**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.

1. Change the word “Netware” in the second line to “Novell Netware”.

%s/Netware/Novell Netware/

b. Insert the text “(such as hard disks and printers)” after “share resources” in the

first line.

Press **gg** to go to the first line.

And arrow keys to navigate

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.”

Press **G** (Shift + g) to go to the last line

Press **o** (lowercase letter "o") to open a new line **below** the last line.

Working shell

1. Type some text on the shell separated by space

$echo “This is the first sentence” “second sentence separated by space”

1: Move cursor one word back

Alt + B

2: Move cursor one word forward

Alt + F

3: Move cursor to the first character

Ctrl + A

4: Move cursor to the end

Ctrl + E

5: Delete test from second word to last character

Ctrl + K

6: Delete the current line

Ctrl + U

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

command completion.

# cat errorlog.txt

3: Display history of command used so far.

# history

4: Search ls command in history file

# history | grep ls

5: Repeat the last command rd

6: Execute 3 command from history file.

7: What are the different shells available.

Bourne Shell(sh), Bourne Again Shell(bash), C Shell(csh), Korn Shell(ksh)

Understanding access permissions

7.1: Create an empty file “demofile” and perform following instruction

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

# touch demofile

# ls

# chmod u-r demofile

# cat demofile

2. Revoke write permission from owner and open using vi

editor and add some contain in it.

#chmod u-w demofile

#type demofile

1. Add read and write permission to owner.

#chmod u+rw demofile

1. Revoke write and execute from other and group

# chmod go-wx demofile

1. Add write permission to group only

# chmod g+w demofile

1. Assign read permission to all

# chmod u+r,g+r,o+r demofile

1. Revoke read permission from others

# chmod o-r demofile

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

# chmod u+x chap1

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

[root@hostname01 /] # touch add.c

[root@hostname01 /] # chmod a+x add.c

10. Remove the execute permission from user, give read permission to

group and others for a file aa.c

# touch aa.c

# chmod u-x aa.c

# chmod og+r aa.c

# ls -l aa.c

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

single command

# touch a.c kk.c nato myfile

# chmod u+x a.c kk.c nato myfile

# ls -l a.c kk.c nato myfile

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

1. Display contents of demo

# mkdir demo

# cp /etc/passwd demo/

# ls demo

2. Revoke read permission from demo directory and use ls

command on it

# chmod -r demo

# ls -l demo

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

/etc/profile file in it

$ chmod -w demo

$ cp /etc/profle demo/

cp: cannot create regular file ‘demo/profile’: Permission denied

4. Delete passwd file from demo directory

$ rm demo/passwd

rm: cannot remove ‘demo/passwd’: Permission denied

5. Revoke execute permission from demo directory and try cd

command on demo.

# chmod -x demo

# cd demo

bash: cd: demo: Permission denied

**Using Process-Related Commands**

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

# ps -u $USER

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

# ps aux

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

process using the PID of the same.

# sleep 60&

# jobs -p

# jobs

1. Run a job in background

command &

1. Bring a last background job in fore ground

fg

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

sleep 60 &

sleep 120 &

sleep 180 &

1. Stop current job

Ctrl + Z

1. Start stopped job

bg %<job\_number>

1. Run a job

<command> &

1. Kill last job

kill %%

1. Kill your shell using process id

kill <PID>

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

nice -n -10 ls

1. Display a date on every hour using cron tab

crontab -e

0 \* \* \* \* date