Linux System Administrator Practicals

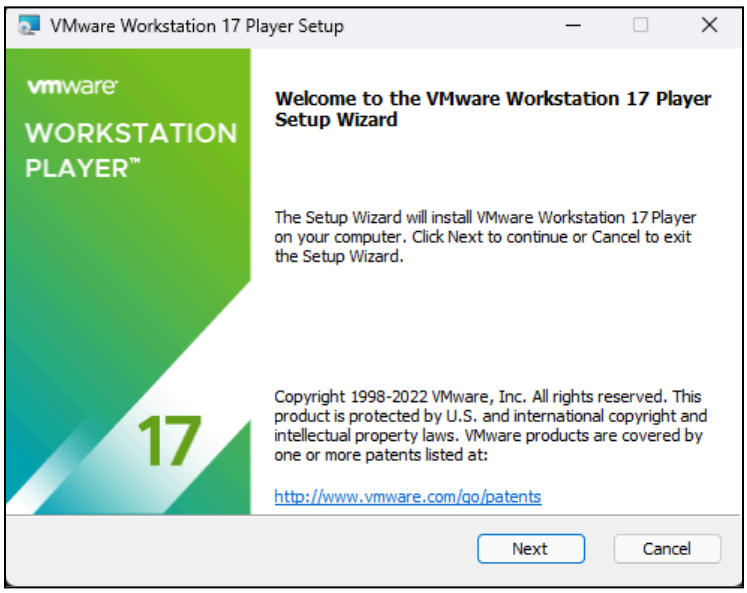
- [ ] NFS??
- [ ] Apache
- [ ] Initial settings

Practical 1
Installation of VMware and Red Hat Linux

Steps :

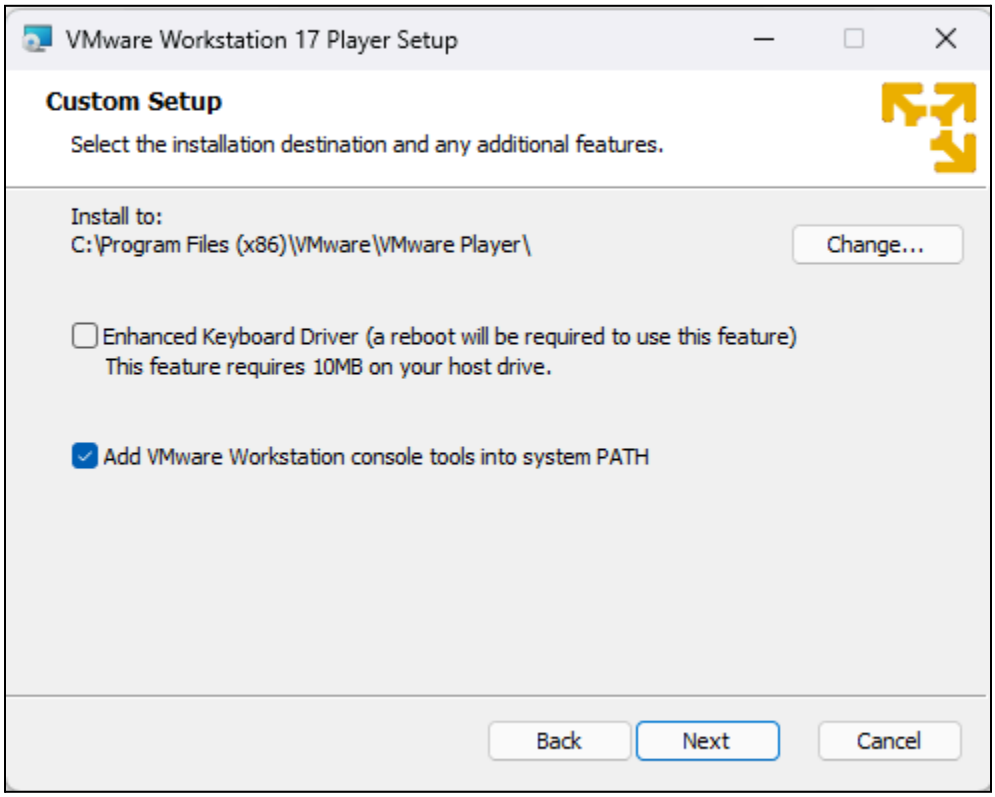
Installation and Setup of VMware :

1. Install VMware Workstation. Then open the setup wizard.

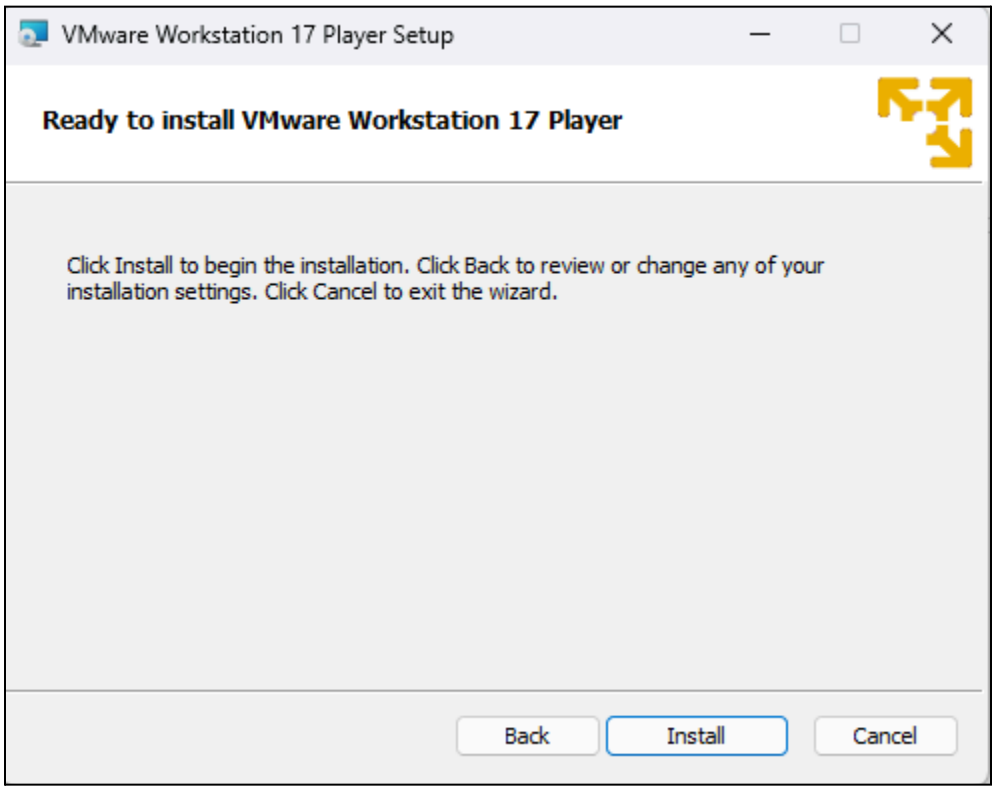


2. After clicking on Next, read the End-User License Agreement and then click on the Accept checkbox. Then click on Next.

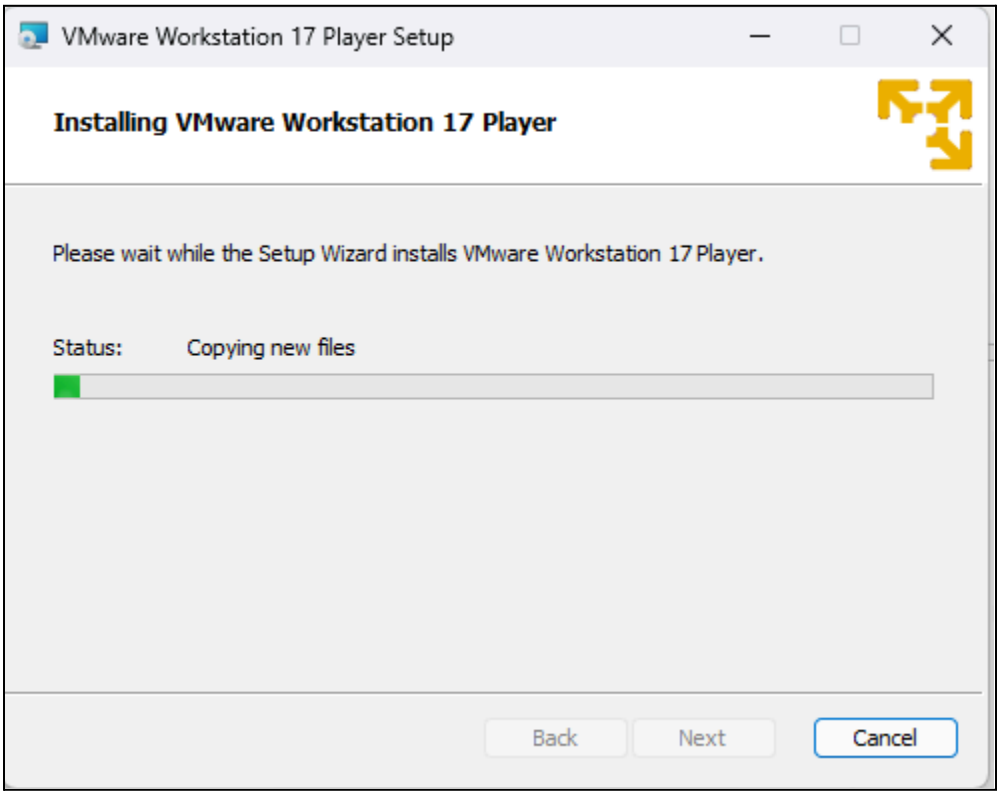
3. Change the installation location as per your wish and then click on Next.



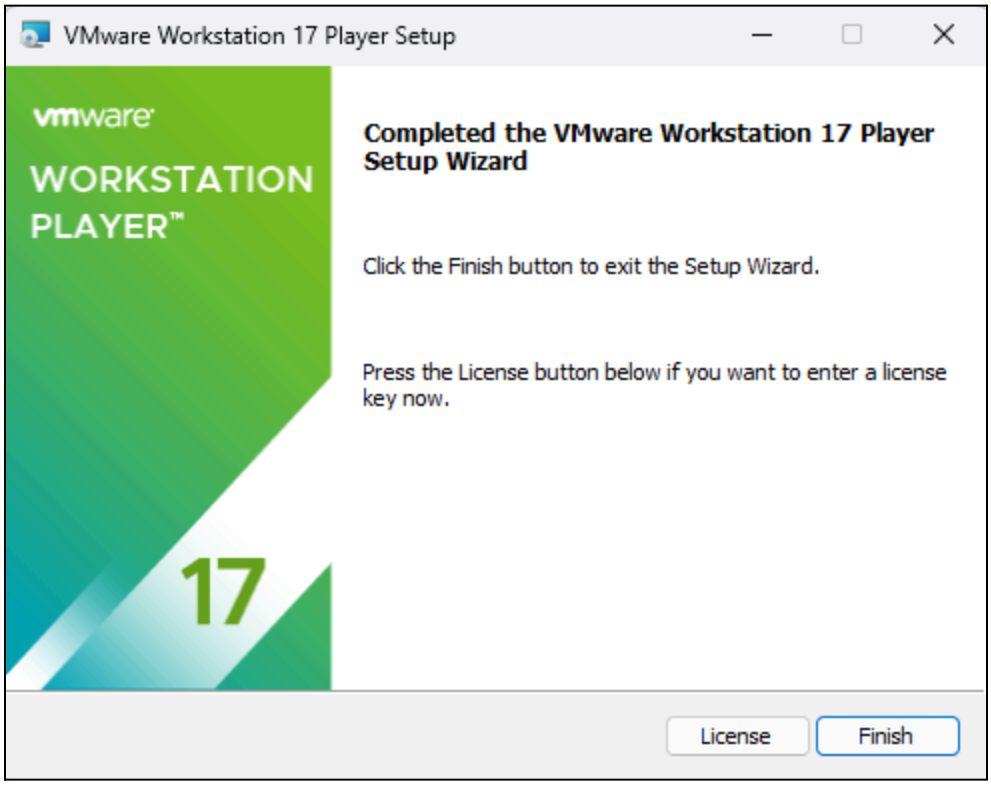
4. Finally click on “Install”



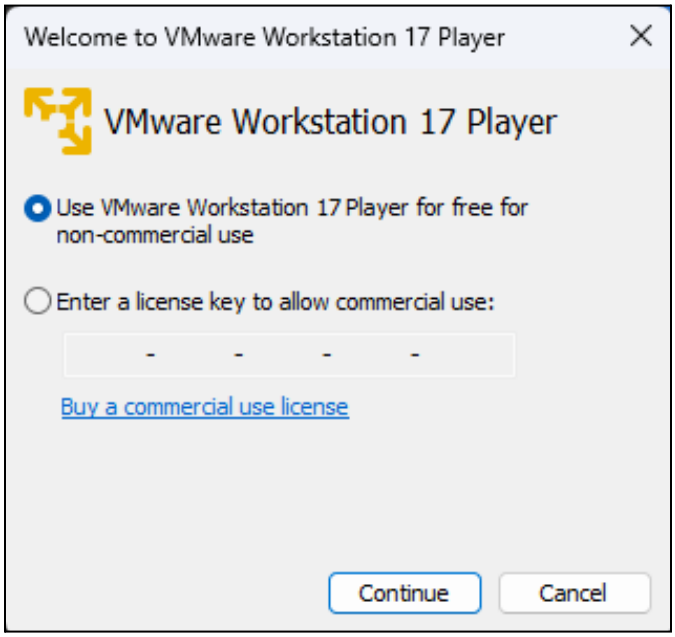
The VMware Workstation is now being installed.



5. Click on Finish.

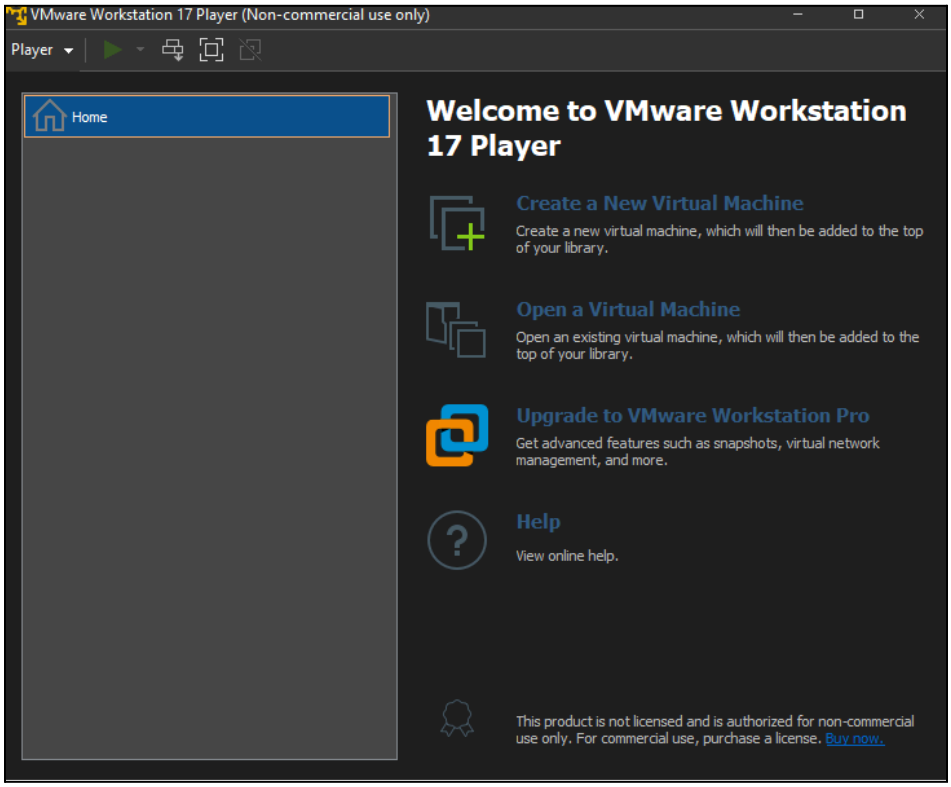


6. Open VMware Workstation after the installation is complete and select “Use VMware Workstation for free for non-commercial use” and click on “Continue”.

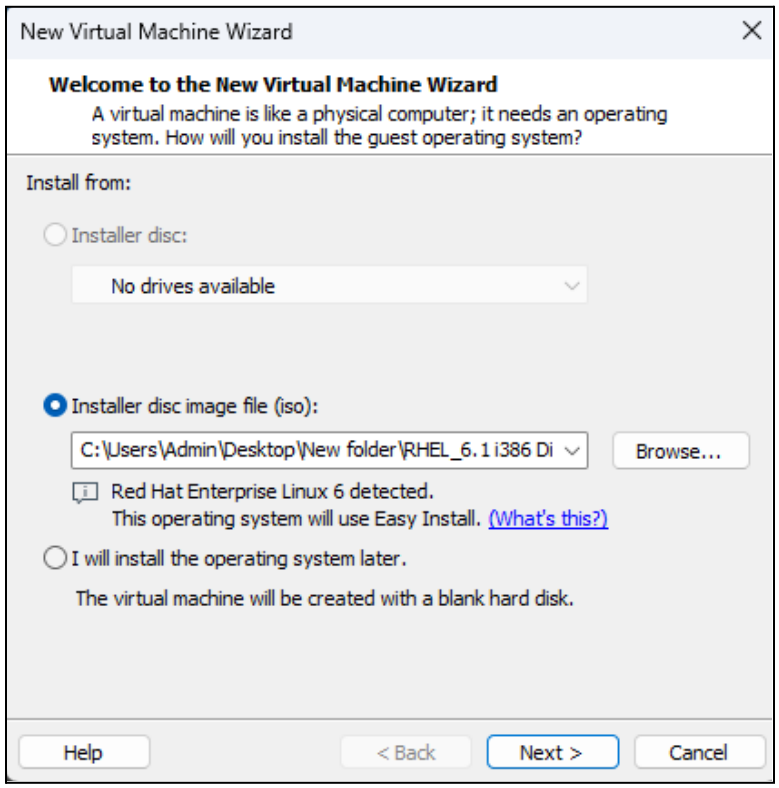


Installation of RHEL

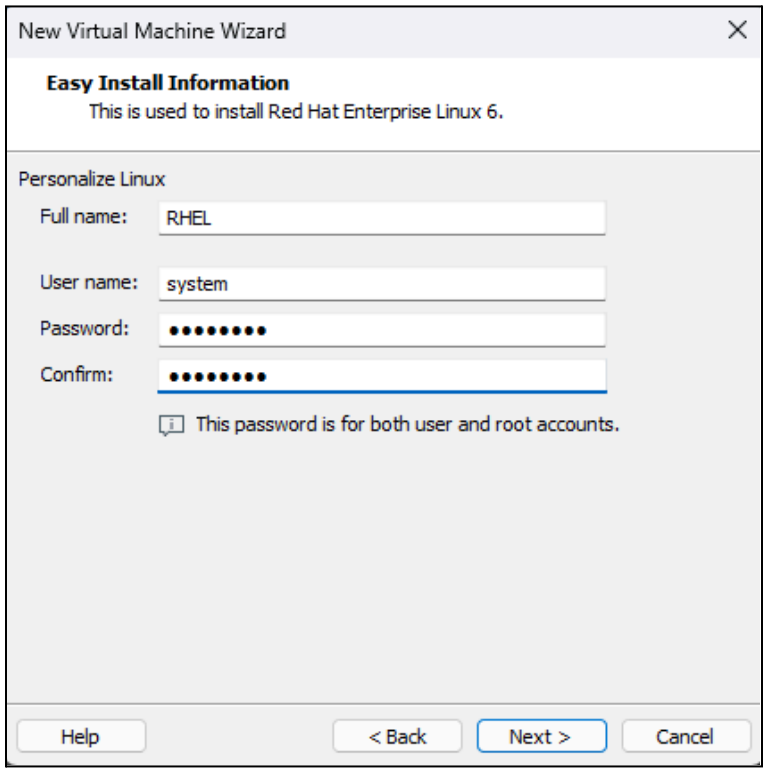
1. The VMware homepage is displayed after the installation and setup. Click on “Create a New Virtual Machine”



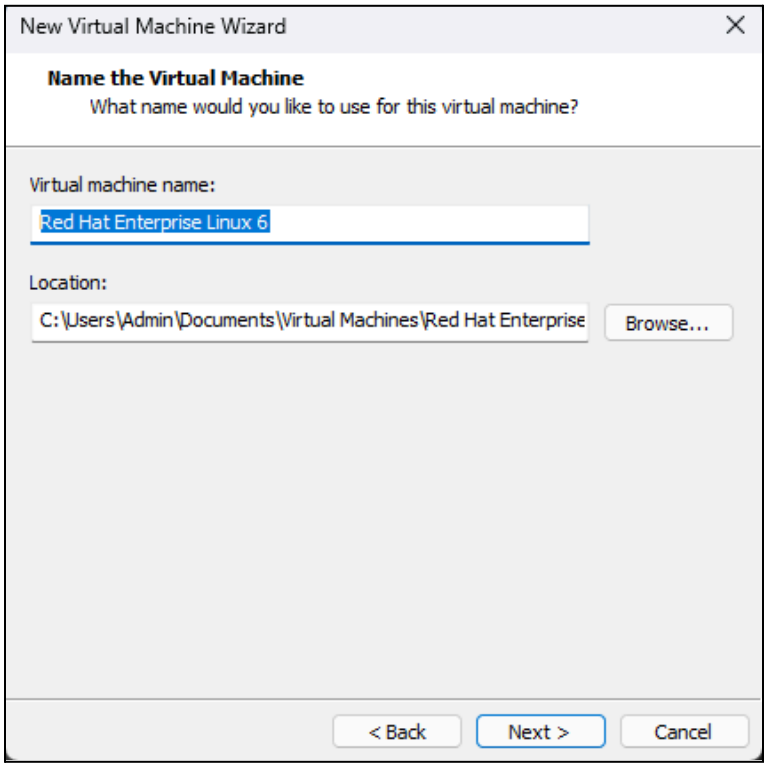
2. Then browse for the ISO file in the directory and open it. Then click on Next



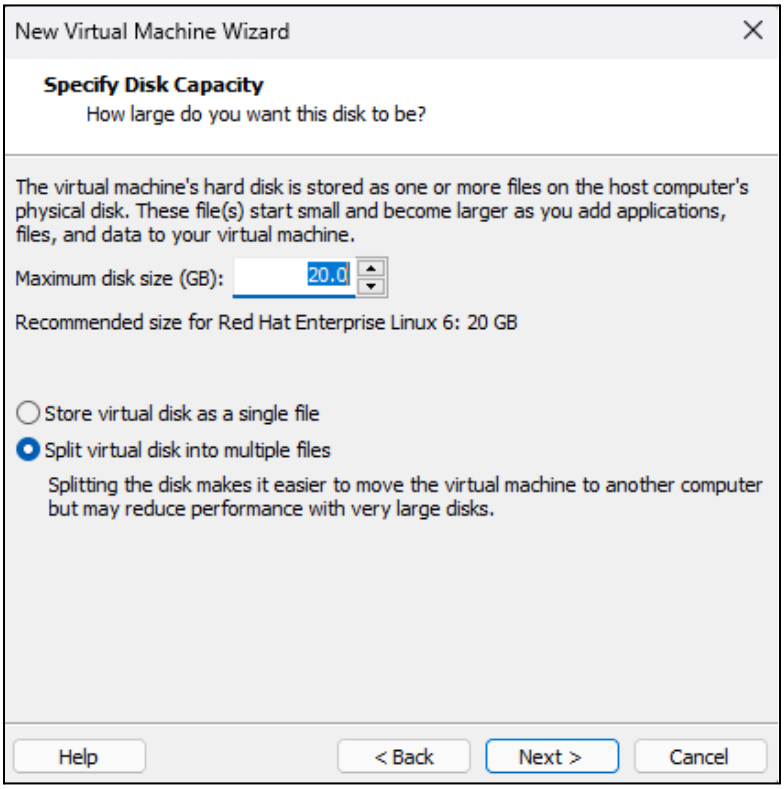
3. Create a user and fill in the details of the user such as the username, password etc and then click on Next



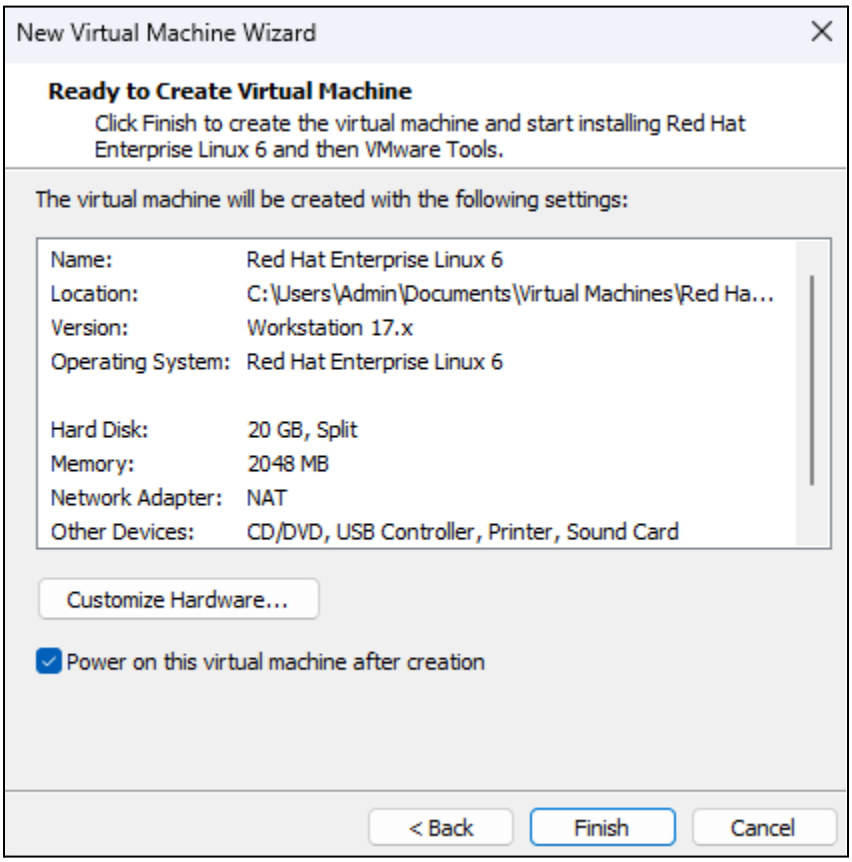
4. Enter the name of the virtual machine and then click on Next

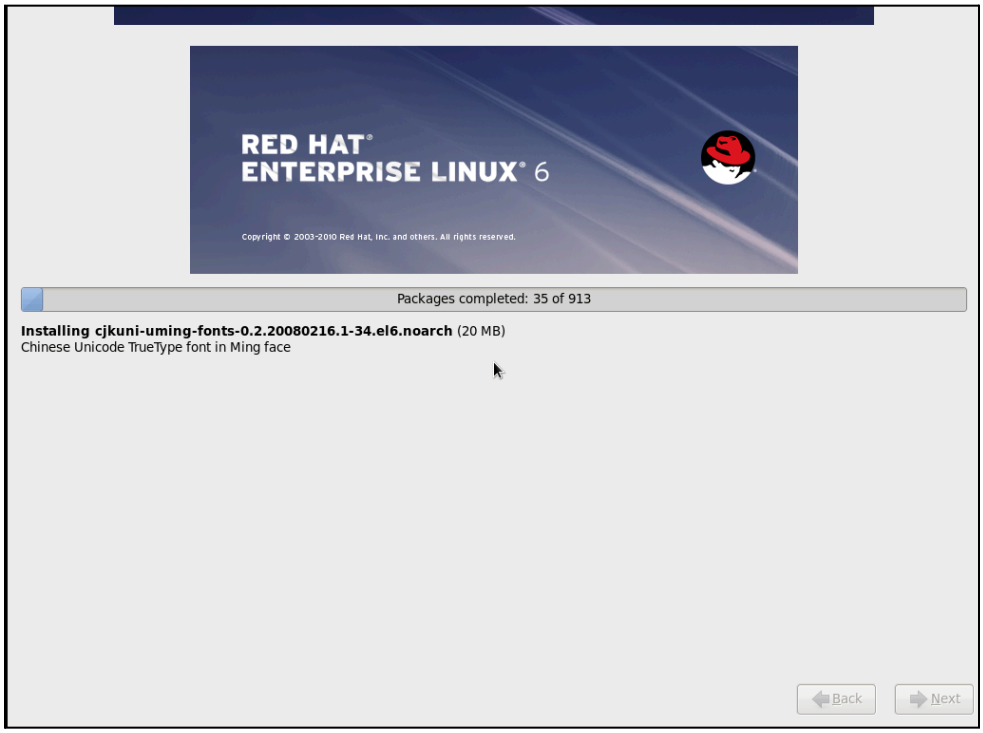


5. Specify the disk capacity (20.0 by default) and the partition configurations. Then click on Next.

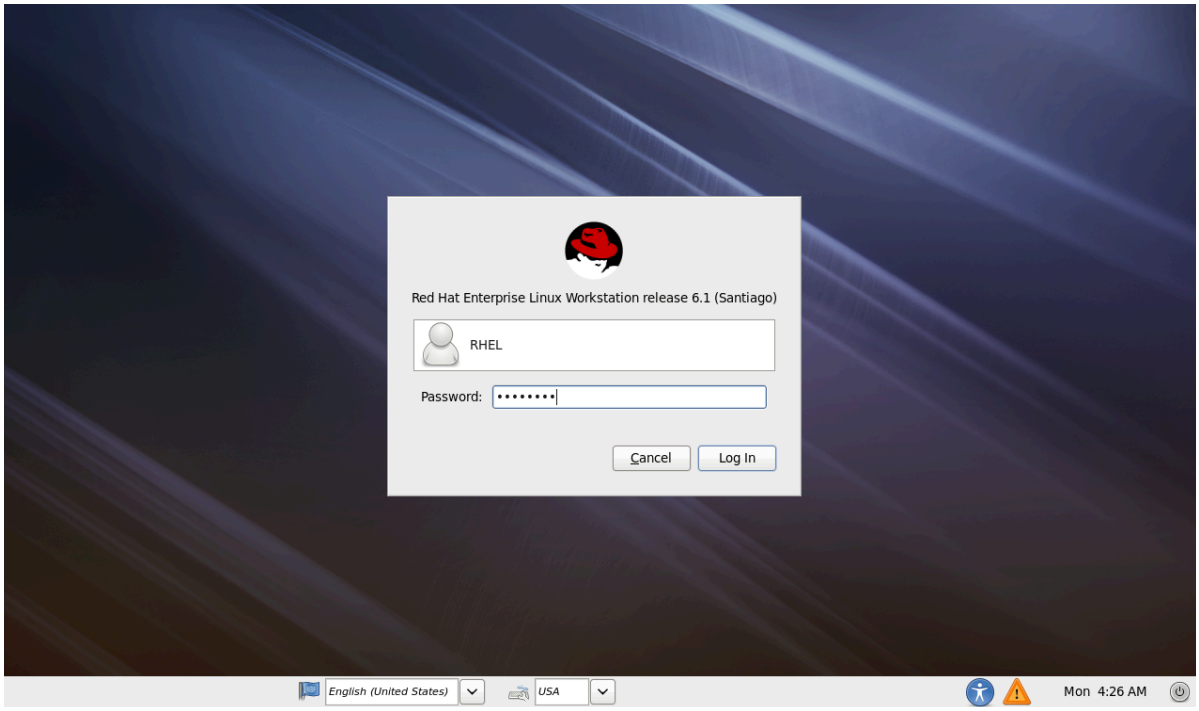


6. Verify all the settings before clicking on Finish.



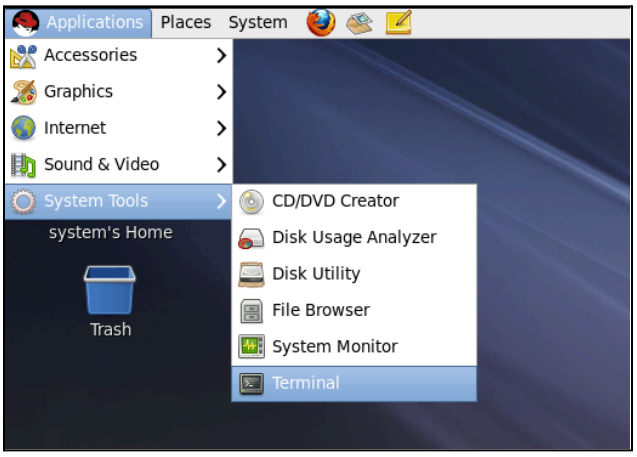


7. Log in to the user and enter the login details and then click on Log In

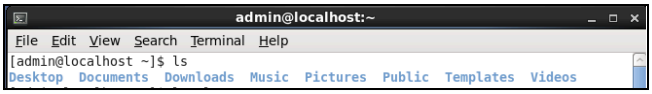


Basic Commands

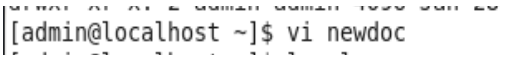
To get to the Linux Terminal, Click on Applications => System Tools => Terminal

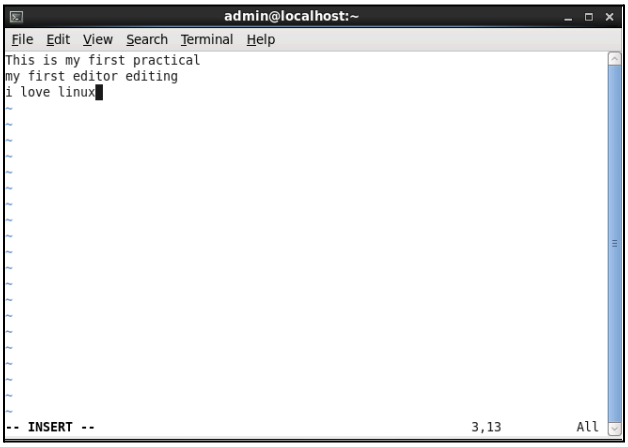


1. **ls:** The ls command in Linux is a shell command that lists the contents of a directory.



2. **vi (file_name):** The vi command in Linux is a text editor that allows you to create and edit files in a terminal window.





3. **ls -l:** The ls -l command in Linux is a shell command that lists the contents of a directory in a long listing format.

```
[admin@localhost ~]$ vi newdoc
[admin@localhost ~]$ ls -l
total 36
drwxr-xr-x. 2 admin admin 4096 Jan 28 08:45 Desktop
drwxr-xr-x. 2 admin admin 4096 Jan 28 08:45 Documents
drwxr-xr-x. 2 admin admin 4096 Jan 28 08:45 Downloads
drwxr-xr-x. 2 admin admin 4096 Jan 28 08:45 Music
-rw-rw-r--. 1 admin admin   64 Jan 28 08:56 newdoc
drwxr-xr-x. 2 admin admin 4096 Jan 28 08:45 Pictures
drwxr-xr-x. 2 admin admin 4096 Jan 28 08:45 Public
drwxr-xr-x. 2 admin admin 4096 Jan 28 08:45 Templates
drwxr-xr-x. 2 admin admin 4096 Jan 28 08:45 Videos
[admin@localhost ~]$
```

4. **ls -i:** The ls -i command in Linux is a shell command that lists the contents of a directory and shows the inode number of each file and directory.

```
[admin@localhost ~]$ ls -i
8351 Desktop    8352 Downloads  8471 newdoc    8354 Public     8358 Videos
8355 Documents  8356 Music      8357 Pictures  8353 Templates
```

5. **ls -l n***:The ls -l n* command in Linux is a shell command that lists the contents of the current directory that start with the letter n and shows detailed information about them in a long listing format.

```
[admin@localhost ~]$ ls -l n*
-rw-rw-r--. 1 admin admin 64 Jan 28 08:56 newdoc
[admin@localhost ~]$
```

6. **chmod 700 (file_name):** The chmod 700 command in Linux is a shell command that sets the permissions of a file or directory to 700. This means that the owner of the file or directory has read, write, and execute permissions, and no one else has any permissions.

```
[admin@localhost ~]$ chmod 700 newdoc
[admin@localhost ~]$ ls -l n*
-rwx------. 1 admin admin 64 Jan 28 08:56 newdoc
```

7. **chmod 600 (file_name):** The chmod 600 command in Linux is a shell command that sets the permissions of a file or directory to 600. This means that the owner of the file or directory has read and write permissions, and no one else has any permissions.

```
[admin@localhost ~]$ chmod 600 newdoc
[admin@localhost ~]$ ls -l n*
-rw-------. 1 admin admin 64 Jan 28 08:56 newdoc
```

8. **chmod 444 (file_name):** The chmod 444 command in Linux is a shell command that sets the permissions of a file or directory to 444. This means that the owner, group, and others of the file or directory have only read permissions, and no one has to write or execute permissions.

```
[admin@localhost ~]$ chmod 444 newdoc
[admin@localhost ~]$ ls -l n*
-r--r--r--. 1 admin admin 64 Jan 28 08:56 newdoc
```

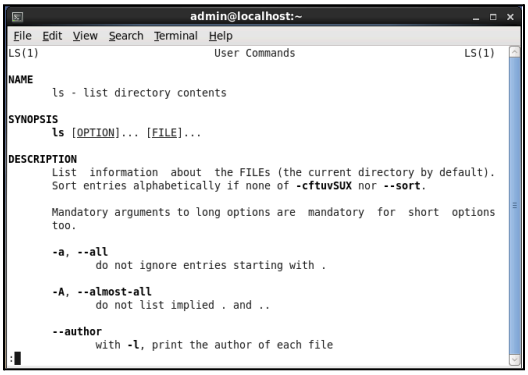
9. **chmod 666 (file_name):** The chmod 666 command in Linux is a shell command that sets the permissions of a file or directory to 666. This means that the owner, group, and

others of the file or directory have read and write permissions, and no one has executed permissions.

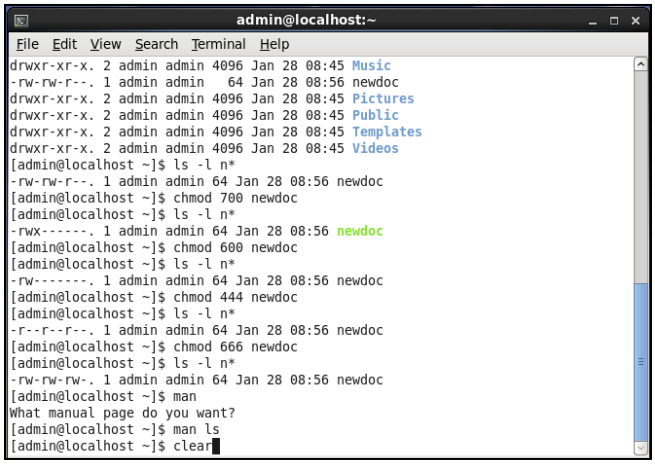
```
[admin@localhost ~]$ chmod 666 newdoc
[admin@localhost ~]$ ls -l n*
-rw-rw-rw-. 1 admin admin 64 Jan 28 08:56 newdoc
[admin@localhost ~]$
```

10. **man ls:** The man ls command in Linux is a shell command that displays the manual page for the ls command.

```
[admin@localhost ~]$ man ls
[admin@localhost ~]$
```



11. **clear:** The clear command in Linux is a shell command that is used to clear the terminal screen.



12. **pwd**: The pwd command in Linux is used to print the full path of the current working directory.

```
[admin@localhost ~]$ pwd
/home/admin
```

13. **ls -R**: The ls -R command in Linux is used to list the contents of a directory and its subdirectories recursively. This means that it will display all the files and folders in the current directory, and then go into each subdirectory and show its contents, and so on.

```
[admin@localhost ~]$ ls -R
.:
Desktop    Downloads  newdoc    Public     Videos
Documents  Music      Pictures  Templates

./Desktop:

./Documents:

./Downloads:

./Music:

./Pictures:

./Public:

./Templates:

./Videos:
```

14. **ls -d:** The ls -d command in Linux is used to list directories themselves, rather than their contents. This can be useful when you want to display only the directories from within your current directory, or when you want to show the full path of a specific directory.

```
./Videos:
[admin@localhost ~]$ ls -d
.
```

15. **ls -RL**: The ls -RL command in Linux is used to list the contents of a directory and its subdirectories recursively, and to follow symbolic links to their targets. This means that it will display all the files and folders in the current directory, then go into each subdirectory and show its contents, and also show the files and folders that are linked by any symbolic links in the directory tree.

```
[admin@localhost ~]$ ls -RL
.:
Desktop    Downloads  newdoc    Public     Videos
Documents  Music      Pictures  Templates

./Desktop:

./Documents:

./Downloads:

./Music:

./Pictures:

./Public:

./Templates:

./Videos:
[admin@localhost ~]$ s
```

16. **whoiam**: The whoami command in Linux is used to display the username of the current user who is logged in to the system.

```
[admin@localhost ~]$ whoami
admin
```

17. **who**: The who command in Linux is used to display information about currently logged-in users on the system. It can also show other useful information, such as the time of the last system boot, the current run level, the active processes, and more.

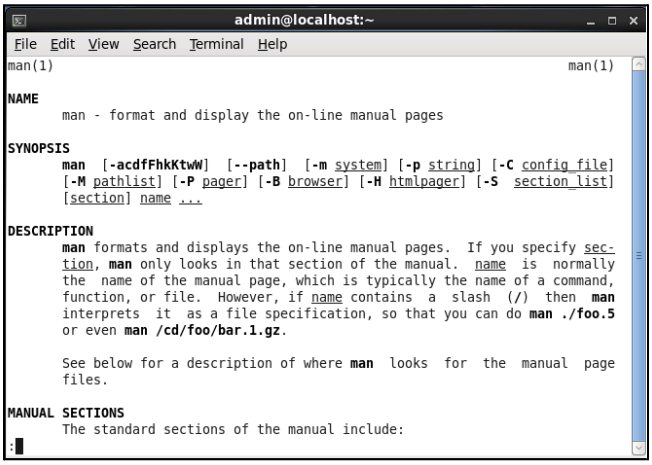
```
[admin@localhost ~]$ who
admin    tty7         2024-01-28 08:45 (:0)
admin    pts/0        2024-01-28 08:52 (:0.0)
```

18. **ps**: The ps command in Linux is used to view information about the processes running on your system.

```
[admin@localhost ~]$ ps
  PID TTY          TIME CMD
10839 pts/0    00:00:00 bash
10941 pts/0    00:00:00 ps
[admin@localhost ~]$
```

19. **man man**: The man man command in Linux is used to display the manual page for the man command itself.

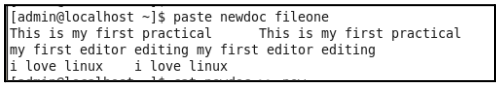
```
[admin@localhost ~]$ man man
[admin@localhost ~]$
```



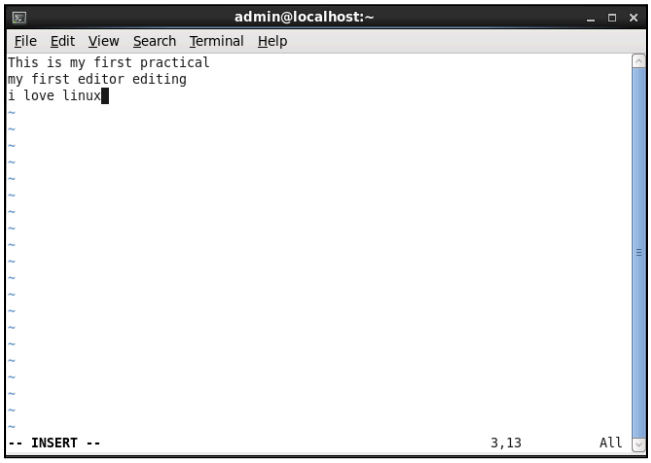
20. **cat (file_name_1) > file_name_2**: The cat (file_name_1) > (file_name_2) command in Linux is used for copying the contents of one file to another file. It concatenates the contents of file_name_1 and redirects the output to file_name_2.

```
[admin@localhost ~]$ cat newdoc > fileone
[admin@localhost ~]$ vi fileone
```

21. **paste file_name_1 file_name_2**: The paste file_name_1 file_name_2 command in Linux is used for merging the corresponding lines of two files horizontally, separated by a tab character. It outputs the merged lines to the standard output.



22. **cat file_name_1 >> file_name_2**: The cat file_name_1 >> file_name_2 command in Linux is used for appending the contents of one file to the end of another file. It concatenates the contents of file_name_1 and redirects the output to file_name_2. If file_name_2 does not exist, it will be created. If it already exists, it will be appended.



Practical 2
Working with users, groups and permissions

i. Users :

1. Open the Linux Terminal and login as the root user.

Code :
su - root
Note : $ sign is for normal user while # sign is for root users

```
[rhel6@localhost ~]$ su - root
Password:
[root@localhost ~]#
```

2. Then, add a new user with the username as “newuser” and password as “Newuser@123”

Code :
useradd newuser
passwd newuser

```
[rhel6@localhost ~]$ su - root
Password:
[root@localhost ~]# useradd newuser
[root@localhost ~]# passwd newuser
Changing password for user newuser.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

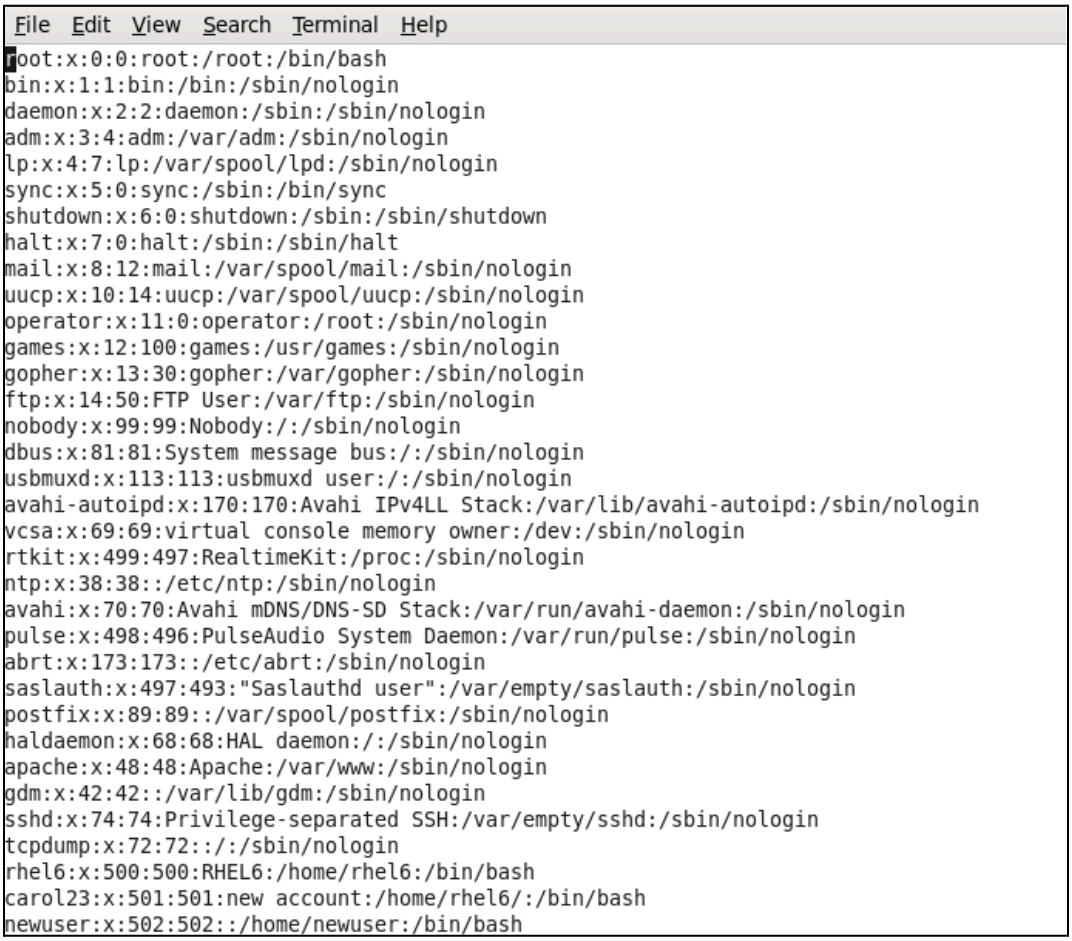
3. Then use the “ls -l” command to list the various directory contents of files in the root directory.

Code :
ls -l

```
[root@localhost ~]# ls -l
total 92
-rw-------. 1 root root  3343 Sep 29  2021 anaconda-ks.cfg
drwxr-xr-x. 2 root root  4096 Oct  4  2021 Desktop
drwxr-xr-x. 2 root root  4096 Oct  4  2021 Documents
drwxr-xr-x. 2 root root  4096 Oct  4  2021 Downloads
-rw-r--r--. 1 root root 38978 Sep 29  2021 install.log
-rw-r--r--. 1 root root 10064 Sep 29  2021 install.log.syslog
drwxr-xr-x. 2 root root  4096 Oct  4  2021 Music
drwxr-xr-x. 2 root root  4096 Oct  4  2021 Pictures
drwxr-xr-x. 2 root root  4096 Oct  4  2021 Public
drwxr-xr-x. 2 root root  4096 Oct  4  2021 Templates
drwxr-xr-x. 2 root root  4096 Oct  4  2021 Videos
```

4. Enter the administrative “/etc/password” file using the vim editor

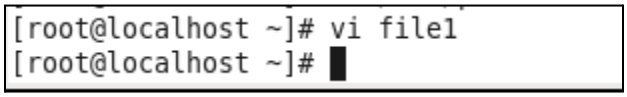
Code :
vi /etc/passwd
(Use :q to quit)



ii. Groups :

1. Create a new file using the vim editor named file1 in the root user.

Code :
vi file1



Enter some text into the file by using the
i key for insert
:w to write

:q to quit

2. Then copy the file created in the root to the new user's directory by using the cp command.

Code :

cp file1 /home/newuser

```
[root@localhost ~]# cp file1 /home/newuser
You have new mail in /var/spool/mail/root
[root@localhost ~]#
```

3. Change the terminal user to the newuser and list their directory contents

Code :

su - new user

ls -l

ls

```
[root@localhost ~]# su - newuser
[newuser@localhost ~]$ ls -l
total 4
-rw-r--r--. 1 root root 24 Mar 11 12:51 file1
[newuser@localhost ~]$ ls
file1
[newuser@localhost ~]$
```

4. Change it back to the root

Code :

su - root

```
[newuser@localhost ~]$ su - root
Password:
```

5. Then change the group of the file from root to the new user using the chgrp command.

Code :

chgrp newuser file1

```
[root@localhost ~]# chgrp newuser file1
[root@localhost ~]# ls -l
total 96
-rw-------. 1 root root     3343 Sep 29  2021 anaconda-ks.cfg
drwxr-xr-x. 2 root root     4096 Oct  4  2021 Desktop
drwxr-xr-x. 2 root root     4096 Oct  4  2021 Documents
drwxr-xr-x. 2 root root     4096 Oct  4  2021 Downloads
-rw-r--r--. 1 root newuser    24 Mar 11 12:18 file1
-rw-r--r--. 1 root root    38978 Sep 29  2021 install.log
-rw-r--r--. 1 root root    10064 Sep 29  2021 install.log.syslog
drwxr-xr-x. 2 root root     4096 Oct  4  2021 Music
drwxr-xr-x. 2 root root     4096 Oct  4  2021 Pictures
drwxr-xr-x. 2 root root     4096 Oct  4  2021 Public
drwxr-xr-x. 2 root root     4096 Oct  4  2021 Templates
drwxr-xr-x. 2 root root     4096 Oct  4  2021 Videos
```

6. Change the ownership of the file from root to the new user using the chown command.

Code :

chown newuser file1

```
[root@localhost ~]# chown newuser file1
[root@localhost ~]# ls -l
total 96
-rw-------. 1 root    root     3343 Sep 29  2021 anaconda-ks.cfg
drwxr-xr-x. 2 root    root     4096 Oct  4  2021 Desktop
drwxr-xr-x. 2 root    root     4096 Oct  4  2021 Documents
drwxr-xr-x. 2 root    root     4096 Oct  4  2021 Downloads
-rw-r--r--. 1 newuser newuser    24 Mar 11 12:18 file1
-rw-r--r--. 1 root    root    38978 Sep 29  2021 install.log
-rw-r--r--. 1 root    root    10064 Sep 29  2021 install.log.syslog
drwxr-xr-x. 2 root    root     4096 Oct  4  2021 Music
drwxr-xr-x. 2 root    root     4096 Oct  4  2021 Pictures
drwxr-xr-x. 2 root    root     4096 Oct  4  2021 Public
drwxr-xr-x. 2 root    root     4096 Oct  4  2021 Templates
drwxr-xr-x. 2 root    root     4096 Oct  4  2021 Videos
```

If you want to change the group of multiple files starting with the same letter then you can use a * after specifying the first letter of the files.

Code :
chgrp newuser f*

```
[root@localhost ~]# vi file2
[root@localhost ~]# chgrp newuser f*
[root@localhost ~]# ls -l
total 100
-rw-------. 1 root    root     3343 Sep 29  2021 anaconda-ks.cfg
drwxr-xr-x. 2 root    root     4096 Oct  4  2021 Desktop
drwxr-xr-x. 2 root    root     4096 Oct  4  2021 Documents
drwxr-xr-x. 2 root    root     4096 Oct  4  2021 Downloads
-rw-r--r--. 1 newuser newuser    24 Mar 11 12:18 file1
-rw-r--r--. 1 root    newuser     4 Mar 11 13:02 file2
-rw-r--r--. 1 root    root    38978 Sep 29  2021 install.log
-rw-r--r--. 1 root    root    10064 Sep 29  2021 install.log.syslog
drwxr-xr-x. 2 root    root     4096 Oct  4  2021 Music
drwxr-xr-x. 2 root    root     4096 Oct  4  2021 Pictures
drwxr-xr-x. 2 root    root     4096 Oct  4  2021 Public
drwxr-xr-x. 2 root    root     4096 Oct  4  2021 Templates
drwxr-xr-x. 2 root    root     4096 Oct  4  2021 Videos
[root@localhost ~]#
```

iii. Permissions

The chmod command is used to change the access mode of a file.

Code :
chmod 777 f*

Here, the permissions given to both the files i.e file1 and file2 is that the file is readable and executable by anybody on the system

```
[root@localhost ~]# chmod 777 f*
[root@localhost ~]# ls -l
total 100
-rw-------. 1 root    root     3343 Sep 29  2021 anaconda-ks.cfg
drwxr-xr-x. 2 root    root     4096 Oct  4  2021 Desktop
drwxr-xr-x. 2 root    root     4096 Oct  4  2021 Documents
drwxr-xr-x. 2 root    root     4096 Oct  4  2021 Downloads
-rwxrwxrwx. 1 newuser newuser    24 Mar 11 12:18 file1
-rwxrwxrwx. 1 root    newuser     4 Mar 11 13:02 file2
-rw-r--r--. 1 root    root    38978 Sep 29  2021 install.log
-rw-r--r--. 1 root    root    10064 Sep 29  2021 install.log.syslog
drwxr-xr-x. 2 root    root     4096 Oct  4  2021 Music
drwxr-xr-x. 2 root    root     4096 Oct  4  2021 Pictures
drwxr-xr-x. 2 root    root     4096 Oct  4  2021 Public
drwxr-xr-x. 2 root    root     4096 Oct  4  2021 Templates
drwxr-xr-x. 2 root    root     4096 Oct  4  2021 Videos
```

Practical 3

Initial settings: Add a User, Network Settings, change to static IP address, Disable IPv6 if not needed, Configure Services, display the list of services which are running, Stop and turn OFF auto-start setting for a service if you don’t need it, Sudo Settings

Practical 4

SSH Server: Password Authentication : Configure SSH Server to manage a server from the remote computer, SSH Client

Steps :

1. Open the Linux Terminal and go to root user and check if ssh is installed.

Code :

su - root

rpm -qa openssh*

```
[root@localhost ~]# su - root
[root@localhost ~]# rpm -qa openssh*
openssh-server-5.3p1-52.el6.i686
openssh-clients-5.3p1-52.el6.i686
openssh-5.3p1-52.el6.i686
openssh-askpass-5.3p1-52.el6.i686
[root@localhost ~]#
```

2. Then check sshd service status using the service status command.

Code :

service sshd status

```
[root@localhost ~]# service sshd status
openssh-daemon (pid  2014) is running...
```

3. Restart sshd service by using the service restart command.

Code :

service sshd restart

```
[root@localhost ~]# service sshd restart
Stopping sshd:                                             [  OK  ]
Starting sshd:                                             [  OK  ]
```

4. Use the ifconfig command to get the ip address

Code :

ifconfig

```
[root@localhost ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:2F:C9:1B
          inet addr:192.168.1.3  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe2f:c91b/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:334 errors:0 dropped:0 overruns:0 frame:0
          TX packets:301 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:34651 (33.8 KiB)  TX bytes:36563 (35.7 KiB)
          Interrupt:19 Base address:0x2024

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:155 errors:0 dropped:0 overruns:0 frame:0
          TX packets:155 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:9649 (9.4 KiB)  TX bytes:9649 (9.4 KiB)
```

5. Check connectivity from SSH server using the ping command

Code :
ping <ip address>

```
[root@localhost ~]# ping 192.168.1.3
PING 192.168.1.3 (192.168.1.3) 56(84) bytes of data.
64 bytes from 192.168.1.3: icmp_seq=1 ttl=64 time=0.163 ms
64 bytes from 192.168.1.3: icmp_seq=2 ttl=64 time=0.040 ms
64 bytes from 192.168.1.3: icmp_seq=3 ttl=64 time=0.034 ms
64 bytes from 192.168.1.3: icmp_seq=4 ttl=64 time=0.030 ms
64 bytes from 192.168.1.3: icmp_seq=5 ttl=64 time=0.059 ms
64 bytes from 192.168.1.3: icmp_seq=6 ttl=64 time=0.045 ms
64 bytes from 192.168.1.3: icmp_seq=7 ttl=64 time=0.087 ms
64 bytes from 192.168.1.3: icmp_seq=8 ttl=64 time=0.062 ms
64 bytes from 192.168.1.3: icmp_seq=9 ttl=64 time=0.058 ms
64 bytes from 192.168.1.3: icmp_seq=10 ttl=64 time=0.071 ms
64 bytes from 192.168.1.3: icmp_seq=11 ttl=64 time=0.031 ms
64 bytes from 192.168.1.3: icmp_seq=12 ttl=64 time=0.427 ms
64 bytes from 192.168.1.3: icmp_seq=13 ttl=64 time=0.121 ms
64 bytes from 192.168.1.3: icmp_seq=14 ttl=64 time=0.084 ms
64 bytes from 192.168.1.3: icmp_seq=15 ttl=64 time=0.031 ms
64 bytes from 192.168.1.3: icmp_seq=16 ttl=64 time=0.060 ms
64 bytes from 192.168.1.3: icmp_seq=17 ttl=64 time=0.060 ms
64 bytes from 192.168.1.3: icmp_seq=18 ttl=64 time=0.047 ms
64 bytes from 192.168.1.3: icmp_seq=19 ttl=64 time=0.075 ms
```

6. Add two users and set the password for both of them.

Code :
useradd user1
passwd user1 (pass is user1@123)

useradd user2
passwd user2 (pass is user2@123)

```
[root@localhost ~]# useradd user1
[root@localhost ~]# passwd user1
Changing password for user user1.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@localhost ~]# useradd user2
[root@localhost ~]# passwd user2
Changing password for user user2.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@localhost ~]#
```

7. Open the ssh configuration file i.e sshd_config

Code :
vi /etc/ssh/sshd_config

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

Check the value of the PasswordAuthentication directive. In order to accept local user password based authentication it must be set to yes. Set it to yes if it is set to no and save the file.

```
# To disable tunneled clear text passwords, change to no here!
#PasswordAuthentication yes
#PermitEmptyPasswords no
PasswordAuthentication yes
```

8. Restart the service if you have made any change in sshd_config

Code :
service sshd restart

```
[root@localhost ~]# service sshd restart
Stopping sshd:                                             [  OK  ]
Starting sshd:                                             [  OK  ]
```

9. Log in as the first user by this command and provide the password when asked.

Code :
ssh (username)@(ipaddress):

```
[root@localhost ~]# ssh user1@192.168.1.3
The authenticity of host '192.168.1.3 (192.168.1.3)' can't be established.
RSA key fingerprint is ee:ae:76:0f:4c:ee:ea:ce:38:d8:39:1e:6e:e4:e3:ba.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.1.3' (RSA) to the list of known hosts.
user1@192.168.1.3's password:
[user1@localhost ~]$
```

10. Confirm that you have logged in to the user by using the who am i command.

Code :
who am i

```
[user1@localhost ~]$ who am i
user1    pts/1        2024-03-12 12:15 (192.168.1.3)
[user1@localhost ~]$
```

11. Exit the user1 and perform the same commands for the second user i.e user2.

Code :
exit
ssh (username)@(ipaddress):
who am i
exit

```
[user1@localhost ~]$ exit
logout
Connection to 192.168.1.3 closed.
[root@localhost ~]# ssh user2@192.168.1.3
user2@192.168.1.3's password:
[user2@localhost ~]$ who am i
user2    pts/1        2024-03-12 12:17 (192.168.1.3)
[user2@localhost ~]$ exit
logout
Connection to 192.168.1.3 closed.
[root@localhost ~]#
```

User and Host based security :
12. In order to make the server more restrictive, open the ssh configuration file once again

Code :
vi /etc/ssh/sshd_config

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

13. Add the following lines at the end of the file and save it using the :wq command.

PermitRootLogin no
DenyUsers user1

```
# Example of overriding settings on a per-user basis
#Match User anoncvs
#       X11Forwarding no
#       AllowTcpForwarding no
#       ForceCommand cvs server

PermitRootLogin no
DenyUsers user1

-- INSERT --
```

14. Restart the ssh server and try logging in using user one and root.

Code :
service sshd restart
ssh (username)@(ipaddress):

```
[root@localhost ~]# service sshd restart
Stopping sshd:                                             [  OK  ]
Starting sshd:                                             [  OK  ]
[root@localhost ~]# ssh user1@192.168.1.3
user1@192.168.1.3's password:
Permission denied, please try again.
user1@192.168.1.3's password:
Permission denied, please try again.
user1@192.168.1.3's password:
Permission denied (publickey,gssapi-keyex,gssapi-with-mic,password).
```

ssh root@ipaddress

```
[root@localhost ~]# ssh root@192.168.1.3
root@192.168.1.3's password:
Permission denied, please try again.
root@192.168.1.3's password:
Permission denied, please try again.
root@192.168.1.3's password:
Permission denied (publickey,gssapi-keyex,gssapi-with-mic,password).
```

15. Next, try with user2 and it allows you to login

Code :
ssh user2@ipaddress
exit

```
[root@localhost ~]# ssh user2@192.168.1.3
user2@192.168.1.3's password:
Last login: Tue Mar 12 12:17:31 2024 from 192.168.1.3
[user2@localhost ~]$ exit
logout
Connection to 192.168.1.3 closed.
[root@localhost ~]#
```

Public key :

16. Open the configuration file and uncomment following directives and save the file
RSAAuthentication yes
PubkeyAuthentication yes
AuthorizedKeysFile .ssh/authorized_keys

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

```
RSAAuthentication yes
PubkeyAuthentication yes
AuthorizedKeysFile .ssh/authorized_keys
#AuthorizedKeysCommand none
#AuthorizedKeysCommandRunAs nobody
```

17. Restart the sshd service using the service restart command

Code :
service sshd restart

```
[root@localhost ~]# service sshd restart
Stopping sshd:                                             [  OK  ]
Starting sshd:                                             [  OK  ]
```

18. Login to the server from user2

```
[root@localhost ~]# ssh user2@192.168.1.3
user2@192.168.1.3's password:
Last login: Tue Mar 12 12:25:05 2024 from 192.168.1.3
[user2@localhost ~]$
```

Create a ssh directory with permission 755

Code :
mkdir ~/.ssh
chmod 755 ~/.ssh
exit

```
[user2@localhost ~]$ mkdir ~/.ssh
[user2@localhost ~]$ chmod 755 ~/.ssh
[user2@localhost ~]$ exit
logout
Connection to 192.168.1.3 closed.
[root@localhost ~]#
```

19. Generate the public/private key pair using the ssh-keygen command. Then press enter to accept the default location for the key file

Code :
ssh-keygen -t rsa

```
[user2@localhost ~]$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/user2/.ssh/id_rsa):
```

20. Enter the passphrase “I love linux” and confirm the passphrase

```
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/user2/.ssh/id_rsa.
Your public key has been saved in /home/user2/.ssh/id_rsa.pub.
The key fingerprint is:
9b:50:c2:6e:0d:8f:ba:17:61:b9:a9:08:5b:cc:20:48 user2@localhost.localdomain
The key's randomart image is:
+--[ RSA 2048]----+
|                 |
| E   .           |
|o     +..        |
|+    .+B         |
|.+   .=+S        |
|. +  o+. o       |
| + ... .o        |
|. . ...          |
|    ..           |
+-----------------+
```

21. The public key is stored in the /home/user2/.ssh/id_rsa.pub. Create a copy of the public key. Then Copy the authorized_keys file on the server to /home/user2/.ssh/authorized_keys. Enter user2 [user account on server] password when asked.

Code :
cat ~/.ssh/id_rsa.pub >> authorized_keys
scp authorized_keys user2@192.168.1.3:/home/user2/.ssh/

```
[user2@localhost ~]$ cat ~/.ssh/id_rsa.pub >> authorized_keys
[user2@localhost ~]$ scp authorized_keys user2@192.168.1.3:/home/user2/.ssh/
The authenticity of host '192.168.1.3 (192.168.1.3)' can't be established.
RSA key fingerprint is ee:ae:76:0f:4c:ee:ea:ce:38:d8:39:1e:6e:e4:e3:ba.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.1.3' (RSA) to the list of known hosts.
user2@192.168.1.3's password:
authorized_keys                               100%  409     0.4KB/s   00:00
```

Then exit out of the server.

```
[user2@localhost ~]$ exit
logout
Connection to 192.168.1.3 closed.
```

22. Go to the user2 and set the permission to 644 for authorized_keys

Code :
su user2
chmod 644 ~/.ssh/authorized_keys

```
[root@localhost ~]# su user2
[user2@localhost root]$ chmod 644 ~/.ssh/authorized_keys
```

23. Go to the root user and restart the service

Code :
service sshd restart

```
[root@localhost ~]# service sshd restart
Stopping sshd:                                             [  OK  ]
Starting sshd:                                             [  OK  ]
```

24. Login into the ssh client from user2 and then enter the passphrase “I love linux”

Code :
ssh user2@192.168.1.3
ssh -l user2 192.168.1.3

```
[root@localhost ~]# ssh user2@192.168.1.3
user2@192.168.1.3's password:
Last login: Fri Mar 15 11:28:50 2024 from 192.168.1.3
[user2@localhost ~]$ ssh -l user2 192.168.1.3
Enter passphrase for key '/home/user2/.ssh/id_rsa':
Last login: Fri Mar 15 11:48:09 2024 from 192.168.1.3
```

Then exit

```
[user2@localhost ~]$ exit
logout
Connection to 192.168.1.3 closed.
[user2@localhost ~]$
```

25. Go to the root user and open the configuration file

Code :
su - root

vi /etc/ssh/sshd_config

```
[user2@localhost ~]$ su - root
Password:
[root@localhost ~]# vi /etc/ssh/sshd_config
```

26. Uncomment the following line and change its value to 2223

Port 22

```
Port 2223
```

27. Restart the service using the service restart command

Code :

service sshd restart

```
[root@localhost ~]# service sshd restart
Stopping sshd:                                             [  OK  ]
Starting sshd:                                             [  OK  ]
```

28. Now try to connect to the ssh client with user 2 and the connection will be refused

Code :

ssh -l user2 192.168.1.3

```
[root@localhost ~]# ssh -l user2 192.168.1.3
ssh: connect to host 192.168.1.3 port 22: Connection refused
```

29. Try connecting the ssh client now by specifying the new port

Code :

ssh -l user2 192.168.1.3 -p 2223

```
[root@localhost ~]# ssh -l user2 192.168.1.3 -p 2223
user2@192.168.1.3's password:
Last login: Fri Mar 15 11:48:28 2024 from 192.168.1.3
[user2@localhost ~]$ ssh -l user2 192.168.1.3 -p 2223
Enter passphrase for key '/home/user2/.ssh/id_rsa':
Last login: Fri Mar 15 12:02:03 2024 from 192.168.1.3
[user2@localhost ~]$
```

Practical 5
Installing and Configure of FTP server

Steps :

1. Go to the root user and then navigate to the Packages directory

Code :
su - root
cd /media/RHEL….. 1/Packages/

```
[rhel6@localhost ~]$ su - root
Password:
[root@localhost ~]# cd /media/RHEL_6.1\ i386\ Disc\ 1/Packages/
[root@localhost Packages]#
```

2. Check if the vsftpd and the ftp packages are installed. If not, then install the vsftpd and ftp package

Code :
rpm -qa | grep vsftpd
rpm -ivh vsftpd*
rpm -ivh ftp*

```
[root@localhost Packages]# rpm -qa | grep vsftpd
[root@localhost Packages]# rpm -ivh vsftpd*
warning: vsftpd-2.2.2-6.el6_0.1.i686.rpm: Header V3 RSA/SHA256 Signature, key ID
 fd431d51: NOKEY
Preparing...                ########################################### [100%]
   1:vsftpd                 ########################################### [100%]
[root@localhost Packages]# rpm -ivh ftp*
warning: ftp-0.17-51.1.el6.i686.rpm: Header V3 RSA/SHA256 Signature, key ID fd43
1d51: NOKEY
Preparing...                ########################################### [100%]
   1:ftp                    ########################################### [100%]
[root@localhost Packages]#
```

3. Check if vsftpd and ftp is installed once again

Code :
rpm -qa | grep vsftpd

```
[root@localhost Packages]# rpm -qa | grep vsftpd
vsftpd-2.2.2-6.el6_0.1.i686
[root@localhost Packages]#
```

rpm -qa | grep ftp

```
[root@localhost Packages]# rpm -qa | grep ftp
report-config-ftp-0.18-9.el6.i686
vsftpd-2.2.2-6.el6_0.1.i686
report-plugin-ftp-0.18-9.el6.i686
ftp-0.17-51.1.el6.i686
gvfs-obexftp-1.4.3-12.el6.i686
```

4. Use the chkonfig command to start the vsftpd services at boot time

Code :
chkconfig vsftpd on
chkconfig –list | grep ftp

```
[root@localhost Packages]# chkconfig vsftpd on
[root@localhost Packages]# chkconfig --list | grep ftp
vsftpd          0:off   1:off   2:on    3:on    4:on    5:on    6:off
[root@localhost Packages]#
```

5. Then navigate to the /var/ftp/pub/ directory using the cd command and create a file for testing ftp.

Code :
cd /var/ftp/pub/
cat -> ftpfile

(ctrl + d to save and exit)

```
[root@localhost Packages]# cd /var/ftp/pub/
[root@localhost pub]# cat -> ftpfile
hi..
This is my ftp file for testing
[root@localhost pub]#
```

To check contents of file
cat ftpfile

```
[root@localhost pub]# cat ftpfile
hi..
This is my ftp file for testing
```

6. Verify IP address of linux machine to be configured as FTP.

```
[root@localhost pub]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:2F:C9:1B
          inet addr:192.168.80.130  Bcast:192.168.80.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe2f:c91b/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:306 errors:0 dropped:0 overruns:0 frame:0
          TX packets:413 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:37230 (36.3 KiB)  TX bytes:45615 (44.5 KiB)
          Interrupt:19 Base address:0x2024

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:10 errors:0 dropped:0 overruns:0 frame:0
          TX packets:10 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:584 (584.0 b)  TX bytes:584 (584.0 b)
```

7. Open configuration file and make following changes

i. Uncomment anonymous_enable = YES

ii. Uncomment local_enable = YES

iii.Uncomment anonymous_upload_enable = YES

iv. Uncomment listen = YES

Code :

vi /etc/vsftpd/vsftpd/conf

```
[root@localhost pub]# vi /etc/vsftpd/vsftpd.conf
```

```
# Example config file /etc/vsftpd/vsftpd.conf
#
# The default compiled in settings are fairly paranoid. This sample file
# loosens things up a bit, to make the ftp daemon more usable.
# Please see vsftpd.conf.5 for all compiled in defaults.
#
# READ THIS: This example file is NOT an exhaustive list of vsftpd options.
# Please read the vsftpd.conf.5 manual page to get a full idea of vsftpd's
# capabilities.
#
# Allow anonymous FTP? (Beware - allowed by default if you comment this out).
anonymous_enable=YES
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
write_enable=YES
#
# Default umask for local users is 077. You may wish to change this to 022,
# if your users expect that (022 is used by most other ftpd's)
local_umask=022
#
# Uncomment this to allow the anonymous FTP user to upload files. This only
"/etc/vsftpd/vsftpd.conf" 118L, 4494C
```

```
# Uncomment this to allow the anonymous FTP user to upload files. This only
# has an effect if the above global write enable is activated. Also, you will
# obviously need to create a directory writable by the FTP user.
anon_upload_enable=YES
```

```
# When "listen" directive is enabled, vsftpd runs in standalone mode and
# listens on IPv4 sockets. This directive cannot be used in conjunction
# with the listen_ipv6 directive.
listen=YES
```

8. Then restart the vsftpd service using the service restart command.

Code :
service vsftpd restart
service vsftpd status
service vsftpd restart

```
[root@localhost pub]# service vsftpd restart
Shutting down vsftpd:                                      [FAILED]
Starting vsftpd for vsftpd:                                [  OK  ]
[root@localhost pub]# service vsftpd status
vsftpd (pid 5338) is running...
[root@localhost pub]# service vsftpd restart
Shutting down vsftpd:                                      [  OK  ]
Starting vsftpd for vsftpd:                                [  OK  ]
[root@localhost pub]#
```

9. Go back to the main directory using the cd command and then login with an anonymous user.

Code :
cd
ftp <ip address>
Name : anonymous
Password : (just click enter)

```
[root@localhost pub]# cd
[root@localhost ~]# ftp 192.168.80.130
Connected to 192.168.80.130 (192.168.80.130).
220 (vsFTPd 2.2.2)
Name (192.168.80.130:rhel6): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

Then, use ls –a command to view the content of ftp home directory.

Code :
ls -a
bye (to exit out of ftp)

```
ftp> ls -a
227 Entering Passive Mode (192,168,80,130,207,136).
150 Here comes the directory listing.
drwxr-xr-x    3 0        0            4096 Mar 11 21:33 .
drwxr-xr-x    3 0        0            4096 Mar 11 21:33 ..
drwxr-xr-x    2 0        0            4096 Mar 11 21:40 pub
226 Directory send OK.
ftp> bye
221 Goodbye.
[root@localhost ~]#
```

10. Now allow ftp anonymous write enable by using the setsebool command

Code :
getsebool –a | grep ftp
setsebool –P allow_ftp_annon_write on or = 1
getsebool –a | grep ftp

```
[root@localhost ~]# getsebool -a | grep ftp
allow_ftpd_anon_write --> off
allow_ftpd_full_access --> off
allow_ftpd_use_cifs --> off
allow_ftpd_use_nfs --> off
ftp_home_dir --> off
ftpd_connect_db --> off
httpd_enable_ftp_server --> off
sftpd_anon_write --> off
sftpd_enable_homedirs --> off
sftpd_full_access --> off
sftpd_write_ssh_home --> off
tftp_anon_write --> off
[root@localhost ~]# setsebool -P allow_ftpd_anon_write on

[root@localhost ~]#
[root@localhost ~]# getsebool -a | grep ftp
allow_ftpd_anon_write --> on
allow_ftpd_full_access --> off
allow_ftpd_use_cifs --> off
allow_ftpd_use_nfs --> off
ftp_home_dir --> off
ftpd_connect_db --> off
httpd_enable_ftp_server --> off
sftpd_anon_write --> off
sftpd_enable_homedirs --> off
sftpd_full_access --> off
sftpd_write_ssh_home --> off
tftp_anon_write --> off
```

11. Then allow system users to get access to ftp server by turning the ftp_home_dir on using the setsebool command.

Code :
getsebool –a | grep ftp
setsebool –P ftp_home_dir on
getsebool –a | grep ftp

```
[root@localhost ~]# getsebool -a | grep ftp
allow_ftpd_anon_write --> on
allow_ftpd_full_access --> off
allow_ftpd_use_cifs --> off
allow_ftpd_use_nfs --> off
ftp_home_dir --> off
ftpd_connect_db --> off
httpd_enable_ftp_server --> off
sftpd_anon_write --> off
sftpd_enable_homedirs --> off
sftpd_full_access --> off
sftpd_write_ssh_home --> off
tftp_anon_write --> off
[root@localhost ~]# setsebool -P ftp_home_dir on
[root@localhost ~]# getsebool -a | grep ftp
allow_ftpd_anon_write --> on
allow_ftpd_full_access --> off
allow_ftpd_use_cifs --> off
allow_ftpd_use_nfs --> off
ftp_home_dir --> on
ftpd_connect_db --> off
httpd_enable_ftp_server --> off
sftpd_anon_write --> off
sftpd_enable_homedirs --> off
sftpd_full_access --> off
sftpd_write_ssh_home --> off
tftp_anon_write --> off
[root@localhost ~]#
```

12. By default the /var/ftp is in the ftp root user's home directory. Check the context of file /var/ftp/pub and change the group and owner to ftp.

Code :
ls –ldZ /var/ftp/pub
chgrp ftp /var/ftp/pub
chown ftp /var/ftp/pub
ls –ldZ /var/ftp/pub

```
[root@localhost ~]# ls -ldZ /var/ftp/pub
drwxr-xr-x. root root system_u:object_r:public_content_t:s0 /var/ftp/pub
[root@localhost ~]# chgrp ftp /var/ftp/pub
[root@localhost ~]# chown ftp /var/ftp/pub
[root@localhost ~]# ls -ldZ /var/ftp/pub
drwxr-xr-x. ftp ftp system_u:object_r:public_content_t:s0 /var/ftp/pub
```

13. Go to the pub directory using the cd command and create a file named ftptest and enter some text.

Code :
cd /var/ftp/pub
touch T1 T2 T3
cat > ftptest

(to save and exit : ctrl +d)

```
[root@localhost ~]# cd /var/ftp/pub
[root@localhost pub]# touch T1 T2 T3
[root@localhost pub]# cat > ftptest
hi..
This file is for FTP server testing[root@localhost pub]#
[root@localhost pub]#
```

15.Fo to the Packages directory and then restart the service of vsftpd and enable it from boot.

Code :
Cd /media/RHEL_1/Packages/
service vsftpd start
service vsftpd restart
chkconfig vsftpd on
chkconfig –list | grep vsftpd

```
[root@localhost ~]# cd /media/RHEL_6.1\ i386\ Disc\ 1/Packages/
[root@localhost Packages]# service vsftpd start
Starting vsftpd for vsftpd:                                [FAILED]
[root@localhost Packages]# service vsftpd restart
Shutting down vsftpd:                                      [  OK  ]
Starting vsftpd for vsftpd:                                [  OK  ]
[root@localhost Packages]# chkconfig vsftpd on
[root@localhost Packages]# chkconfig --list | grep vsftpd
vsftpd          0:off   1:off   2:on    3:on    4:on    5:on    6:off
[root@localhost Packages]#
```

Now FTP has been configured.

16. Now, we do the testing as an FTP client

Code :
ftp <ip address>
Name : anonymous
Password : (just click enter)

Here, it allows anonymous user to log in

```
[root@localhost Packages]# ftp 192.168.80.130
Connected to 192.168.80.130 (192.168.80.130).
220 (vsFTPd 2.2.2)
Name (192.168.80.130:rhel6): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> bye
221 Goodbye.
```

17. Then, we disable the anonymous user to use FTP login
Open configuration file.

Code :
cd /var/ftp/pub
vi /etc/vsftpd/vsftpd.conf

i) Go to directive anonymous _enable = YES and make it anonymous _enable = NO.
ii) Go to directive anonymous_upload_enable = YES and make it anonymous_upload_enable = NO.

(:wq to save and quit)

```
# Allow anonymous FTP? (Beware - allowed by default if you comment this out).
anonymous_enable=NO
#
```

```
# Uncomment this to allow the anonymous FTP user to upload files. This only
# has an effect if the above global write enable is activated. Also, you will
# obviously need to create a directory writable by the FTP user.
anon_upload_enable=NO
```

18. Restart the vsftpd service by used the service restart command

Code :
service vsftpd restart

```
[root@localhost Packages]# service vsftpd restart
Shutting down vsftpd:                                      [  OK  ]
Starting vsftpd for vsftpd:                                [  OK  ]
[root@localhost Packages]#
```

19. Then, try logging into ftp as an anonymous user.

Code :
ftp <ip address>
Name : anonymous
Password : (just click enter)

Login will be incorrect

```
[root@localhost pub]# ftp 192.168.80.130
Connected to 192.168.80.130 (192.168.80.130).
220 (vsFTPd 2.2.2)
Name (192.168.80.130:rhel6): anonymous
331 Please specify the password.
Password:

530 Login incorrect.
Login failed.
```

Practical 6
Configure DHCP (Dynamic Host Configuration Protocol) Server

Steps :

1. Open the Linux Terminal, then go to the root user and navigate to the Packages directory

Code :
su - root
cd /media/RHEL….. 1/Packages/

```
[root@localhost ~]# su - root
[root@localhost ~]# rpm -qa dhcp
[root@localhost ~]# cd /media/RHEL_6.1\ i386\ Disc\ 1/Packages/
```

2. Check if the dhcp package is installed. If not, then install the dhcp package

Code :
rpm -qa | grep dhcp
rpm -ivh dhcp*

```
[root@localhost Packages]# rpm -ivh dhcp*
warning: dhcp-4.1.1-19.P1.el6.i686.rpm: Header V3 RSA/SHA256 Signature, key ID fd431d51: NOKEY
Preparing...                ########################################### [100%]
   1:dhcp                   ########################################### [100%]
```

3. Check if the dhcp package is installed once again.

Code :
rpm -qa | grep sdhcp

```
[root@localhost Packages]# rpm -qa | grep dhcp
dhcp-4.1.1-19.P1.el6.i686
[root@localhost Packages]#
```

4. Check hostname of the linux system

Code :
hostname

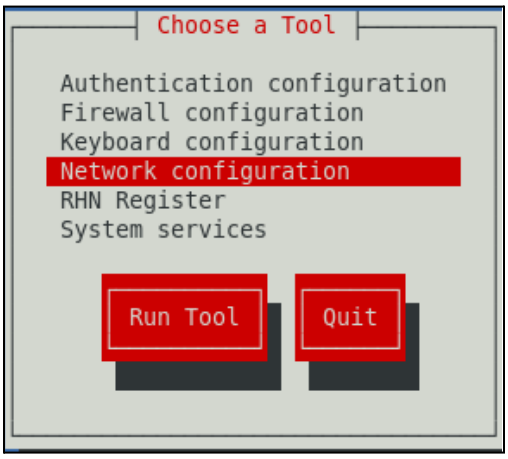
```
[root@localhost Packages]# hostname
localhost.localdomain
[root@localhost Packages]#
```

5. Now check dhcpd service in system service it should be on

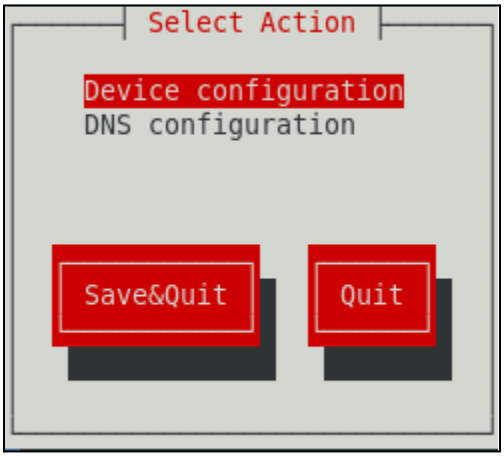
Code :
setup

```
[root@localhost Packages]# setup
```

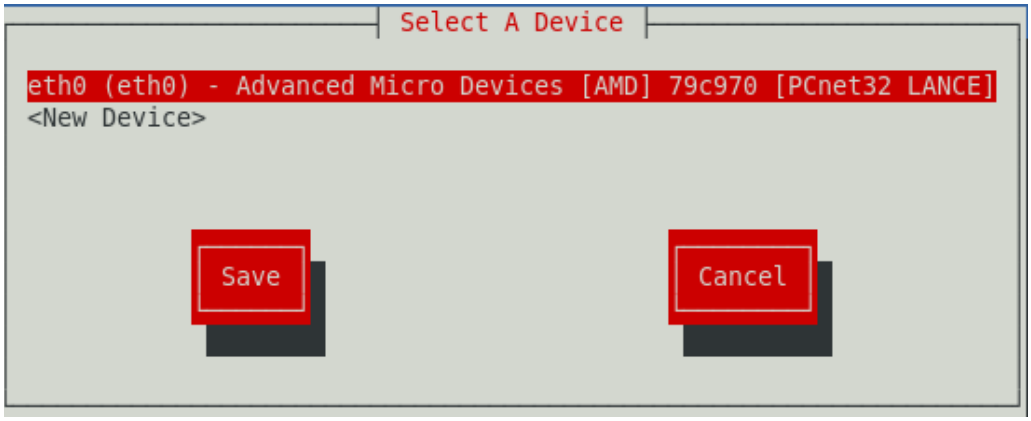
Select Network configuration and click Enter.



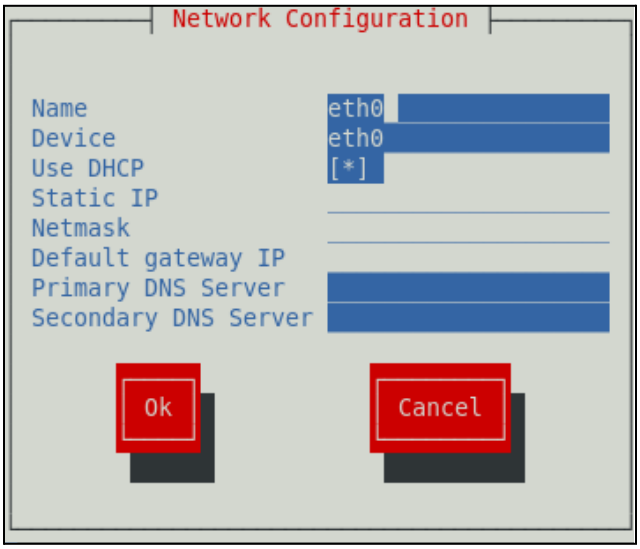
Then select Device configuration and click Enter.



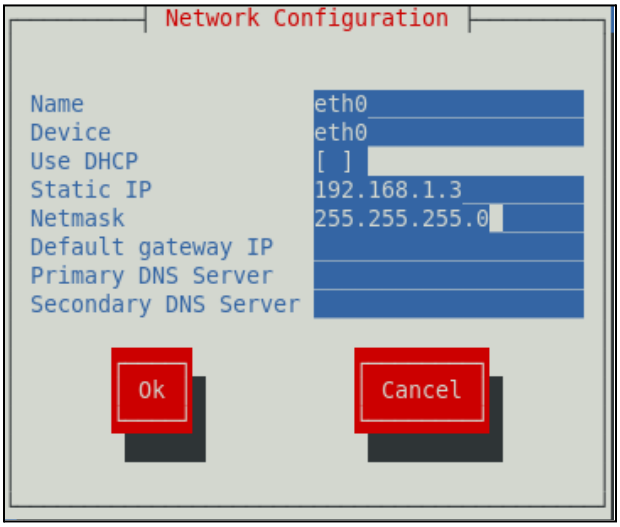
Select the LAN card (here it is eth0)



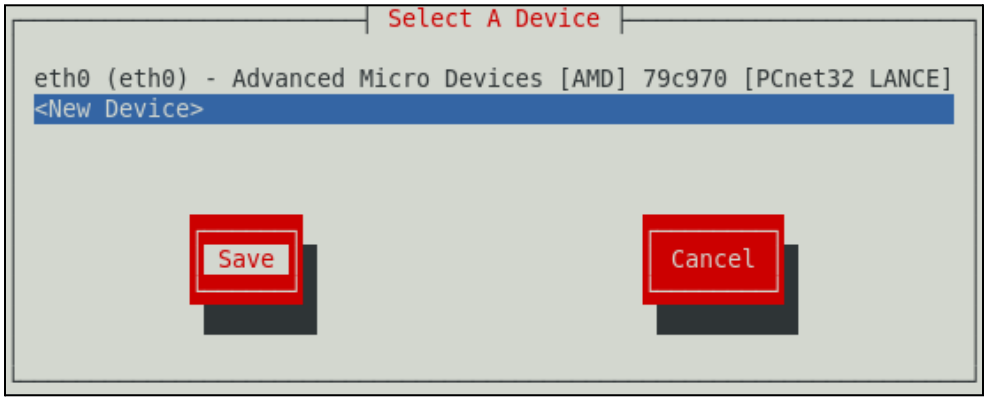
Select “Use DHCP” option and remove the * in between the brackets.



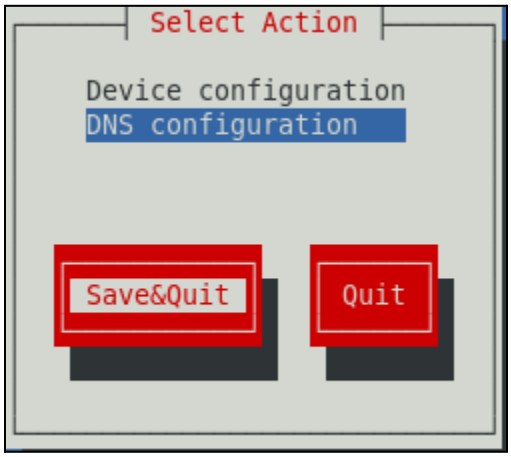
Then enter the IP address (192.168.1.3) as well as the Netmask (255.255.255.0). Then click on OK



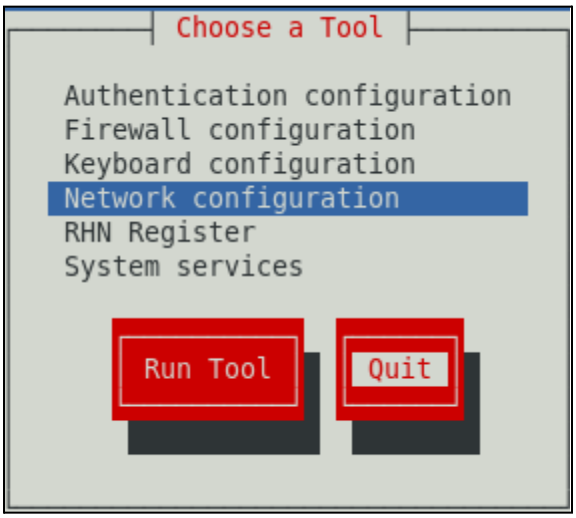
Then click on Save



Then click on “Save&Quit”



Then click on Quit



6. Now open the main configuration file of dhcp

Code :

vi /etc/dhcp/dhcpd.conf

```
[root@localhost Packages]# vi /etc/dhcp/dhcpd.conf
[root@localhost Packages]#
```

```
File  Edit  View  Search  Terminal  Help
#
# DHCP Server Configuration file.
#   see /usr/share/doc/dhcp*/dhcpd.conf.sample
#   see 'man 5 dhcpd.conf'
```

7. Copy the /usr/share/doc/dhcp-4.1.1/dhcpd.conf.sample file to the configuration file using the cp command.

Code :

cp /usr/share/doc/dhcp-4.1.1/dhcpd.conf.sample file /etc/dhcp/dhcpd.conf

```
[root@localhost Packages]# cp /usr/share/doc/dhcp-4.1.1/dhcpd.conf.sample /etc/dhcp/dhcpd.conf
cp: overwrite `/etc/dhcp/dhcpd.conf'? Y
[root@localhost Packages]#
```

8. Now open the configuration file once again

Code :

vi /etc/dhcp/dhcpd.conf

```
[root@localhost Packages]# vi /etc/dhcp/dhcpd.conf
[root@localhost Packages]#
```

```
# dhcpd.conf
#
# Sample configuration file for ISC dhcpd
#

# option definitions common to all supported networks...
option domain-name "example.org";
option domain-name-servers ns1.example.org, ns2.example.org;

default-lease-time 600;
max-lease-time 7200;

# Use this to enble / disable dynamic dns updates globally.
#ddns-update-style none;

# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
#authoritative;

# Use this to send dhcp log messages to a different log file (you also
# have to hack syslog.conf to complete the redirection).
log-facility local7;

# No service will be given on this subnet, but declaring it helps the
# DHCP server to understand the network topology.

subnet 10.152.187.0 netmask 255.255.255.0 {
}

# This is a very basic subnet declaration.

subnet 10.254.239.0 netmask 255.255.255.224 {
  range 10.254.239.10 10.254.239.20;
  option routers rtr-239-0-1.example.org, rtr-239-0-2.example.org;
}

# This declaration allows BOOTP clients to get dynamic addresses,
# which we don't really recommend.
"/etc/dhcp/dhcpd.conf" 104L, 3262C
```

9. Then uncomment line number 18

```
16 # If this DHCP server is the official DHCP server for the local
17 # network, the authoritative directive should be uncommented.
18 #authoritative;
```

10. Then comment line number 27 and line number 28

```
27 #subnet 10.152.187.0 netmask 255.255.255.0 {
28 #}
```

Change these lines number 32 and 33 to
Subnet 198.168.1.0 netmask 255.255.255.0
{
Range 192.168.1.10 192.168.1.20;
}

```
32 subnet 192.168.1.0 netmask 255.255.255.0 {
33   range 192.168.1.10 192.168.1.20;
```

Comment line number 34

```
34  # option routers rtr-239-0-1.example.org, rtr-239-0-2.example.org;
35 }
```

Then save the file

11. Then start and restart the dhcpd service using the service start and service restart commands

Code :
service dhcpd status
service dhcpd start
service dhcpd restart

```
[root@localhost Packages]# service dhcpd status
dhcpd is stopped
[root@localhost Packages]# service dhcpd start
Starting dhcpd:                                            [  OK  ]
[root@localhost Packages]# service dhcpd restart
Shutting down dhcpd:                                       [  OK  ]
Starting dhcpd:                                            [  OK  ]
```

12. Use the chkonfig command to start the dhcpd services at boot time

Code :
chkconfig –list dhcp
chkconfig dhcp on
chkconfig –list dhcp

```
[root@localhost Packages]# chkconfig --list dhcpd
dhcpd           0:off   1:off   2:off   3:off   4:off   5:off   6:off
[root@localhost Packages]# chkconfig dhcpd on
[root@localhost Packages]# chkconfig --list dhcpd
dhcpd           0:off   1:off   2:on    3:on    4:on    5:on    6:off
[root@localhost Packages]#
```

Practical 7
Install and configure NFS server.

Steps :

1. Go to the root user and then navigate to the Packages directory. Check if the nfs package is installed

Code :
su - root
cd /media/RHEL….. 1/Packages/
rpm -qa | grep nfs

```
[root@localhost ~]# su - root
[root@localhost ~]# cd /media/RHEL_6.1\ i386\ Disc\ 1/Packages/
[root@localhost Packages]# rpm -qa | grep nfs
```

2. If packages are not installed, then install the nfs package

Code :
rpm -ivh nfs*

```
[root@localhost Packages]# rpm -ivh nfs*
warning: nfs4-acl-tools-0.3.3-5.el6.i686.rpm: Header V3 RSA/SHA256 Signature, key ID fd431d51: NOKEY
Preparing...                ########################################### [100%]
        package nfs4-acl-tools-0.3.3-5.el6.i686 is already installed
[root@localhost Packages]#
```

3. Check if the nfs package installed once again

Code :
rpm -qa | grep nfs

```
[root@localhost Packages]# rpm -qa | grep nfs
nfs4-acl-tools-0.3.3-5.el6.i686
```

4. Verify IP address of the linux machine to be setup as NFS Server:

Code :
ifconfig eth0

```
[root@localhost Packages]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0C:29:2F:C9:1B
          inet addr:192.168.1.3  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe2f:c91b/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:278 errors:0 dropped:0 overruns:0 frame:0
          TX packets:267 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:29686 (28.9 KiB)  TX bytes:31863 (31.1 KiB)
          Interrupt:19 Base address:0x2024
```

5. Make a directory to be exported, create few files into it and give it full permission

Code :
cd /home/
mkdir servernfs
cd servernfs
cat > newfilenfs
(enter text)

(ctrl + d to save and exit)

```
[root@localhost Packages]# cd /home/
[root@localhost home]# mkdir servernfs
[root@localhost home]# cd servernfs
[root@localhost servernfs]# cat > newfilenfs
hello nfs file
[root@localhost servernfs]#
```

6. Open the configuration file of NFS, i.e, /etc/exports

Code :
vi /etc/exports

```
[root@localhost servernfs]# vi /etc/exports
```

Write the following lines in the file :
/home/servernfs * (rw,sync)

```
/home/servernfs * (rw,sync)

"/etc/exports" 1L, 28C
```

The above entry says that server export directory has been exported to the network 192.168.1.3

7. Save and quit the file. (shift + zz)
Then restart the service of nfs and enable it from boot

(NOT WORKING??)

```
[root@localhost servernfs]# service nfs start
nfs: unrecognized service
```

Practical 8
Apache Server

Steps :

1. Open the terminal and go to the root user

Code :
su - root

```
[rhel6@localhost ~]$ su - root
Password:
```

2. Use the ifconfig command to find the ip address of your machine and remember it.

Code :
ifconfig

```
[root@localhost ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:2F:C9:1B
          inet addr:192.168.1.3  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe2f:c91b/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:17 errors:0 dropped:0 overruns:0 frame:0
          TX packets:202 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1284 (1.2 KiB)  TX bytes:20958 (20.4 KiB)
          Interrupt:19 Base address:0x2024

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:328 errors:0 dropped:0 overruns:0 frame:0
          TX packets:328 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:30548 (29.8 KiB)  TX bytes:30548 (29.8 KiB)
```

3. Use the yum command to check if yum is working fine before we try to install httpd

Code :
yum list all

```
[root@localhost ~]# yum list all
```

```
-RedHatEnterpriseLinux-201105101829.i386/6.1
xz-libs.i686                        4.999.9-0.3.beta.20091007git.el6    @anaconda
-RedHatEnterpriseLinux-201105101829.i386/6.1
xz-lzma-compat.i686                 4.999.9-0.3.beta.20091007git.el6    @anaconda
-RedHatEnterpriseLinux-201105101829.i386/6.1
yelp.i686                           2.28.1-8.el6                        @anaconda
-RedHatEnterpriseLinux-201105101829.i386/6.1
yum.noarch                          3.2.29-17.el6                       @anaconda
-RedHatEnterpriseLinux-201105101829.i386/6.1
yum-metadata-parser.i686            1.1.2-16.el6                        @anaconda
-RedHatEnterpriseLinux-201105101829.i386/6.1
yum-rhn-plugin.noarch               0.9.1-26.el6                        @anaconda
-RedHatEnterpriseLinux-201105101829.i386/6.1
yum-utils.noarch                    1.1.30-6.el6                        @anaconda
-RedHatEnterpriseLinux-201105101829.i386/6.1
zd1211-firmware.noarch              1.4-4.el6                           @anaconda
-RedHatEnterpriseLinux-201105101829.i386/6.1
zenity.i686                         2.28.0-1.el6                        @anaconda
-RedHatEnterpriseLinux-201105101829.i386/6.1
zip.i686                            3.0-1.el6                           @anaconda
-RedHatEnterpriseLinux-201105101829.i386/6.1
zlib.i686                           1.2.3-25.el6                        @anaconda
-RedHatEnterpriseLinux-201105101829.i386/6.1
```

4. Use the yum command to install httpd

Code :

yum install httpd*

```
[root@localhost html]# yum install httpd*
Loaded plugins: product-id, refresh-packagekit, subscription-manager
Updating Red Hat repositories.
Setting up Install Process
Nothing to do
```

5. Then open the httpd configuration file in vim editor

Code :

vi /etc/httpd/conf/httpd/conf

```
[root@localhost ~]# vi /etc/httpd/conf/httpd.conf
```

Go to the end of the file, uncomment the following lines and make the changes. Then save the file.

<VirtualHost *:80>

ServerAdmin localhost

DocumentRoot /var/www/html

DirectoryIndex linuxman.html

</VirtualHost>

```
<VirtualHost *:80>
    ServerAdmin localhost
    DocumentRoot /var/www/html
    DirectoryIndex linuxman.html
```

6. Then go to the /var/www/html directory using the cd command and check if there is any error in the configuration file using httpd -t

Code :
cd /var/www/html
httpd -t

```
[root@localhost ~]# cd /var/www/html
[root@localhost html]# httpd -t
httpd: Could not reliably determine the server's fully qualified domain name, us
ing localhost.localdomain for ServerName
Syntax OK
```

7. Create a new file in the /var/www/html folder and enter some text. Then save the file.

Code :
vi linuxman.html

```
[root@localhost html]# vi linuxman.html
```

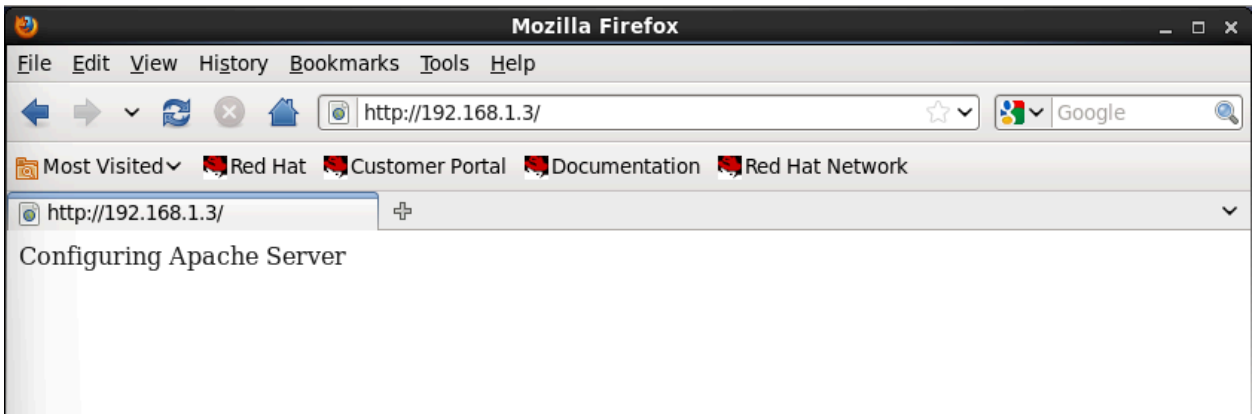
```
Configuring Apache Server
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --
```

8. Then restart the httpd service using the service restart command

Code :
service httpd restart

```
[root@localhost html]# service httpd restart
Stopping httpd:                                            [  OK  ]
Starting httpd: httpd: Could not reliably determine the server's fully qualified
 domain name, using localhost.localdomain for ServerName
                                                           [  OK  ]
```

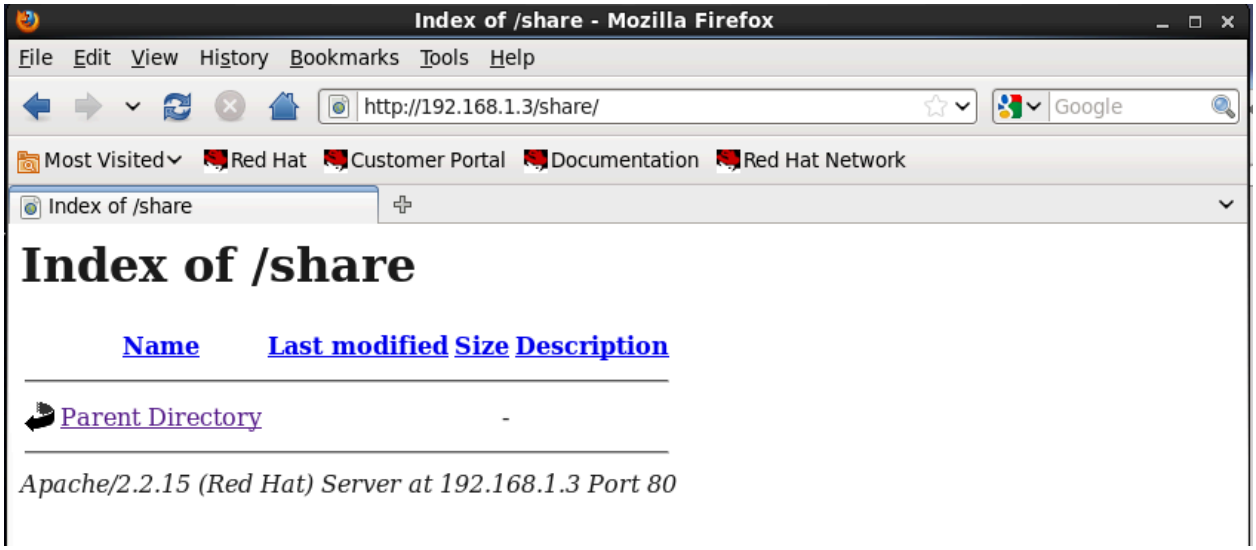
9.




```
[root@localhost html]# mkdir share
[root@localhost html]# vi /etc/httpd/conf/httpd.conf
<Directory "/var/www/html/share">
Options Indexes
Order ALlow,Deny
Allow from all
</Directory>

[root@localhost html]# service httpd restart
Stopping httpd:                                            [  OK  ]
Starting httpd: httpd: Could not reliably determine the server's fully qualified domain
name, using localhost.localdomain for ServerName
                                                           [  OK  ]
```

```
[root@localhost html]# touch 1.csv 2.html df.flv lm lk.xml
[root@localhost html]#
```

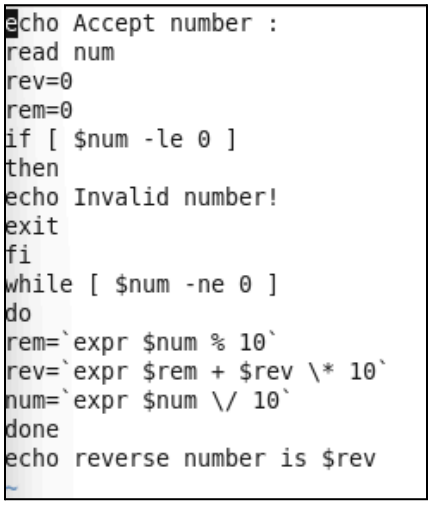

Index of /share - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://192.168.1.3/share/
Google
Most Visited
Red Hat
Customer Portal
Documentation
Red Hat Network
Index of /share
Index of /share
Name
Last modified
Size
Description
Parent Directory
-
Apache/2.2.15 (Red Hat) Server at 192.168.1.3 Port 80

Practical 9
Programs

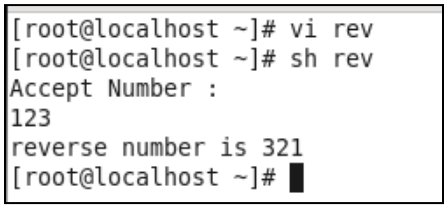
1. Reverse of a number

Code :

```
echo Accept number :
read num
rev=0
rem=0
if [ $num -le 0 ]
then
echo Invalid Number!
exit
fi
while [ $num -ne 0 ]
do
rem=`expr $num % 10`
rev=`expr $rem + $rev \* 10`
num=`expr $num \/ 10`
done
echo reverse number is $rev
```



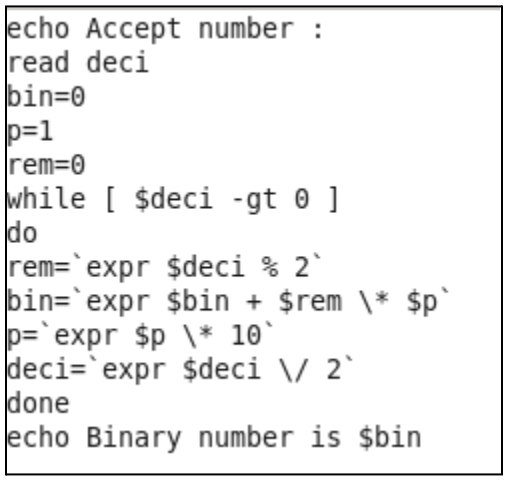
Output :



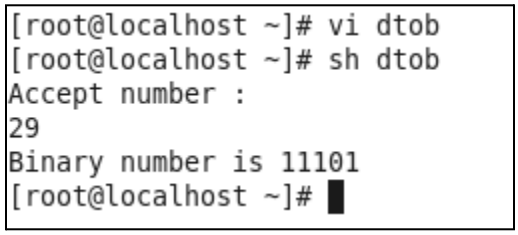
2. Decimal to binary conversion

Code :

```
echo Accept number :
read deci
bin=0
p=1
rem=0
while [ $deci -gt 0 ]
do
rem=`expr $deci % 2`
bin=`expr $bin + $rem \* $p`
p=`expr $p \* 10`
deci=`expr $deci \/ 2`
done
echo Binary number is $bin
```



Output :



3.Code : Binary to decimal

```
echo Accept number :
read bin
deci=0
p=1
rem=0
while [ $bin -gt 0 ]
do
rem=`expr $bin % 10`
deci=`expr $deci + $rem \* $p`
p=`expr $p \* 2`
bin=`expr $bin \/ 10`
done
echo Decimal number is $deci
```

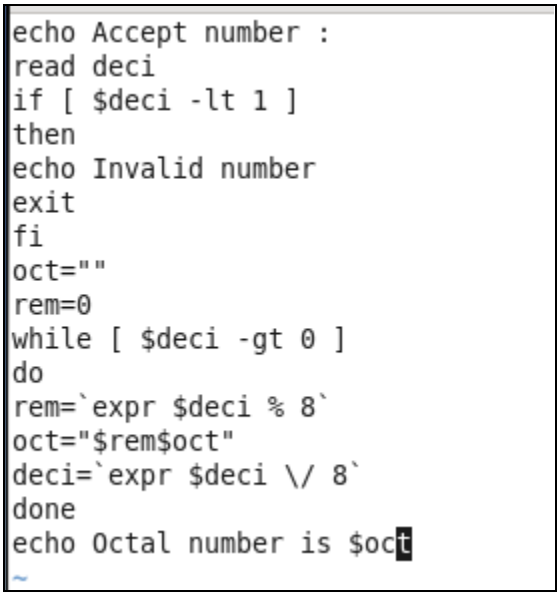
Output :

```
[root@localhost ~]# vi btod
[root@localhost ~]# sh btod
Accept number :
11101
Decimal number is 29
```

4. Decimal to Octal

Code :

```
echo Accept number :
read deci
if [ $deci -lt 1 ]
then
echo invalid number
exit
fi
oct=""
rem=0
while [ $deci -gt 0 ]
do
rem=`expr $deci % 8`
oct="$rem$oct"
deci=`expr $deci \/ 8`
done
echo Octal number is $oct
```



Output :

```
[root@localhost ~]# vi dtoo
[root@localhost ~]# sh dtoo
Accept number :
29
Octal number is 35
```

5. Convert lowercase to uppercase

Code :

```
if [ $* -le 2 ]
then
echo insufficient arguments
fi

if [ ! -f $1 ]
then
echo File name does not exist
fi

echo Converting lowercase to uppercase
cat $1 | tr '[a-z]' '[A-Z]'
```

```
if [ $* -le 2 ]
then
echo insufficient arguments
fi

if [ ! -f $1 ]
then
echo File name does not exist
fi

echo Cnverting lowercase to uppercase
cat $1 | tr '[a-z]' '[A-Z]'
```

Output :

```
[root@localhost ~]# vi lowtoupp
[root@localhost ~]# sh lowtoupp btod
lowtoupp: line 1: [: btod: integer expression expected
Cnverting lowercase to uppercase
ECHO ACCEPT NUMBER :
READ BIN
DECI=0
P=1
REM=0
WHILE [ $BIN -GT 0 ]
DO
REM=`EXPR $BIN % 10`
DECI=`EXPR $DECI + $REM \* $P`
P=`EXPR $P \* 2`
BIN=`EXPR $BIN \/ 10`
DONE
ECHO DECIMAL NUMBER IS $DECI

[root@localhost ~]#
```

5. Convert uppercase to lowercase

Code :

```
if [ $* -le 2 ]
then
echo insufficient arguments
fi

if [ ! -f $1 ]
then
echo File name does not exist
fi

echo Converting lowercase to uppercase
cat $1 | tr '[A-Z]' '[a-z]'
```

```
if [ $* -le 2 ]
then
echo insufficient arguments
fi

if [ ! -f $1 ]
then
echo File name does not exist
fi

echo Cnverting lowercase to uppercase
cat $1 | tr '[A-Z]' '[a-z]'
```

File1 :

```
HELLO WORLD
THIS IS FILE 1 IN UPPER CASE
```

Output :

```
[root@localhost ~]# vi upptolow
[root@localhost ~]# sh upptolow file1
upptolow: line 1: [: file1: integer expression expected
Cnverting lowercase to uppercase
hello world
this is file 1 in upper case


[root@localhost ~]#
```