**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.

Process to do this is:

Type vi network.txt into the terminal

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

Command Used🡪 :s/Netware/Novell Network/

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

first line.

Command used🡪 :1s/share resources/&(such as hard disks and printers)/

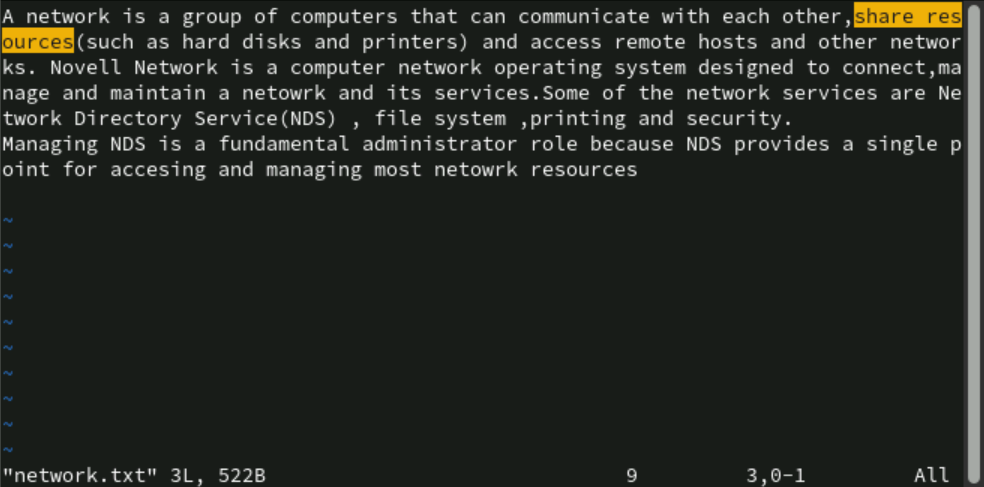
c. Append the following text to the file:

“Managing NDS is a fundamental administrator role because NDS provides a single

point for accessing and managing most network resources.”

Process to do this is :

Just go to the terminal and enter vi network.txt or else just :w and save the file and enter into insert mode in existing activity my pressing o.



Working shell

1. Type some text on the shell separated by space

1: Move cursor one word back -ctrl + w

2: Move cursor one word forward 🡪ctrl +E

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.

A screenshot of a computer error

Description automatically generated

3: Display history of command used so far.

**[admin@hostname01 ~]$ history**

**1 ifconfig**

**2 exit**

**3 passwd root**

**4 cd /root/**

**5 exit**

**6 hostname host01**

**7 su - root**

**8 yum update -y**

**9 su root**

**10 su -**

**11 cd**

**12 poweroff**

**13 ifconfig**

**14 exit**

**15 yum -y update**

**16 su - root**

**17 init 0**

**18 su - root**

**19 pwd**

**20 echo $HOME**

**21 whoami**

**22 ls**

**23 ls -a**

**24 cd**

**25 ls**

**26 ls -a**

**27 ls $home**

**28 ls chap[0-9a-z]\***

**29 mkdir ~/c\_prog**

**30 ls ~**

**31 cd ~/c\_prog**

**32 pwd**

**33 cd**

**34 vi network.txt**

**35 This is an exmaple text for shell navigation**

**36 cat errorlog.txt**

**37 ls**

**38 cat data.txt 2> errorlog.txt**

**39 cat errorlog.txt**

**40 pwd**

**41 ls**

**42 ls | grep errorlog.txt**

**43 cat errorlog.txt**

**44 find ~ -name "errorlog.txt"**

**45 cd**

**46 /home/admin/errorlog.txt**

**47 echo "Sample error log content" > errorlog.txt**

**48 ls | grep errorlog.txt**

**49 cat errorlog.txt**

**50 cat err**

**51 cat errorlog.txt**

**52 history**

**[**

4: Search ls command in history file

**[admin@hostname01 ~]$ history | grep ls**

**22 ls**

**23 ls -a**

**25 ls**

**26 ls -a**

**27 ls $home**

**28 ls chap[0-9a-z]\***

**30 ls ~**

**37 ls**

**41 ls**

**42 ls | grep errorlog.txt**

**48 ls | grep errorlog.txt**

**53 history | grep ls**

5: Repeat the last command rd

Command used🡪[admin@hostname01 ~]$ !!

6: Execute 3 command from history file.

**1.[admin@hostname01 ~]$ !3**

**passwd root**

**passwd: Only root can specify a user name.**

**2.[admin@hostname01 ~]$ !3**

**passwd root**

**passwd: Only root can specify a user name.**

**3. [admin@hostname01 ~]$ !61**

**pwd**

**/home/admin**

7: What are the different shells available.

A screen shot of a computer

Description automatically generated

You can also check which shell you are currently in by using which command and also echo $shell.

Understanding access permissions

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

A screenshot of a computer

Description automatically generated

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

**[admin@hostname01 ~]$ chmod u-r demofile**

**[admin@hostname01 ~]$ ls -l demofile**

**--w-r--r--. 1 admin admin 0 Jan 12 23:49 demofile**

**[admin@hostname01 ~]$ cat demofile**

**cat: demofile: Permission denied**

2. Revoke write permission from owner and open using vi

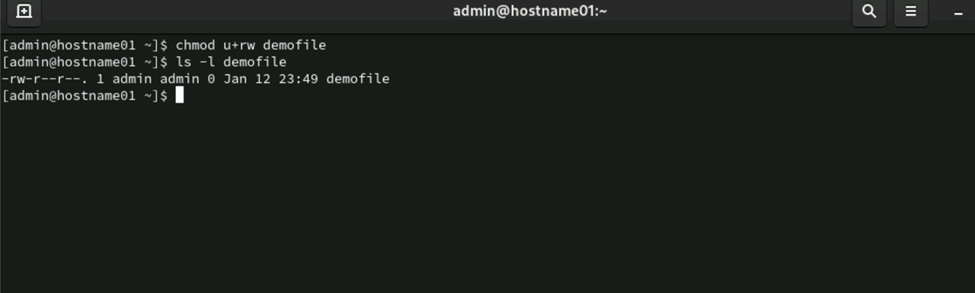
editor and add some contain in it.

A screen shot of a computer

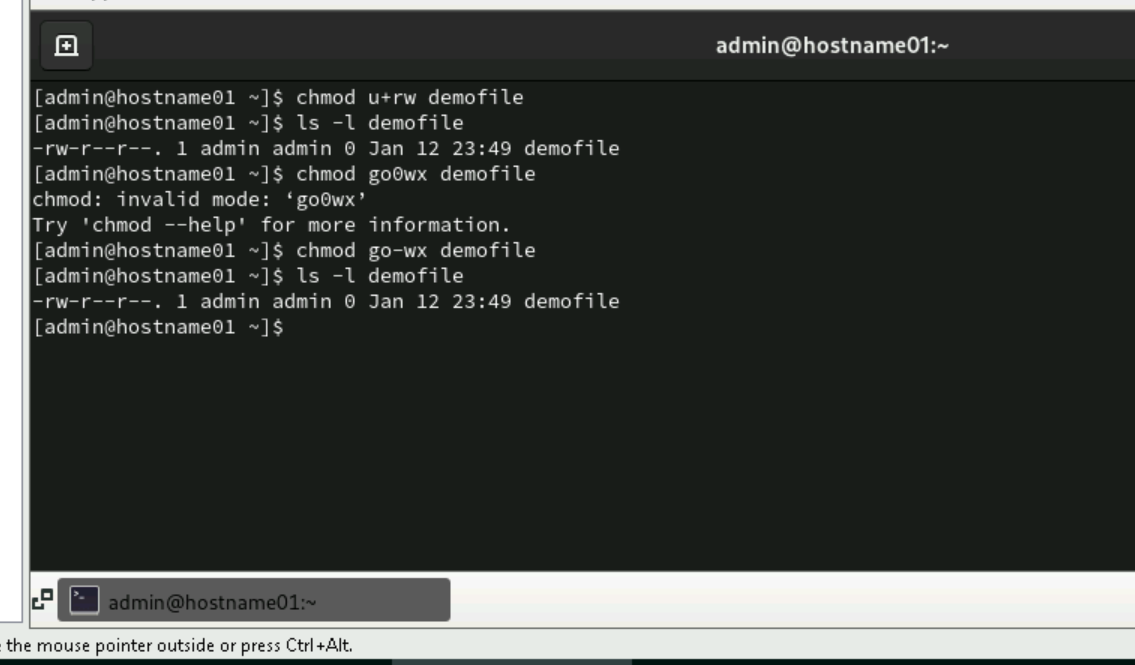
Description automatically generated

After trying to edit it in vi mode it gives permission denied error

1. Add read and write permission to owner.



1. Revoke write and execute from other and group



1. Add write permission to group only

A screen shot of a computer

Description automatically generated

1. Assign read permission to all

**[admin@hostname01 ~]$ chmod a+r demofile**

**[admin@hostname01 ~]$ ls -l demofile**

**-rw-rw-r--. 1 admin admin 0 Jan 12 23:49 demofile**

1. Revoke read permission from others

**[admin@hostname01 ~]$ chmod o-r demofile**

**[admin@hostname01 ~]$ ls -l demofile**

**-rw-rw----. 1 admin admin 0 Jan 12 23:49 demofile**

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

**[admin@hostname01 ~]$ ls -l chap1**

**-rwxr--r--. 1 admin admin 0 Jan 9 20:09 chap1**

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

**admin@hostname01 ~]$ touch add.c**

**[admin@hostname01 ~]$ chmod +x add.c**

**[admin@hostname01 ~]$ ls -l add.c**

**-rwxr-xr-x. 1 admin admin 0 Jan 14 20:26 add.c**

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

group and others for a file aa.c

**admin@hostname01 ~]$ touch add.c**

**[admin@hostname01 ~]$ chmod +x add.c**

**[admin@hostname01 ~]$ ls -l add.c**

**-rwxr-xr-x. 1 admin admin 0 Jan 14 20:26 add.c**

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

single command

**-rw-r--r--. 1 admin admin 0 Jan 14 20:28 aa.c**

**[admin@hostname01 ~]$ touch a.c kk.c nato myfile**

**[admin@hostname01 ~]$ chmod u+x a.c kk.c nato myfile**

**[admin@hostname01 ~]$ ls -l a.c kk.c nato myfile**

**-rwxr--r--. 1 admin admin 0 Jan 14 20:31 a.c**

**-rwxr--r--. 1 admin admin 0 Jan 14 20:31 kk.c**

**-rwxr--r--. 1 admin admin 0 Jan 14 20:31 myfile**

**-rwxr--r--. 1 admin admin 0 Jan 14 20:31 nato**

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

1. Display contents of demo

**[admin@hostname01 ~]$ ls -l demo**

**total 4**

**-rw-r--r--. 1 admin admin 2055 Jan 14 20:35 passwd**

2. Revoke read permission from demo directory and use ls

command on it

**[admin@hostname01 ~]$ chmod -r demo**

**[admin@hostname01 ~]$ ls demo**

**ls: cannot open directory 'demo': Permission denied**

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

/etc/profile file in it

**[admin@hostname01 ~]$ cp /etc/profile demo/**

**cp: cannot create regular file 'demo/profile': Permission denied**

4. Delete passwd file from demo directory

**[admin@hostname01 ~]$ chmod +w demo**

**[admin@hostname01 ~]$ rm demo/passwd**

**[admin@hostname01 ~]$ ls demo**

**ls: cannot open directory 'demo': Permission denied**

[admin@hostname01 ~]$

5. Revoke execute permission from demo directory and try cd

command on demo.

**[admin@hostname01 ~]$ chmod -x demo**

**[admin@hostname01 ~]$ cd demo**

**bash: cd: demo: Permission denied**

**Using Process-Related Commands**

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

**[admin@hostname01 ~]$ ps -u $user**

**USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND**

**admin 2046 0.0 0.1 374144 9568 tty2 Ssl+ Jan11 0:00 /usr/libexec/gdm-wayland-session env GNOME\_SHELL\_SESSION\_MODE=cl**

**admin 2055 0.0 0.2 513020 19100 tty2 Sl+ Jan11 0:00 /usr/libexec/gnome-session-binary**

**admin 10410 0.0 0.0 224112 5504 pts/0 Ss 20:14 0:00 bash**

**admin 10693 0.0 0.0 225368 3456 pts/0 R+ 20:43 0:00 ps -u**

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

**ps -ef**

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 300 &**

**[1] 10709**

1. Run a job in background

**[admin@hostname01 ~]$ sleep 200 &**

**[2] 10738**

**[1] Done sleep 300**

5. Bring a last background job in fore ground

**[admin@hostname01 ~]$ fg**

**sleep 200**

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

**[admin@hostname01 ~]$ sleep 300 & sleep 400 & sleep 500 &**

**[2] 10831**

**[3] 10832**

**[4] 10833**

7. Stop current job 🡪 **ctrl + z**

8. Start stopped job 🡪**bg**

9. Run a job

**[admin@hostname01 ~]$ sleep 100 &**

**[5] 10846**

10. Kill last job

**[admin@hostname01 ~]$ kill %%**

**[5]+ Terminated sleep 100**

11. Kill your shell using process id

**[admin@hostname01 ~]$ kill %%**

**[5]+ Terminated sleep 100**

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

**[admin@hostname01 ~]$ nice -n -10 ls**

**nice: cannot set niceness: Permission denied**

**aa.c chap1 chapb demofile Downloads filea fileUNIX lsdoc nato Pictures users**

**a.c chap2 chapz Desktop errorlog.txt filec friends Music network.txt Public Videos**

**add.c chapa demo Documents exampleUNIX fileo kk.c myfile newfriend Templates**

**[2] Done sleep 300**

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**

**[3]- Done sleep 400**

**[4]+ Done sleep 500**