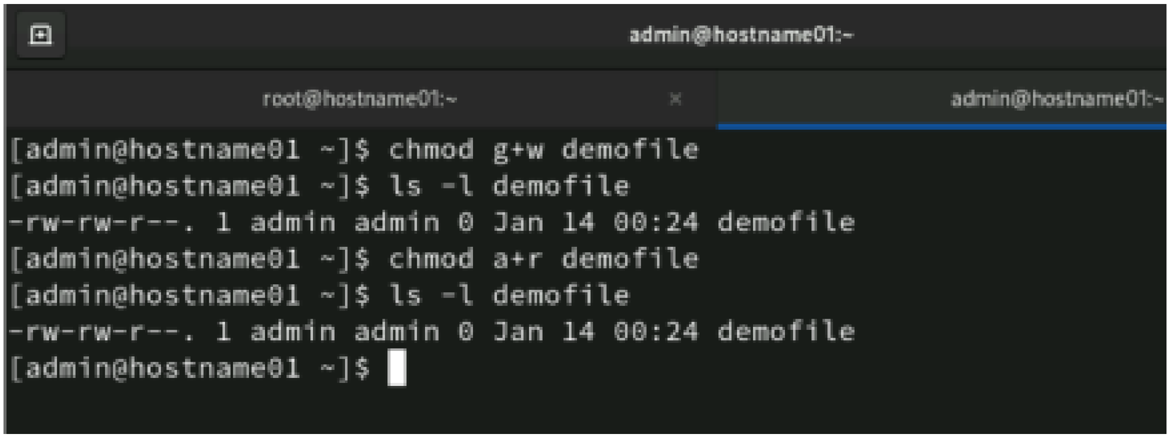
```
[admin@hostname01 ~]$ ls -l
total 0
-rw-r--r--. 1 admin admin  0 Jan 14 00:24 demofile
drwxr-xr-x. 2 admin admin  6 Dec 24 19:53 Desktop
drwxr-xr-x. 2 admin admin  6 Aug 25  2022 Documents
drwxr-xr-x. 2 admin admin 50 Aug 25  2022 Downloads
drwxr-xr-x. 2 admin admin  6 Aug 25  2022 Music
drwxr-xr-x. 2 admin admin  6 Aug 25  2022 Pictures
drwxr-xr-x. 2 admin admin  6 Aug 25  2022 Public
drwxr-xr-x. 2 admin admin  6 Aug 25  2022 Templates
drwxr-xr-x. 2 admin admin  6 Aug 25  2022 Videos
[admin@hostname01 ~]$ chmod go-wx demofile
[admin@hostname01 ~]$ ls -l demofile
-rw-r--r--. 1 admin admin 0 Jan 14 00:24 demofile
[admin@hostname01 ~]$ 
```

5. Add write permission to group only



```
[admin@hostname01 ~]$ chmod g+w demofile
[admin@hostname01 ~]$ ls -l demofile
-rw-rw-r--. 1 admin admin 0 Jan 14 00:24 demofile
[admin@hostname01 ~]$
```

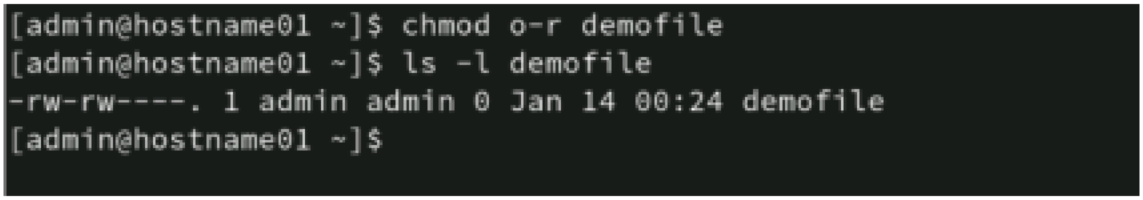
6. Assign read permission to all



```
[admin@hostname01 ~]$ chmod g+w demofile
[admin@hostname01 ~]$ ls -l demofile
-rw-rw-r--. 1 admin admin 0 Jan 14 00:24 demofile
[admin@hostname01 ~]$ chmod a+r demofile
[admin@hostname01 ~]$ ls -l demofile
-rw-rw-r--. 1 admin admin 0 Jan 14 00:24 demofile
[admin@hostname01 ~]$ 
```

C2 General

7. Revoke read permission from others



```
[admin@hostname01 ~]$ chmod o-r demofile
[admin@hostname01 ~]$ ls -l demofile
-rw-rw----. 1 admin admin 0 Jan 14 00:24 demofile
[admin@hostname01 ~]$
```

8. Give the execute permission for the user for a file chap1

Ans: chmod u+x chap1

9. Give the execute permission for user, group and others for a file add.c

Ans: chmod a+x add.c

10. Remove the execute permission from user, give read permission to group and others for a file aa.c

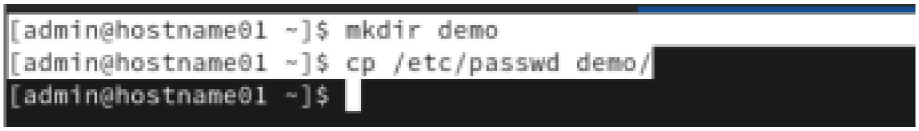
Ans: chmod u-x,go+r aa.c

ls -l aa.c

11. Give execute permission for users for a.c, kk.c, nato and myfile using single command

Ans: chmod u+x a.c kk.c nato myfile

7.2: Create an directory "demo" and copy /etc/passwd file in it



```
[admin@hostname01 ~]$ mkdir demo
[admin@hostname01 ~]$ cp /etc/passwd demo/
[admin@hostname01 ~]$
```

1. Display contents of demo
   Ans: ls demo

2. Revoke read permission from demo directory and use ls command on it

   Ans: chmod -r demo

   ls demo

3. Revoke write permission from demo directory and try to copy

   /etc/profile file in it

Ans: chmod -w demo

cp /etc/profile demo/

4. Delete passwd file from demo directory

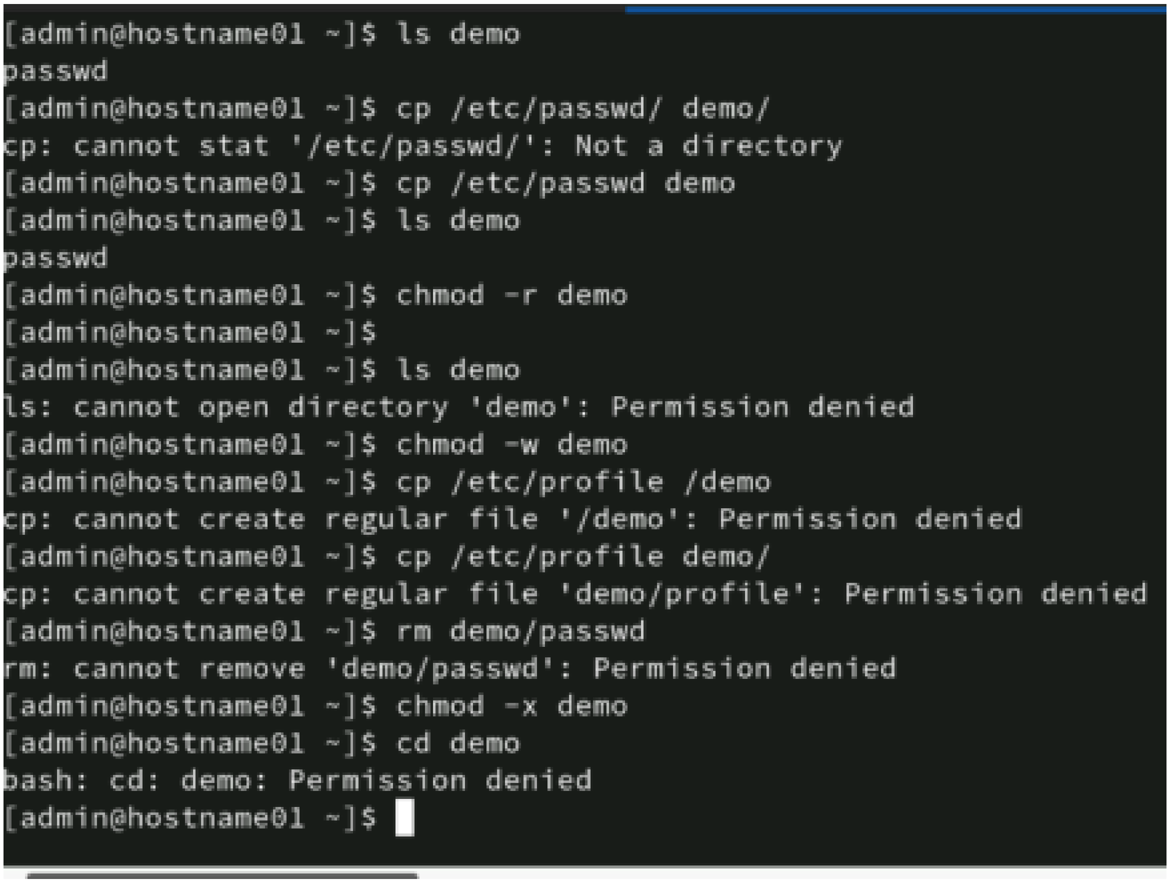
Ans: rm demo/passwd

5. Revoke execute permission from demo directory and try cd

   command on demo.

Ans: chmod -x demo

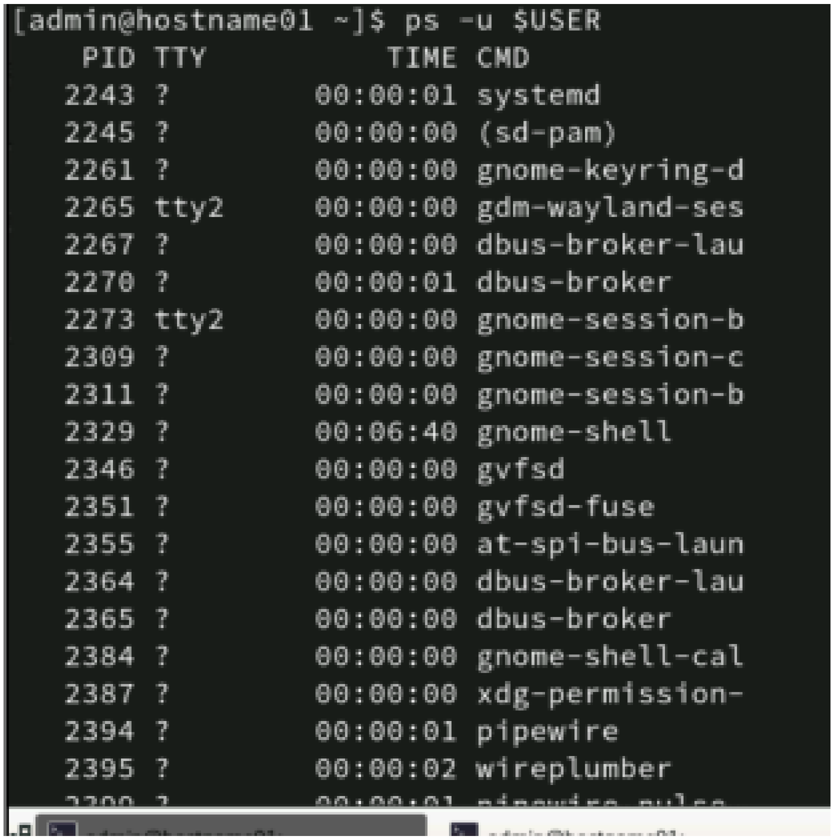
cd demo



```
[admin@hostname01 ~]$ ls demo
passwd
[admin@hostname01 ~]$ cp /etc/passwd/ demo/
cp: cannot stat '/etc/passwd/': Not a directory
[admin@hostname01 ~]$ cp /etc/passwd demo
[admin@hostname01 ~]$ ls demo
passwd
[admin@hostname01 ~]$ chmod -r demo
[admin@hostname01 ~]$
[admin@hostname01 ~]$ ls demo
ls: cannot open directory 'demo': Permission denied
[admin@hostname01 ~]$ chmod -w demo
[admin@hostname01 ~]$ cp /etc/profile /demo
cp: cannot create regular file '/demo': Permission denied
[admin@hostname01 ~]$ cp /etc/profile demo/
cp: cannot create regular file 'demo/profile': Permission denied
[admin@hostname01 ~]$ rm demo/passwd
rm: cannot remove 'demo/passwd': Permission denied
[admin@hostname01 ~]$ chmod -x demo
[admin@hostname01 ~]$ cd demo
bash: cd: demo: Permission denied
[admin@hostname01 ~]$ 
```

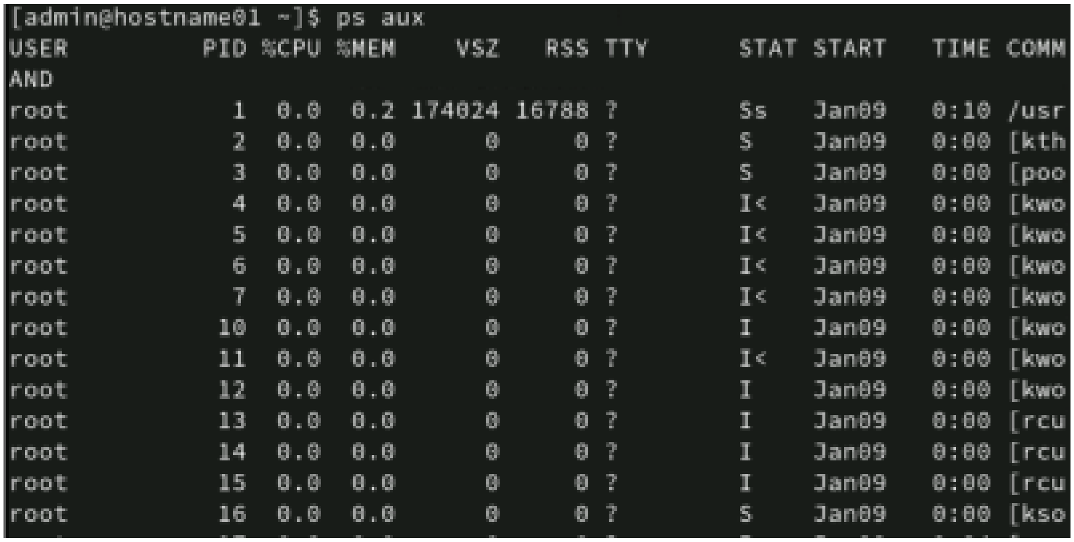
Using Process-Related Commands

1. Find out the PID of the processes that are activated by you



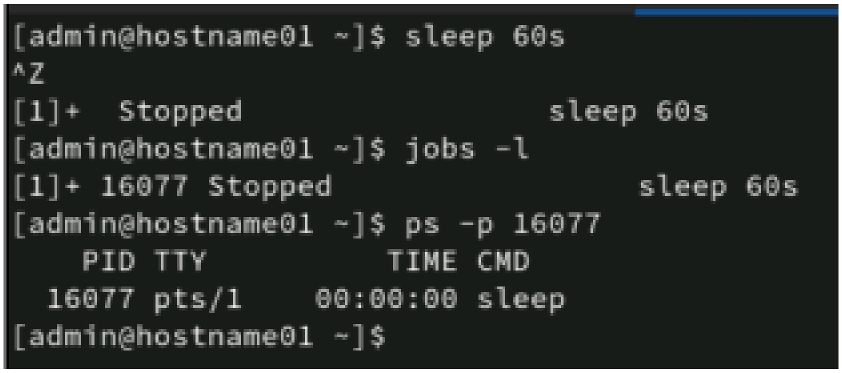
```
[admin@hostname01 ~]$ ps -u $USER
    PID TTY          TIME CMD
   2243 ?        00:00:01 systemd
   2245 ?        00:00:00 (sd-pam)
   2261 ?        00:00:00 gnome-keyring-d
   2265 tty2     00:00:00 gdm-wayland-ses
   2267 ?        00:00:00 dbus-broker-lau
   2270 ?        00:00:01 dbus-broker
   2273 tty2     00:00:00 gnome-session-b
   2309 ?        00:00:00 gnome-session-c
   2311 ?        00:00:00 gnome-session-b
   2329 ?        00:06:40 gnome-shell
   2346 ?        00:00:00 gvfsd
   2351 ?        00:00:00 gvfsd-fuse
   2355 ?        00:00:00 at-spi-bus-laun
   2364 ?        00:00:00 dbus-broker-lau
   2365 ?        00:00:00 dbus-broker
   2384 ?        00:00:00 gnome-shell-cal
   2387 ?        00:00:00 xdg-permission-
   2394 ?        00:00:01 pipewire
   2395 ?        00:00:02 wireplumber
```

2. Find out the information about all the processes that are currently active



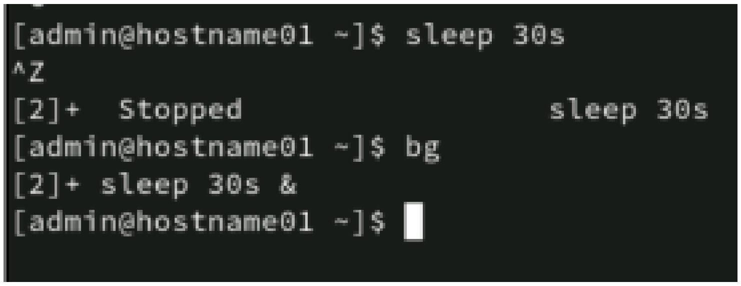
```
[admin@hostname01 ~]$ ps aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMM
AND
root           1  0.0  0.2 174024 16788 ?        Ss   Jan09   0:10 /usr
root           2  0.0  0.0      0     0 ?        S    Jan09   0:00 [kth
root           3  0.0  0.0      0     0 ?        S    Jan09   0:00 [poo
root           4  0.0  0.0      0     0 ?        I<   Jan09   0:00 [kwo
root           5  0.0  0.0      0     0 ?        I<   Jan09   0:00 [kwo
root           6  0.0  0.0      0     0 ?        I<   Jan09   0:00 [kwo
root           7  0.0  0.0      0     0 ?        I<   Jan09   0:00 [kwo
root          10  0.0  0.0      0     0 ?        I    Jan09   0:00 [kwo
root          11  0.0  0.0      0     0 ?        I<   Jan09   0:00 [kwo
root          12  0.0  0.0      0     0 ?        I    Jan09   0:00 [kwo
root          13  0.0  0.0      0     0 ?        I    Jan09   0:00 [rcu
root          14  0.0  0.0      0     0 ?        I    Jan09   0:00 [rcu
root          15  0.0  0.0      0     0 ?        I    Jan09   0:00 [rcu
root          16  0.0  0.0      0     0 ?        S    Jan09   0:00 [kso
```

3. Start a different process in the background. Find out the status of the background process using the PID of the same.



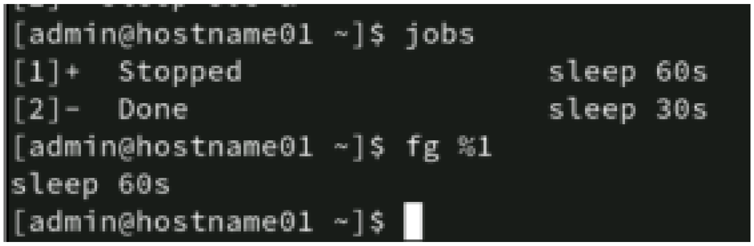
```
[admin@hostname01 ~]$ sleep 60s
^Z
[1]+  Stopped                 sleep 60s
[admin@hostname01 ~]$ jobs -l
[1]+ 16077 Stopped                 sleep 60s
[admin@hostname01 ~]$ ps -p 16077
    PID TTY          TIME CMD
  16077 pts/1    00:00:00 sleep
[admin@hostname01 ~]$
```

4. Run a job in background



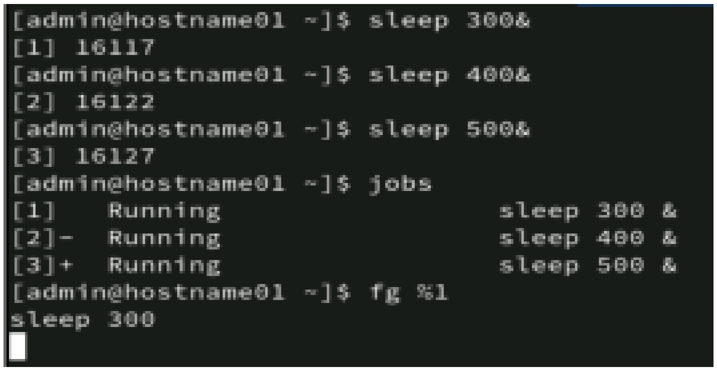
```
[admin@hostname01 ~]$ sleep 30s
^Z
[2]+  Stopped                 sleep 30s
[admin@hostname01 ~]$ bg
[2]+ sleep 30s &
[admin@hostname01 ~]$ █
```

5. Bring a last background job in fore ground



```
[admin@hostname01 ~]$ jobs
[1]+  Stopped                 sleep 60s
[2]-  Done                    sleep 30s
[admin@hostname01 ~]$ fg %1
sleep 60s
[admin@hostname01 ~]$ █
```

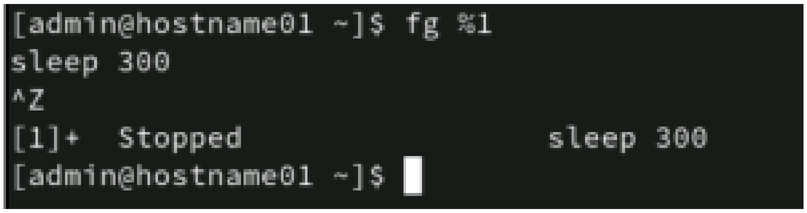
6. Run 3 jobs in background and bring first job in foreground



```
[admin@hostname01 ~]$ sleep 300&
[1] 16117
[admin@hostname01 ~]$ sleep 400&
[2] 16122
[admin@hostname01 ~]$ sleep 500&
[3] 16127
[admin@hostname01 ~]$ jobs
[1]   Running                 sleep 300 &
[2]-  Running                 sleep 400 &
[3]+  Running                 sleep 500 &
[admin@hostname01 ~]$ fg %1
sleep 300
█
```

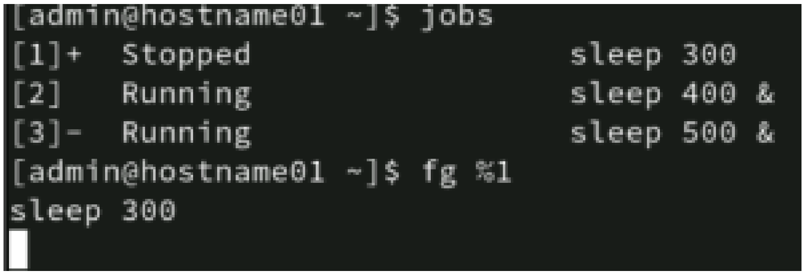
7. Stop current job

Ctrl +z



```
[admin@hostname01 ~]$ fg %1
sleep 300
^Z
[1]+  Stopped                 sleep 300
[admin@hostname01 ~]$ █
```

8. Start stopped job



```
[admin@hostname01 ~]$ jobs
[1]+  Stopped                 sleep 300
[2]   Running                 sleep 400 &
[3]-  Running                 sleep 500 &
[admin@hostname01 ~]$ fg %1
sleep 300
█
```

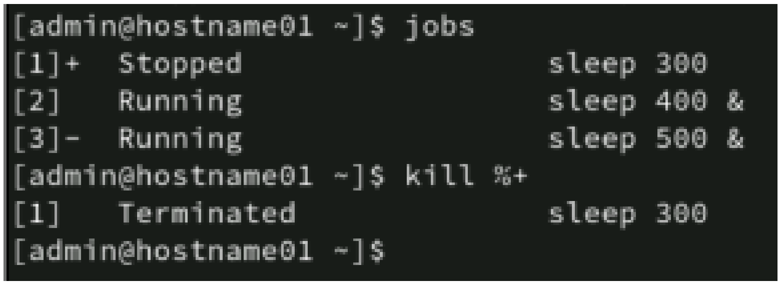
9. Run a job

Ans:Example

Sleep 300 (in foreground)

Sleep 300 &(in background)

10. Kill last job



```
[admin@hostname01 ~]$ jobs
[1]+  Stopped                 sleep 300
[2]   Running                 sleep 400 &
[3]-  Running                 sleep 500 &
[admin@hostname01 ~]$ kill %+
[1]   Terminated              sleep 300
[admin@hostname01 ~]$
```

11. Kill your shell using process id

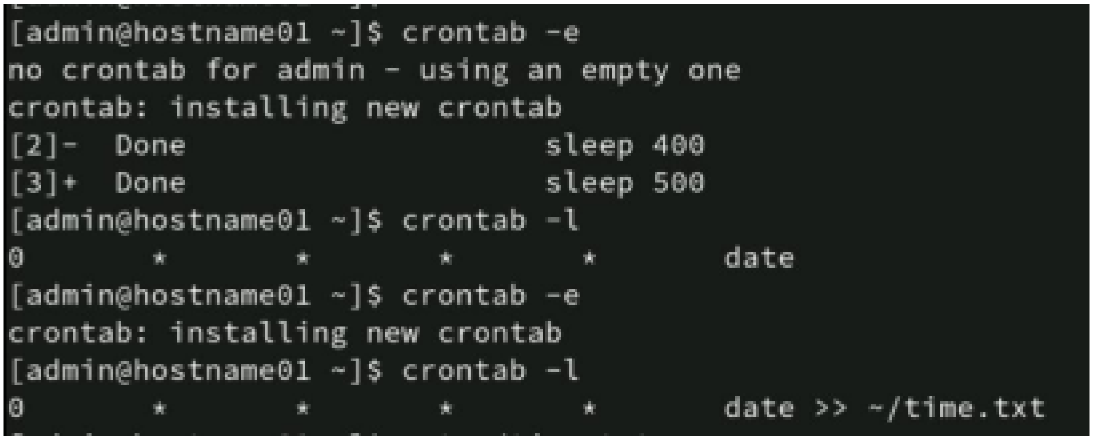
Ans:1. Find the pid:echo $$

2.kill the shell :kill -9 pid

12. Execute a ls command by setting priority as -10 using nice command

Ans: sudo nice -n -10 ls

13. Display a date on every hour using cron tab



```
[admin@hostname01 ~]$ crontab -e
no crontab for admin - using an empty one
crontab: installing new crontab
[2]-  Done                    sleep 400
[3]+  Done                    sleep 500
[admin@hostname01 ~]$ crontab -l
0       *       *       *       *       date
[admin@hostname01 ~]$ crontab -e
crontab: installing new crontab
[admin@hostname01 ~]$ crontab -l
0       *       *       *       *       date >> ~/time.txt
```