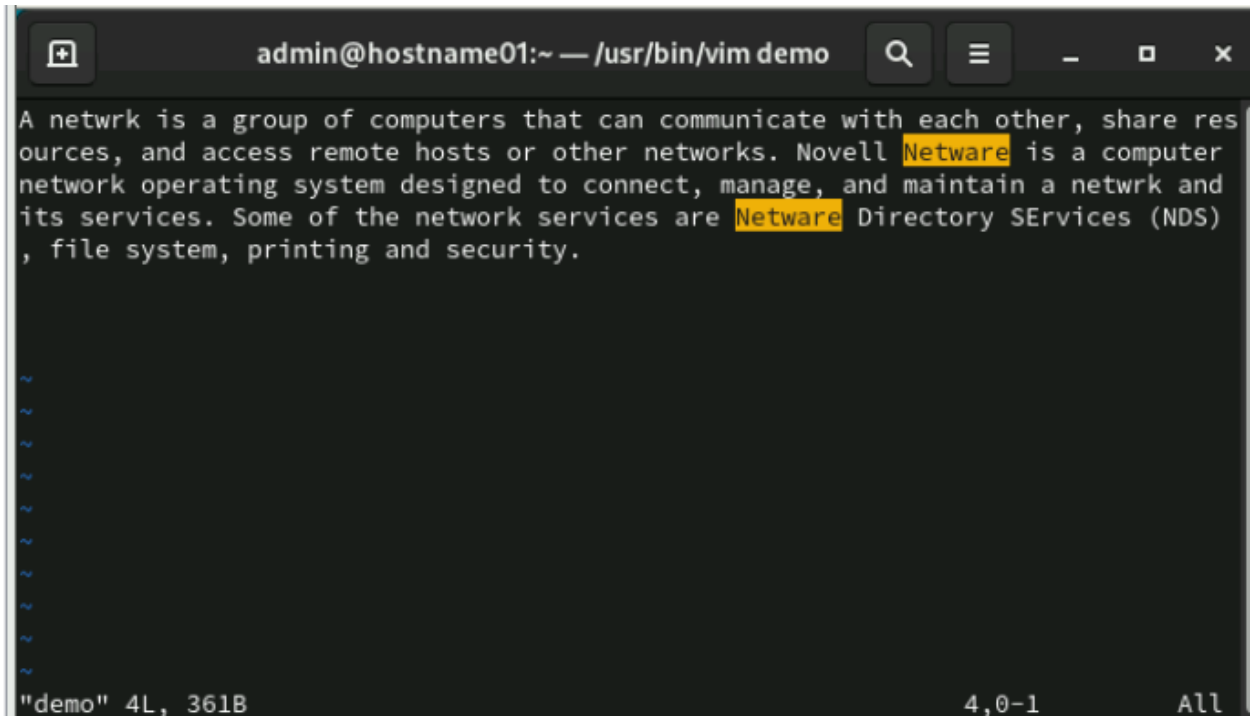
Working with a Vi Editor:

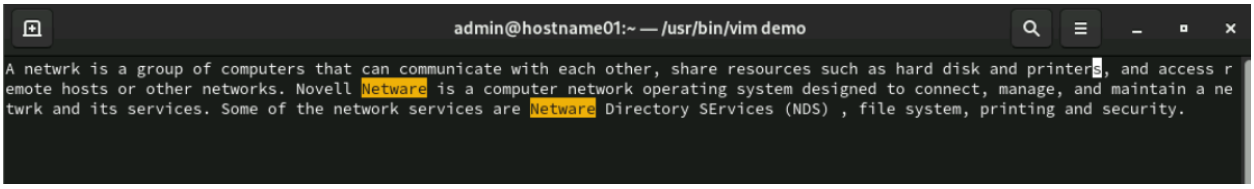
1: Create a file using vi. Enter the following text:

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”.



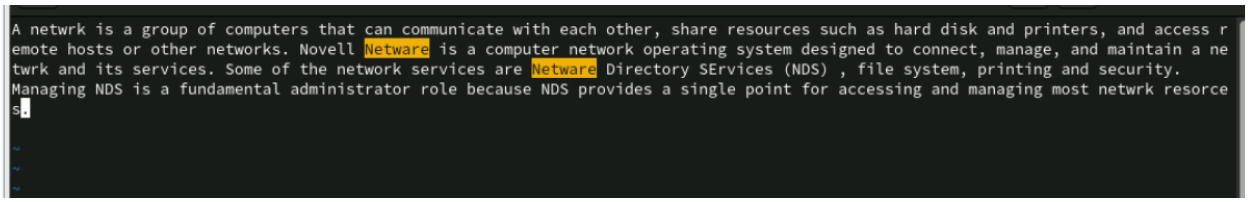
b. Insert the text “(such as hard disks and printers)” after “share resources” in the first line.



c. Append the following text to the file:

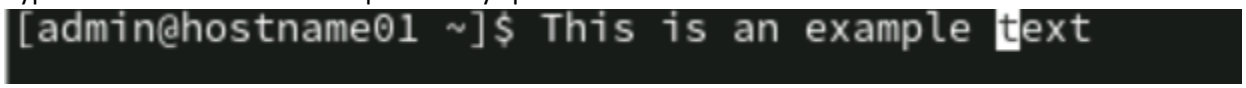
“Managing NDS is a fundamental administrator role because NDS provides a single

point for accessing and managing most network resources.”

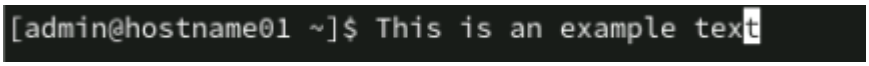


Working shell

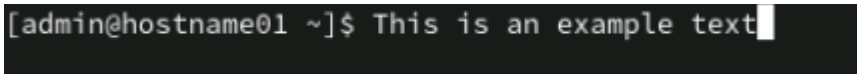
1. Type some text on the shell separated by space



1: Move cursor one word back

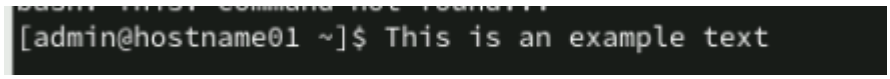


2: Move cursor one word forward

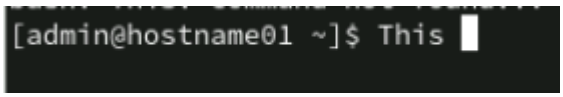


3: Move cursor to the first character

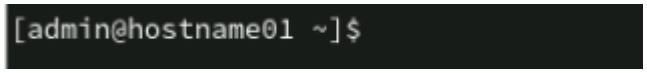
4: Move cursor to the end



5: Delete test from second word to last character



6: Delete the current line



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

```
[admin@hostname01 ~]$ cat errorlog.txt
Managing NDS is a fundamental administartor role because NDS provides a single poin
t for accessing and managing most network resources.
```

typing cat erro and then press Tab to auto-complete the filename.

3: Display history of command used so far.

```
244  vi example
245  cd /
246  cd home
247  ls
248  cd admin
249  ls
250  cat demo
251  cd admin
252  vi demo
253  /admin
254  cd /
255  cd home/admin
256  ls
257  This is an example text
258  history
259  gedit errorlog.txt
260  cat errorlog.txt
261  history
```

4: Search ls command in history file

```
165  ls
170  ls
172  ls
174  ls
178  ls
180  ls
182  ls
184  ls
186  ls
189  ls
192  ls
203  ls
207  ls
243  ls
247  ls
249  ls
256  ls
262  history | grep ls
```

5: Repeat the last command rd
USE “!!”

6: Execute 3 command from history file.

```
[admin@hostname01 ~]$ !3
passwd root
passwd: Only root can specify a user name.
```

7: What are the different shells available.

```
[admin@hostname01 ~]$ cat /etc/shells
/bin/sh
/bin/bash
/usr/bin/sh
/usr/bin/bash
[admin@hostname01 ~]$
```

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.**

```
~
E45: 'readonly' option is set (add ! to override)              2,0-1         All
```

2. **Add read and write permission to owner.**

```
[admin@hostname01 ~]$ chmod u+rw demofile
```

3. **Revoke write and execute from other and group**

```
[admin@hostname01 ~]$ chmod go-wx demofile
```

4. **Add write permission to group only**

```
[admin@hostname01 ~]$ chmod g+w demofile
```

5. **Assign read permission to all**

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

6. **Revoke read permission from others**

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

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

```
[admin@hostname01 ~]$ chmod u+x chap1
```

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

```
[admin@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**

```
[admin@hostname01 ~]$ chmod u-x,g+r,o+r aa.c
```

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

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

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

1. **Display contents of demo**

```
[admin@hostname01 ~]$ mkdir demo1
[admin@hostname01 ~]$ cp /etc/passwd demo1/
[admin@hostname01 ~]$ ls demo1
passwd
```

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

```
[admin@hostname01 ~]$ chmod -r demo1
[admin@hostname01 ~]$ ls demo1
ls: cannot open directory 'demo1': Permission denied
[admin@hostname01 ~]$
```

3. **Revoke write permission from demo directory and try to copy /etc/profile file in it**

```
[admin@hostname01 ~]$ cp /etc/profile demo1/
cp: cannot create regular file 'demo1/profile': Permission denied
```

4. **Delete passwd file from demo directory**

```
[admin@hostname01 ~]$ rm demo1/passwd
rm: cannot remove 'demo1/passwd': Permission denied
```

5. **Revoke execute permission from demo directory and try cd command on demo.**

```
[admin@hostname01 ~]$ chmod -x demo1
[admin@hostname01 ~]$ cd demo1
bash: cd: demo1: 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
   2049 ?        00:00:02 systemd
   2051 ?        00:00:00 (sd-pam)
   2067 ?        00:00:00 gnome-keyring-d
   2071 tty2     00:00:00 gdm-wayland-ses
   2078 ?        00:00:00 dbus-broker-lau
   2080 ?        00:00:01 dbus-broker
   2084 tty2     00:00:00 gnome-session-b
   2116 ?        00:00:00 gnome-session-c
   2117 ?        00:00:00 gnome-session-b
   2134 ?        00:07:29 gnome-shell
   2150 ?        00:00:00 gvfsd
   2155 ?        00:00:00 gvfsd-fuse
   2164 ?        00:00:00 at-spi-bus-laun
   2169 ?        00:00:00 dbus-broker-lau
   2170 ?        00:00:00 dbus-broker
   2182 ?        00:00:00 xdg-permission-
   2184 ?        00:00:00 gnome-shell-cal
```

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 COMMAND
root           1  0.1  0.2 173960 16512 ?        Ss   16:06   0:09 /usr/lib/systemd
root           2  0.0  0.0      0     0 ?        S    16:06   0:00 [kthreadd]
root           3  0.0  0.0      0     0 ?        S    16:06   0:00 [pool_workqueue_
root           4  0.0  0.0      0     0 ?        I<   16:06   0:00 [kworker/R-rcu_g
root           5  0.0  0.0      0     0 ?        I<   16:06   0:00 [kworker/R-sync_
root           6  0.0  0.0      0     0 ?        I<   16:06   0:00 [kworker/R-slub_
root           7  0.0  0.0      0     0 ?        I<   16:06   0:00 [kworker/R-netns
root          10  0.0  0.0      0     0 ?        I    16:06   0:00 [kworker/u512:0-
root          11  0.0  0.0      0     0 ?        I<   16:06   0:00 [kworker/R-mm_pe
root          12  0.0  0.0      0     0 ?        I    16:06   0:00 [kworker/u512:1-
root          13  0.0  0.0      0     0 ?        I    16:06   0:00 [rcu_tasks_kthre
root          14  0.0  0.0      0     0 ?        I    16:06   0:00 [rcu_tasks_rude_
root          15  0.0  0.0      0     0 ?        I    16:06   0:00 [rcu_tasks_trace
root          16  0.0  0.0      0     0 ?        S    16:06   0:00 [ksoftirqd/0]
root          17  0.0  0.0      0     0 ?        I    16:06   0:03 [rcu_preempt]
root          18  0.0  0.0      0     0 ?        S    16:06   0:00 [rcu_exp_par_gp_
root          19  0.0  0.0      0     0 ?        S    16:06   0:00 [rcu_exp_gp_kthr
```

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 30 &
[1] 4214
[admin@hostname01 ~]$ jobs -l
[1]+  4214 Done                    sleep 30
[admin@hostname01 ~]$
```

3. **Run a job in background**
   command &

4. **Bring a last background job in fore ground**
   fg

5. **Run 3 jobs in background and bring first job in foreground**
   sleep 60 &
   sleep 120 &
   sleep 180 &

6. **Stop current job**
   Ctrl + Z

7. **Start stopped job**
   bg %<job_number>

8. **Run a job**
   <command> &

9. **Kill last job**
   kill %%

10. **Kill your shell using process id**
    kill <PID>

11. **Execute a ls command by setting priority as -10 using nice command**
    nice -n -10 ls

12. **Display a date on every hour using cron tab**
    crontab -e
    0 * * * * date