T.Y. B.Sc. (IT) : Sem. V

Linux Administration

Time : 2½ Hrs.] Prelim Question Paper Solution [Marks : 75

Q.1Attempt the following (any THREE) [15]

Q.1(a) List and explain duties of System Administrator. [5]

Ans.: By definition, the Linux system administrator can be anyone who has "root" access – anyone who has root access is the system's "superuser."

Installing and Configuring Servers

When you hear the word server to describe a computer, you probably think of a computer that offers some type of service to clients. The server may provide file or printer sharing, File Transfer Protocol (FTP) or Web access, or email-processing tasks. Don't think of a server as a standalone workstation; think of it as a computer that specifically performs these services for many users.

In the Linux world, the word server has a broader meaning that what you might be used to. For instance, the standard Red Hat graphical user interface (GUI) requires a graphical layer called XFree86. This is a server. It must be configured.

Likewise, printing in Linux takes place only after you configure a print server. Again, this has become so easy as to be nearly trivial.

You need to know exactly which servers you need and how to employ them, and to be aware that it is bad practice and a potential security nightmare to enable services that the system isn't using and doesn't need.

Installing and Configuring Application Software

Although it is possible for individual users to install some applications in their home directories drive space set aside for their own files and customizations these applications may not be available to other users without the intervention of the user who installed the program or the system administrator. Besides, if an application is to be used by more than one user, it probably needs to be installed higher up in the Linux file hierarchy, which is a job that only the system administrator can perform.

Creating and Maintaining User Accounts

Not just anyone can show up and log on to a Linux machine. An account must be created for each user and you guessed it no one but the system administrator can do this. That's simple enough.

But there's more. It involves decisions that either you or your company must make. You might want to let users select their own passwords, which would no doubt make them easier to remember but which probably would be easier for a malefactor to crack. You might want to assign passwords, which is more secure in theory but increases the likelihood that users will write them down on a conveniently located scrap of paper a risk if many people have access to the area where the machine(s) is located. You might decide that users must change their passwords periodically something you can configure Red Hat Enterprise Linux to prompt users about.

What happens to old accounts? Suppose that someone leaves the company. You probably don't want that person to gain access to the company's network, but you also don't want to delete the account wholesale, only to discover later that essential data resided nowhere else.

Backing Up and Restoring Files

Until computer equipment becomes infallible, until people lose the desire to harm others' property, and truth be told until system administrators become perfect, there is considerable need to back up important files so that the system can be up and running again with minimal disruption in the event of hardware, security, or administration failure. Only the system administrator may do this.

Monitoring and Tuning Performance

On a modern standalone system, Linux runs pretty quickly. If it doesn't, there's something wrong something the system administrator can fix. Still, you might want to squeeze one last little bit of performance out of your hardware or a number of people might be using the same file server, mail server, or other shared machine, in which case seemingly small improvements in system performance add up.

System tuning is an ongoing process aided by a variety of diagnostic and monitoring tools. Some performance decisions are made at installation time, while others are added or tweaked later.

Proper monitoring allows you to detect a misbehaving application that consumes more resources than it should or fails to exit completely upon closing.

Possibly most important, careful system monitoring and diagnostic practices give you a heads-up when a system component is showing early signs of failure, so that you can minimize any potential downtime.

In any case, careful system monitoring plus wise use of the built-in configurability of Linux allows you to squeeze the best possible performance from your existing equipment, from customizing video drivers to applying special kernel patches or simply turning off unneeded services to free memory and processor cycles.

Configuring a Secure System

The system administrator's task, first and foremost, is to make certain that no data on the machine or network is likely to become corrupted, whether by hardware or power failure, misconfiguration or user error (to the extent that the latter can be avoided), or malicious or inadvertent intrusion from elsewhere. This means doing all the tasks described throughout this chapter, and doing them well, with a full understanding of their implications.

Linux machines themselves were invulnerable, but the huge amount of traffic generated by this worm infection nevertheless prevented many Linux machines from accomplishing much Web-based work for several weeks, so fierce was the storm raging across the Internet. Security is an ongoing process.

Using Tools to Monitor Security

People who, for purposes of larceny or to amuse themselves, like to break into computers they're called crackers are a clever bunch. If there is a vulnerability in a system, they will find it. Fortunately, the Linux development community is quick to find potential exploits and to create ways of slamming the door shut before crackers can enter. Your first and best security tool, therefore, is making sure that whenever a security advisory is issued, you download and install the repaired package. This line of defense can be annoying but it is nothing compared to rebuilding a compromised system.
Red Hat equips you with tools to detect and deal with unauthorized access of many kinds.

Q.1(b) Discuss Linux Distribution. [5]

Ans.: Linux Distributions

Although there is only one standard version of Linux, there are actually several different distributions. Different companies and groups have packaged Linux and Linux software in slightly different ways.

Red Hat Linux

Red Hat Linux is currently the most popular Linux distribution. As a company, Red Hat provides software and services to implement and support professional and commercial Linux systems. Red Hat freely distributes its version of Linux under the GNU Public License. Red Hat generates income by providing professional level support, consulting, and training services. Red Hat originated the RPM package system used on several distributions, which automatically installs and removes software packages.

Red Hat maintains an extensive library of Linux documentation that is freely accessible online. On its Web site, you can link to its support page, which lists the complete set of Red Hat manuals, all in Web page format for easy viewing with any Web browser.

Red Hat offers several commercial products and services for business and e-commerce solutions.

Mandrake

Mandrake Linux is another popular Linux distribution with many of the same features as Red Hat. It focuses on providing up-to-date enhancements and an easy-to-use installation and GUI configuration.

SuSE

Originally a German language-based distribution, SuSE has become very popular throughout Europe and is currently one of the fastest growing distributions worldwide.

Debian

Debian Linux is an entirely noncommercial project, maintained by hundreds of volunteer programmers.

Its aim is to enhance Linux with new and improved applications and implementations.

Caldera

Caldera OpenLinux is designed for corporate commercial use. Caldera has organized its OpenLinux distribution into several different packages, each geared to different markets. These include the Desktop package, which is designed for basic workstation operations, and the eServer package, which is designed for Linux servers.

Caldera also offers a line of commercial and proprietary Linux packages.

Slackware

Slackware is available from numerous Internet sites, and you can order the CD from Walnut Creek Software. The Slackware distribution takes special care to remain as closely Unix compliant as possible.

TurboLinux

TurboLinux provides English, Chinese, and Japanese versions of Linux. It includes several of its own packages, such as TurboPkg, for automatically updating applications.

Q.1(c) Discuss the different ways to view the content of Text file. [5]

Ans.: Viewing the Contents of Text Files

When administering your RHEL server, you will very often find that you are modifying configuration files, which are all ASCII text files. Therefore, the ability to browse the content of these files is very important. Different methods exist to perform this task.

- **cat** This command displays the contents of a file by dumping it to the screen, This can be useful if the contents of the file do not fit on the screen. You will see some text scrolling by, and as the final result, you will see only the last lines of the file being displayed on the screen.
- **tac** This command does the same thing as cat but inverts the result; that is, not only is the name of tac the opposite of cat, but the result is the opposite as well. This command will dump the contents of a file to the screen, but with the last line first and the first line last.
- **tail** This command shows only the last lines of a text file. If no options are used, this command will show the last 10 lines of a text file. The command can also be modified to show any number of lines on the bottom of a file. For example, tail -n 2 /etc/passwd will show you the last two lines of the configuration file where usernames are stored. The option to keep tail open on a given log file is also very useful for monitoring what happens on your system. For example, if you use tail -f /var/log/messages, the most generic log file on your system is opened, and when a new line is written to the bottom of that file, you will see it immediately.
- **head** This command is the opposite of tail. It displays the first lines of a text file.
- **less** The last command used to monitor the contents oftext files is less. This command will open a plain text file viewer, In the viewer, you can browse the file using the Page Down key, Page Up key, or spacebar. It also offers a search capability. From within the less viewer, use /sotnetext to find scwetext in the file. To quit less, use q.
- **more** This command is similar to less but not as advanced.

Q.1(d) What are Links? Explain different types of Links. [5]

Ans.: In a Linux file system, it Is very useful to be able to access a single file from different locations. This discourages you from copying a file to different locations, where subsequently different versions of the file may come to exist. In a Linux file system, you can use links for this purpose. A link appears to be a regular file, but it's more like a pointer that exists in one location to show you how to get to another location.

In Linux, there are two different types of links. A symbolic link is the most flexible link type you can use. It points to any other file and any other directory, no matter where it is.

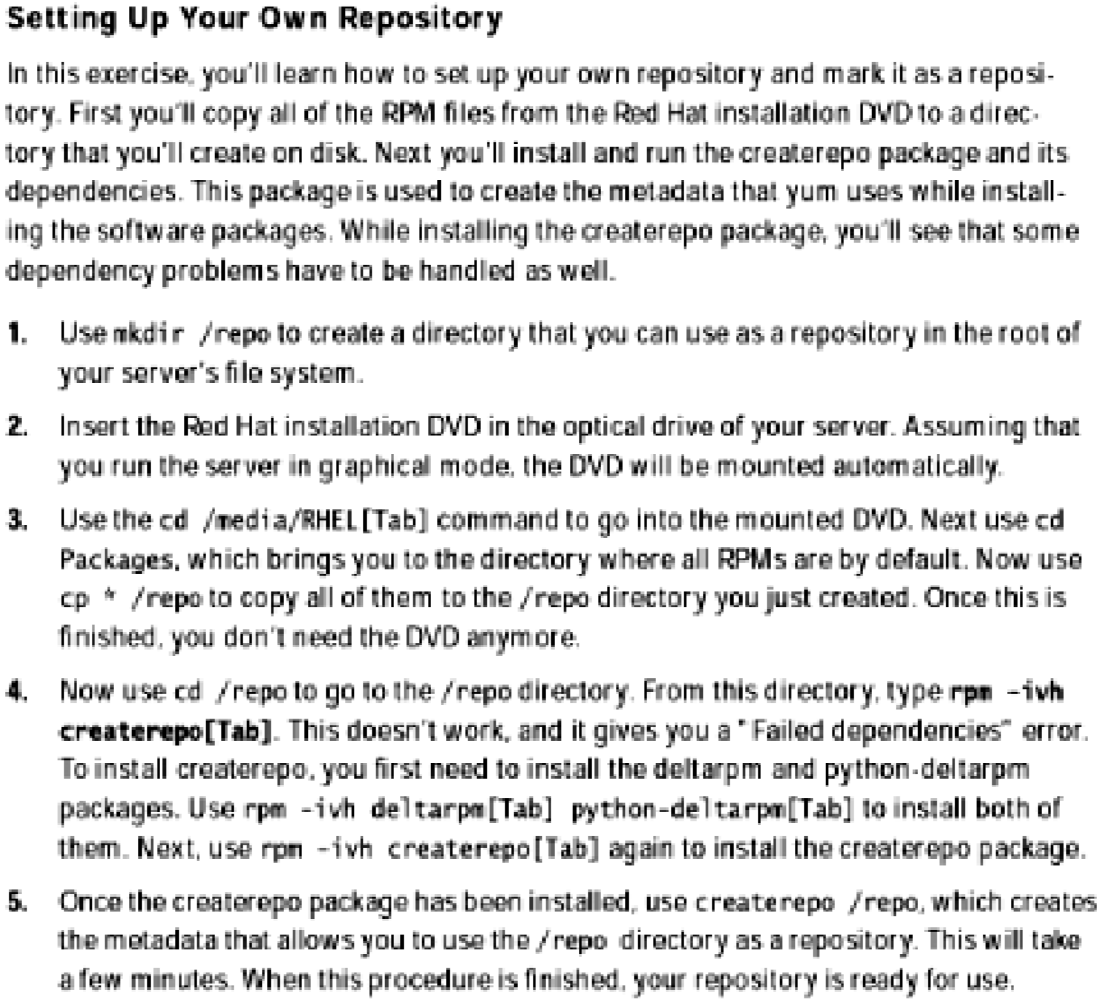
A hard link can be used only to point to a file that exists on the same device.

With symbolic links, there is a difference between the original file and the link. If you remove the original file, the symbolic link won't work anymore and thus is invalid.

A hard link is more like an additional name you'd give to a file. To understand hard links, you have to appreciate how Linux file systems work with inodes. The inode is the administration of a file. To get to a file, the file system reads the file's inode in the file system metadata, and from there it learns how to access the block where the actual data of the file is stored. To get to the inode, the file system uses the filename that exists somewhere in a directory. A hard link is an additional filename that you can create anywhere in a directory on the same device that gives access to the same file system metadata. With hard links, you only need the original filename to create the hard link. Once it has been created, it isn't needed anymore, and the original filename can be removed. In general, you'll use symbolic links, not hard links, because hard links have some serious limitations.

Q.1(e) How to create and manage Own Repositories. [5]

Ans.:



Setting Up Your Own Repository

In this exercise, you'll learn how to set up your own repository and mark it as a repository. First you'll copy all of the RPM files from the Red Hat installation DVD to a directory that you'll create on disk. Next you'll install and run the createrepo package and its dependencies. This package is used to create the metadata that yum uses while installing the software packages. While installing the createrepo package, you'll see that some dependency problems have to be handled as well.

1. Use `mkdir /repo` to create a directory that you can use as a repository in the root of your server's file system.
2. Insert the Red Hat installation DVD in the optical drive of your server. Assuming that you run the server in graphical mode, the DVD will be mounted automatically.
3. Use the `cd /media/RHEL[Tab]` command to go into the mounted DVD. Next use `cd Packages`, which brings you to the directory where all RPMs are by default. Now use `cp * /repo` to copy all of them to the /repo directory you just created. Once this is finished, you don't need the DVD anymore.
4. Now use `cd /repo` to go to the /repo directory. From this directory, type **`rpm -ivh createrepo[Tab]`**. This doesn't work, and it gives you a "Failed dependencies" error. To install createrepo, you first need to install the deltarpm and python-deltarpm packages. Use `rpm -ivh deltarpm[Tab] python-deltarpm[Tab]` to install both of them. Next, use `rpm -ivh createrepo[Tab]` again to install the createrepo package.
5. Once the createrepo package has been installed, use `createrepo /repo`, which creates the metadata that allows you to use the /repo directory as a repository. This will take a few minutes. When this procedure is finished, your repository is ready for use.

Q.1(f) Write a short note on Nice and Renice Command, show how to set and change priority using them. [5]

Ans.: When using the nice command, you can adjust the process niceness from -20, which is good for the most favorable scheduling, to 19 for the least favorable scheduling. By default, all processes are started with a niceness of 0. The following sample code line shows how to start the dd command with an adjusted niceness of -10, which makes It more favorable and therefore allows it to finish its work faster:

```
nice -n -10 dd if=/dev/sda of=/dev/sdb
```

Aside from specifying which niceness setting to use when starting a process, you can also use the renice command to adjust the niceness of a command that has already started. By default, renice works on the PID of the process whose priority you want to adjust.

Thus, you have to find this PID before using renice. The ps command described earlier in this chapter is used to do this.

If, for example, you want to adjust the niceness of the f i nd command that you just started, you would begin by using ps aux | grep fi nd, which gives you the PID of the command. Assuming that would give you the PID I 234, you can use renice -10 1234 to adjust the niceness of the command.

Another method of adjusting process niceness is to do it from top. The convenience of using top for this purpose is that top shows only the busiest processes on your server, which are typically the processes whose niceness you want to adjust anyway. After identifying the

PID of the process you want to adjust, from the top interface press r. You'll now see the PID to renice message on the sixth line of the top window, N ow enter the PID of the process you want to adjust. The top program then prompts you with Renice PID 3284 to value. Here you enter the positive or negative nice value you want to use. Finally, press Enter to apply the niceness to the selected process.

Q.2Attempt the following (any THREE) [15]

Q.2(a) What is Partition and how to create Partition using fdisk command. [5]

Ans.: Creating partition in linux :

1) insert a user
2) execute following command that will gives information about the pen drive.
   #dmesg al (Disk message)
3) Due to command of step 2 user can understand where the pen drive is mounted :
   e.g. : /dev/sdb (storage device b)
4) Use fdisk command to start creating the partion :
   #_fdisk=cu/der/sdb
   (create utility)
5) Now a menu will appear.
6) For primary :
   i) Type n (new)
   ii) Type p (primary)
   iii) Type 1 (partition no)
   iv) Press enter twice to select first and last sector of primary.
   v) Primary is created.
7) For extended :
   i) Type n (new)
   ii) Type e (extended)
   iii) Type 2 (partition no.)
   iv) Press enter twice to select first and last sector.
   v) Extended is created.
8) For logical :
   i) Type n (new)
   ii) Type l (logical)
   iii) Press enter twice to select first and last sector of logical partition.
   iv) Logical is created.
9) Type 'w' (write) to save and exit.
10) No error indicates partition is created.

Q.2(b) Write a short note on run level in Linux. [5]

Ans.:

Level No.	Level Name	Description
0	Halt	It is used for shutting down the system.
1	Single user	Used by an administrator for maintenance.
2	Multi user without nfs	No feature of mount and unmounts of the file system.
3	Multi user without GUI	Used by professional user mostly on server side.
4	Not used	
5	Multi user with GUI	Default run level mostly used on client side.
6	Reboot	To restart the system.

Q.2(c) How to configure key based SSH configuration, also show setting up two way authentication. [5]

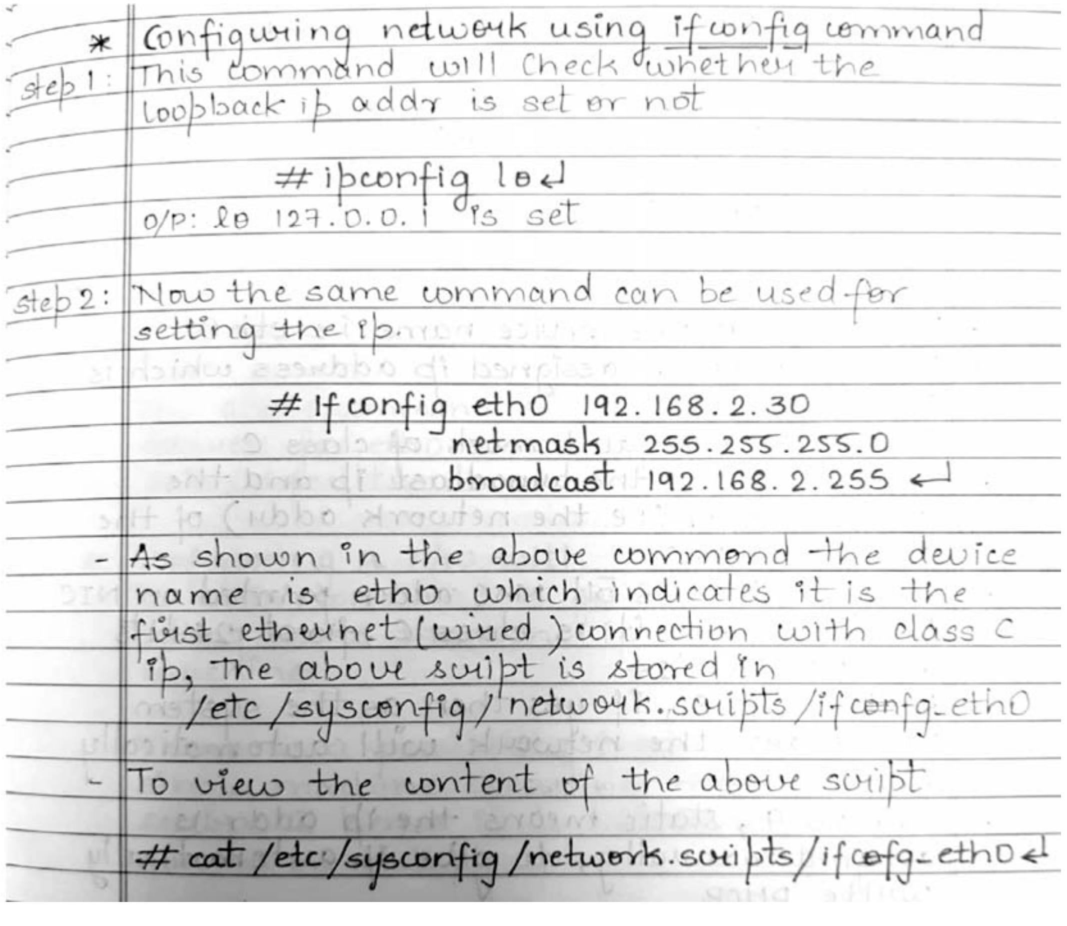
Ans.: Setting Up Key-Based SSH Authentication Protected with a Passphrase

In this exercise, you'll generate an SSH public/private key pair. You'll protect the privatekey by adding a passphrase. Next you will start ssh-agent to cache the passphrase.

1. Open a root shell, and type ssh-keygen. When asked where to save the file, press Enter. You will be prompted that the file /root/.ssh/id_rsa already exists. Type Y to confirm that you want to overwrite this file.
2. Now enter a passphrase, and press Enter to confirm. Type the same passphrase again, and once more press Enter. The key will now be saved.
3. Copy the new public key to your server using ssh-copy-id server. You need to enter your password once to perform this operation.
4. Establish an SSH session to your server using ssh server. Enter the passphrase when prompted. Next type exit to close this session.
5. Type ssh-agent /bin/bash Next type ssh-add to add your current passphrase. Enter the passphrase, and you will sec a confirmation prompting Identity added.
6. Type ssh server. At this point, you'll notice that you can enter a session without entering a passphrase.
7. Type exit to close the current ssh session.
8. Type exit to close the ssh-agent session. When you do this, the passphrase is forgotten.
9. Type ssh server to establish a new session. Notice that you are prompted to enter the passphrase again.

Q.2(d) How to set up network using command? [5]

Ans.:



* Configuring network using ifconfig command

step 1: This command will Check whether the loopback ip addr is set or not

```
# ipconfig lo ↵
```

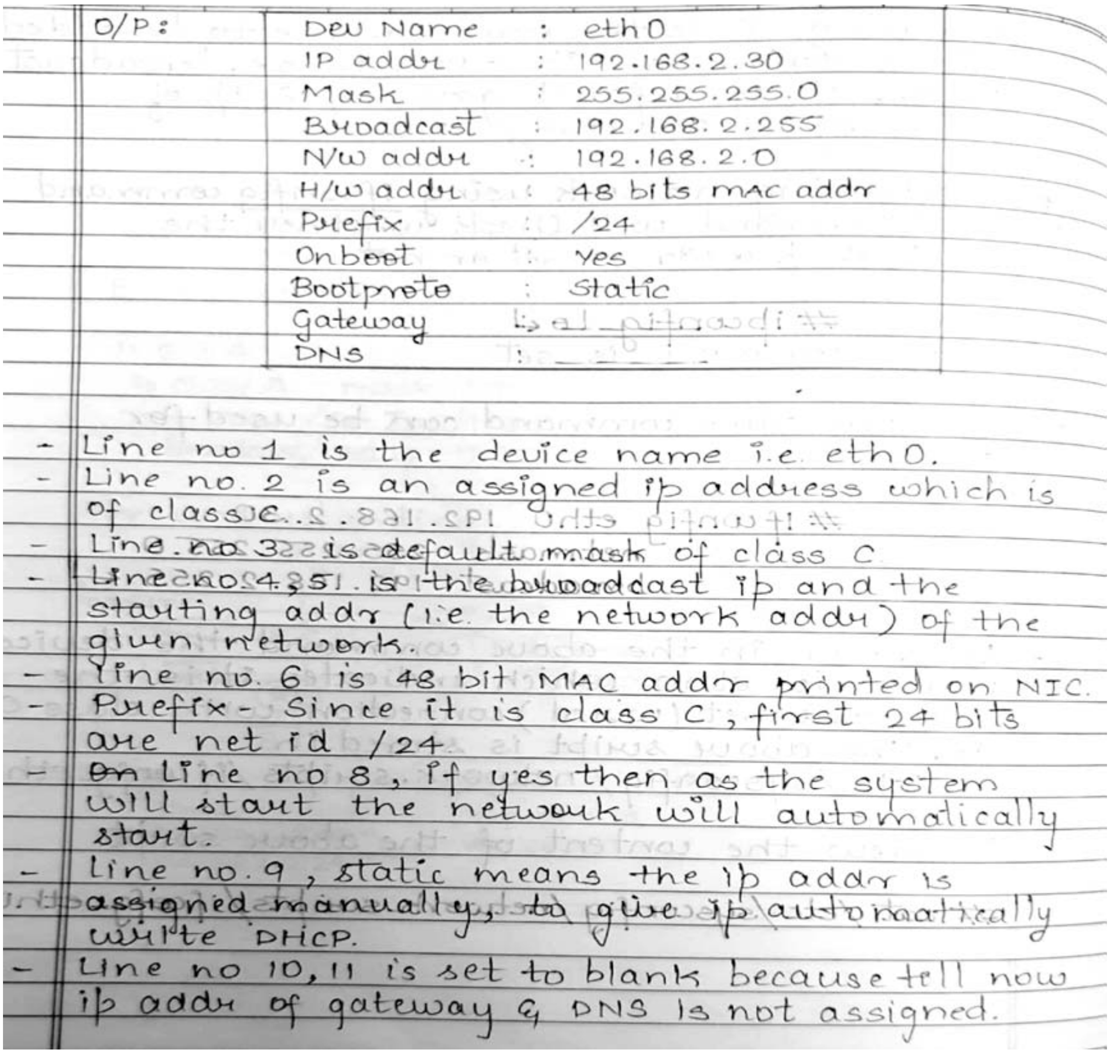
o/p: lo 127.0.0.1 is set

step 2: Now the same command can be used for setting the ip.

```
# Ifconfig eth0 192.168.2.30
                netmask 255.255.255.0
                broadcast 192.168.2.255 ↵
```

- As shown in the above commond the device name is eth0 which indicates it is the first ethernet (wired) connection with class C ip, The above script is stored in /etc/sysconfig/network.scripts/ifconfg-eth0
- To view the content of the above script

```
# cat /etc/sysconfig/network.scripts/ifcofg-eth0 ↵
```



O/P:	Dev Name	:	eth0
	IP addr	:	192.168.2.30
	Mask	:	255.255.255.0
	Broadcast	:	192.168.2.255
	N/w addr	:	192.168.2.0
	H/w addr	:	48 bits MAC addr
	Prefix	:	/24
	Onboot	:	Yes
	Bootproto	:	Static
	Gateway		
	DNS		

- Line no 1 is the device name i.e. eth0.
- Line no. 2 is an assigned ip address which is of classC.
- Line no 3 is default mask of class C
- Line no 4 & 5 is the broadcast ip and the starting addr (i.e. the network addr) of the given network.
- line no. 6 is 48 bit MAC addr printed on NIC.
- Prefix - Since it is class C, first 24 bits are net id /24
- ~~On~~ Line no 8, if yes then as the system will start the network will automatically start.
- Line no. 9, static means the ip addr is assigned manually, to give ip automatically write DHCP.
- Line no 10, 11 is set to blank because tell now ip addr of gateway & DNS is not assigned.

Q.2(e) What are the different Advance Permissions? **[5]**

Ans.:

Understanding Advanced Permissions

There are three advanced permissions. The first of them is the *set user ID (SUID) permission.* On some very specific occasions, you may want to apply this permission to executable files. By default, a user who runs an executable file runs this file with their own permissions. For standard users, this normally means the use of the program is restricted. In some cases, however, the user needs special permissions for the execution of a certain task.

Consider the situation where a user needs to change their password. To do this, the user needs to write the new password to the /etc/shadow file. This file, however, is not writable for users who don't have root permissions:

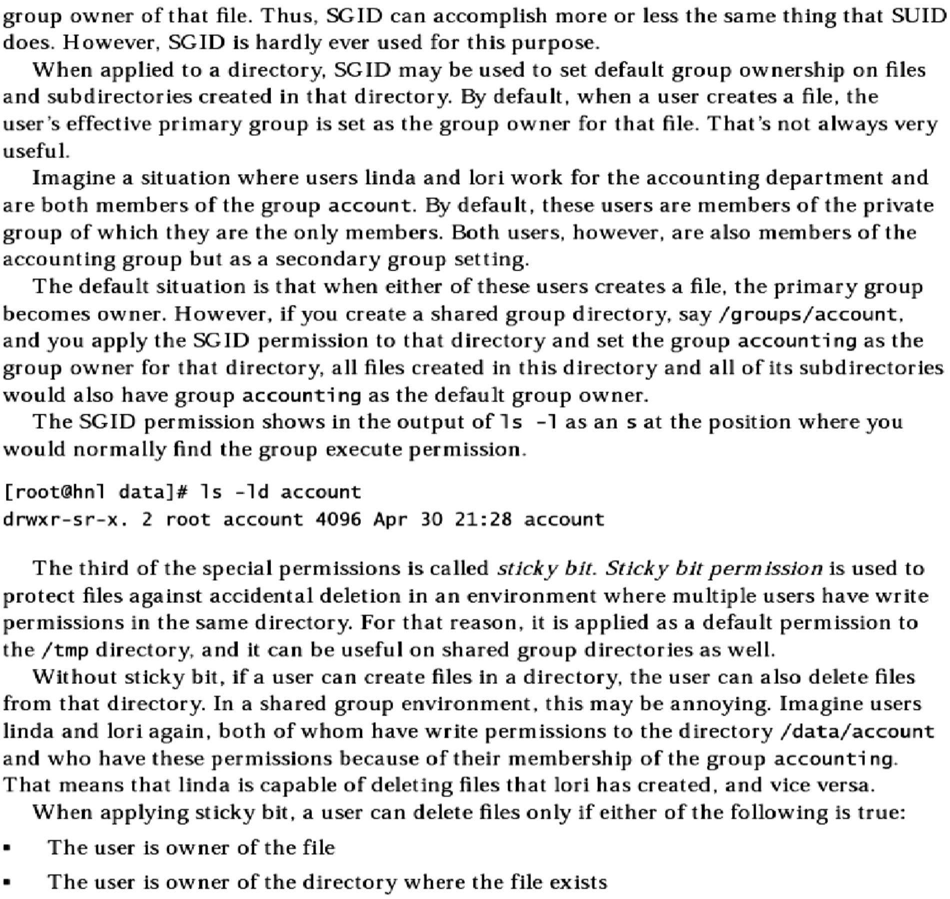
```
[root@hnl ~]# ls -l /etc/shadow
----------. 1 root root 1184 Apr 30 16:54 /etc/shadow
```

The SUID permission offers a solution for this problem. On the /usr/bin/passwd utility, this permission is applied by default. This means that when changing a password, the user temporarily has root permissions, which allow the user to write to the /etc/shadow file. You can see the SUID permission with `ls -l` as an s at the position where normally you would expect to see the x for the user permissions.

```
[root@hnl ~]# ls -l /usr/bin/passwd
-rwsr-xr-x. 1 root root 32680 Jan 28  2010 /usr/bin/passwd
```

The SUID permission may look useful, and it is in rare cases. At the same time, however, it is potentially dangerous. If applied incorrectly, you can give away root permissions by accident. I therefore recommend using it with the greatest care only.

The second special permission is *set group ID (SGID).* This permission has two effects. If applied on an executable file, it gives the user who executes the file the permissions of the



group owner of that file. Thus, SGID can accomplish more or less the same thing that SUID does. However, SGID is hardly ever used for this purpose.

When applied to a directory, SGID may be used to set default group ownership on files and subdirectories created in that directory. By default, when a user creates a file, the user's effective primary group is set as the group owner for that file. That's not always very useful.

Imagine a situation where users linda and lori work for the accounting department and are both members of the group `account`. By default, these users are members of the private group of which they are the only members. Both users, however, are also members of the accounting group but as a secondary group setting.

The default situation is that when either of these users creates a file, the primary group becomes owner. However, if you create a shared group directory, say `/groups/account`, and you apply the SGID permission to that directory and set the group `accounting` as the group owner for that directory, all files created in this directory and all of its subdirectories would also have group `accounting` as the default group owner.

The SGID permission shows in the output of `ls -l` as an `s` at the position where you would normally find the group execute permission.

```
[root@hnl data]# ls -ld account
drwxr-sr-x. 2 root account 4096 Apr 30 21:28 account
```

The third of the special permissions is called *sticky bit*. *Sticky bit permission* is used to protect files against accidental deletion in an environment where multiple users have write permissions in the same directory. For that reason, it is applied as a default permission to the `/tmp` directory, and it can be useful on shared group directories as well.

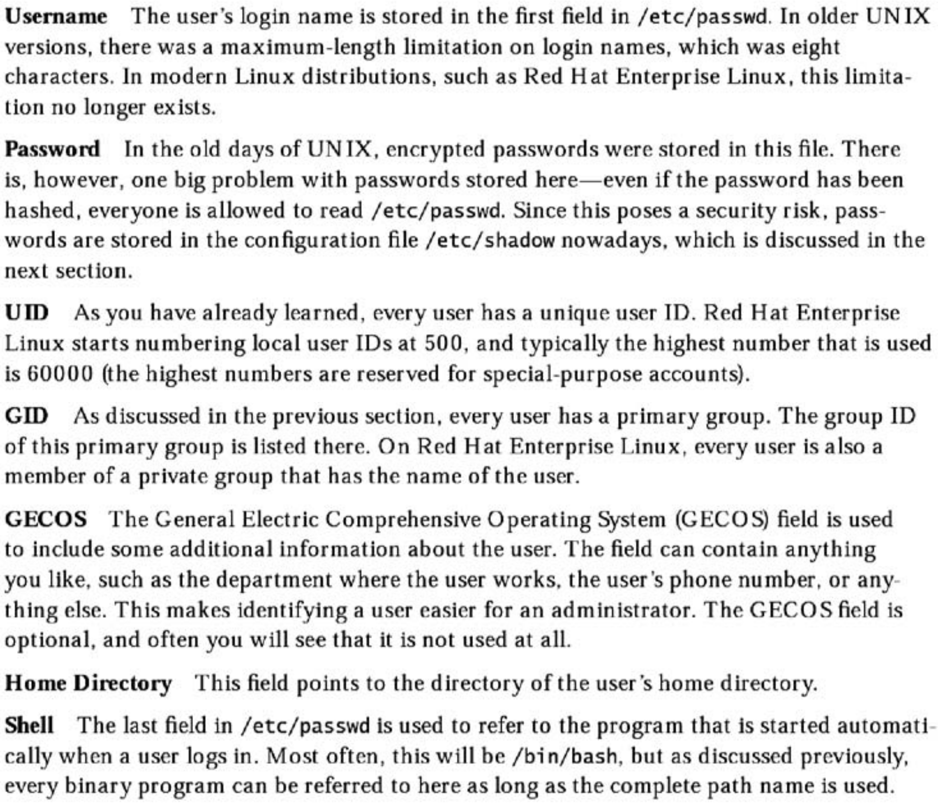
Without sticky bit, if a user can create files in a directory, the user can also delete files from that directory. In a shared group environment, this may be annoying. Imagine users linda and lori again, both of whom have write permissions to the directory `/data/account` and who have these permissions because of their membership of the group `accounting`. That means that linda is capable of deleting files that lori has created, and vice versa.

When applying sticky bit, a user can delete files only if either of the following is true:

- The user is owner of the file
- The user is owner of the directory where the file exists

Q.2(f) Write a short note on /etc/passwd. [5]

Ans.:



Username The user's login name is stored in the first field in `/etc/passwd`. In older UNIX versions, there was a maximum-length limitation on login names, which was eight characters. In modern Linux distributions, such as Red Hat Enterprise Linux, this limitation no longer exists.

Password In the old days of UNIX, encrypted passwords were stored in this file. There is, however, one big problem with passwords stored here—even if the password has been hashed, everyone is allowed to read `/etc/passwd`. Since this poses a security risk, passwords are stored in the configuration file `/etc/shadow` nowadays, which is discussed in the next section.

UID As you have already learned, every user has a unique user ID. Red Hat Enterprise Linux starts numbering local user IDs at 500, and typically the highest number that is used is 60000 (the highest numbers are reserved for special-purpose accounts).

GID As discussed in the previous section, every user has a primary group. The group ID of this primary group is listed there. On Red Hat Enterprise Linux, every user is also a member of a private group that has the name of the user.

GECOS The General Electric Comprehensive Operating System (GECOS) field is used to include some additional information about the user. The field can contain anything you like, such as the department where the user works, the user's phone number, or anything else. This makes identifying a user easier for an administrator. The GECOS field is optional, and often you will see that it is not used at all.

Home Directory This field points to the directory of the user's home directory.

Shell The last field in `/etc/passwd` is used to refer to the program that is started automatically when a user logs in. Most often, this will be `/bin/bash`, but as discussed previously, every binary program can be referred to here as long as the complete path name is used.

Q.3Attempt the following (any THREE) [15]

Q.3(a) What is Masquerading and how to configure NAT? [5]

Ans.:

In IP masquerading, you can configure a server to connect your local network to the Internet. In this configuration, IP addresses from the private address ranges are used on the private network. These addresses cannot communicate on the Internet, but they will be translated to the public IP address on the interface that faces the Internet. This process is known as IP masquerading, also referred to as Network Address Translation (NAT). The major benefit of using masquerading is that with just one public IP address, you can connect many devices on the private network to the Internet. IP masquerading is commonly used in home and corporate networks.

Configuring NAT

In this exercise, you'll learn how to set up NAT. You'll use the host computer as the NAT router and the virtual machine as the web server on the private network. All configuration is done on the host computer, but you must make sure that all iptables configuration is erased. (You'll take care of this during the first steps of this exercise.) To test the configuration, you'll need a third machine that is connected to the same network as the RHEL server that you are using as host. If you don't have a third machine available, that's fine; you just can't test the configuration.

1. On the virtual machine, use iptables -F, iptables -P INPUT ALLOW, iptables -P OUTPUT ALLOW, and iptables -P FORWARD ALLOW. Next use service iptables save to save this configuration.

2. Repeat step 1 on the host computer, and use service httpd stop followed by chkconfig httpd off to make sure the web server is stopped on the host computer.

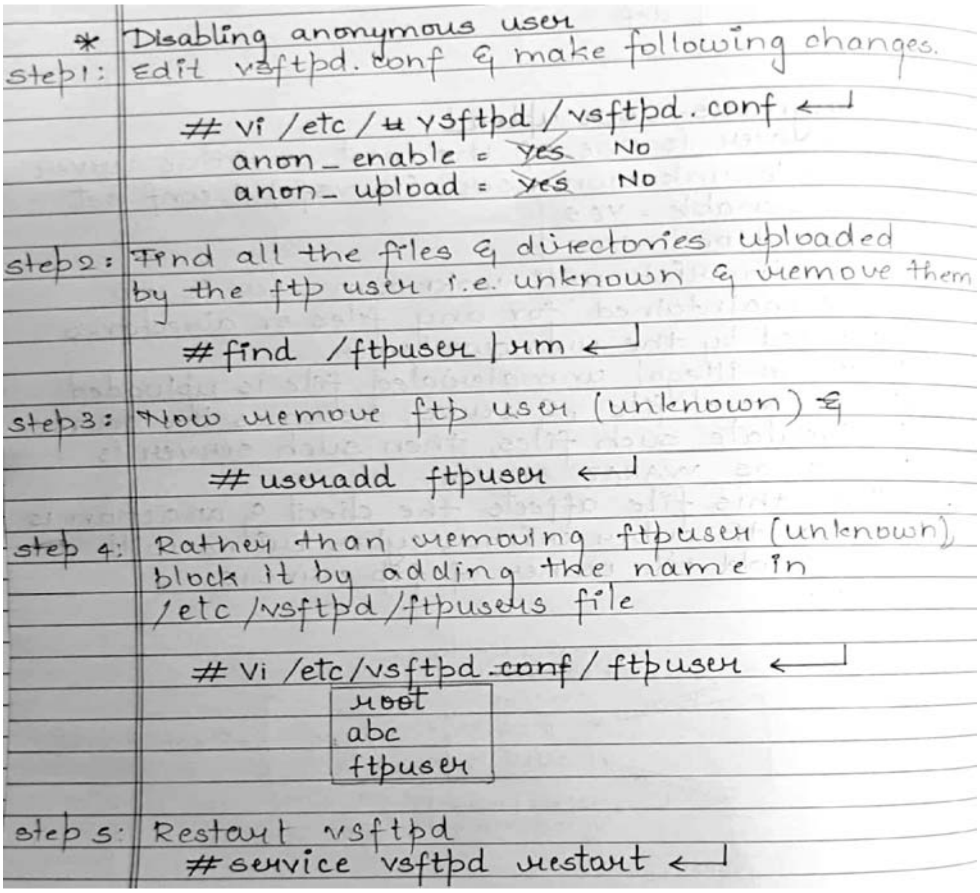
3. On the host computer, enable routing by opening /etc/sysctl.conf with an editor. Make sure it includes the line net.ipv.ip_forward = 1, and use sysctl -p to the sysctl service to make the change effective.

4. On the host, use iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j DNAT --to-destination 192.168.100.76. Make sure that you use the actual name of your network card on the host machine and that the --to-destination option uses the IP address of the virtual machine.

5. Also on the host, use iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE to enable IP masquerading.

Q.3(b) How to disable unknown user in FTP? **[5]**

Ans.:



* Disabling anonymous user

step1: Edit vsftpd.conf & make following changes.

```
# vi /etc/ ~~u~~ vsftpd/vsftpd.conf ←┘
anon_enable = ~~yes~~ No
anon_upload = ~~yes~~ No
```

step2: Find all the files & directories uploaded by the ftp user i.e. unknown & remove them

```
# find /ftpuser | rm ←┘
```

step3: Now remove ftp user (unknown) &

```
# useradd ftpuser ←┘
```

step 4: Rather than removing ftpuser (unknown), block it by adding the name in /etc/vsftpd/ftpusers file

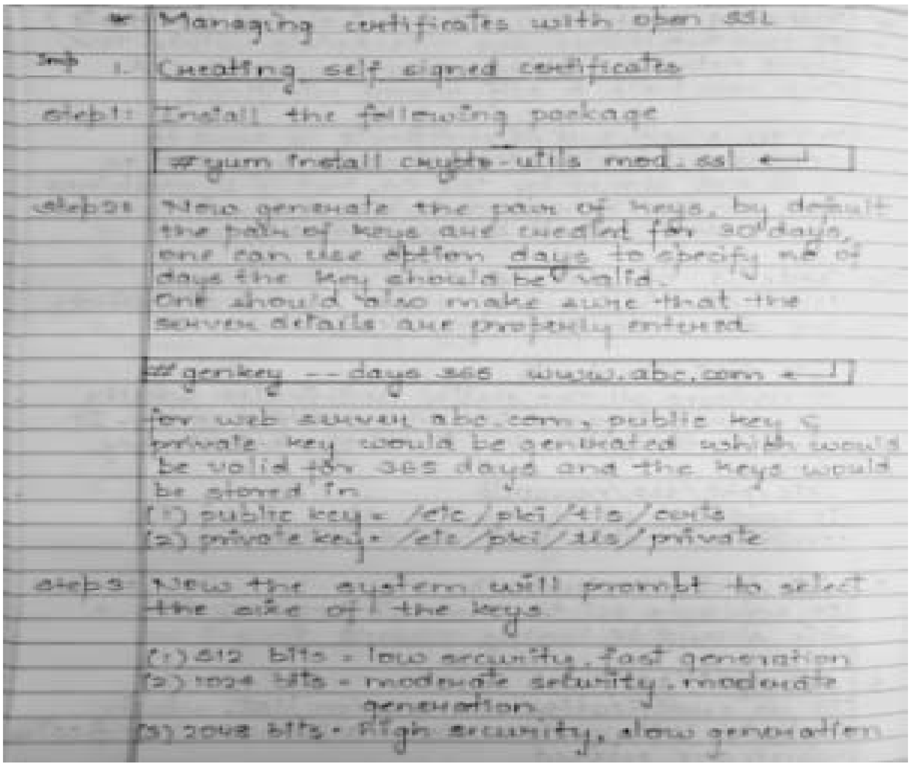
```
# vi /etc/vsftpd ~~.conf~~ / ftpuser ←┘
root
abc
ftpuser
```

step 5: Restart vsftpd

```
# service vsftpd restart ←┘
```

Q.3(c) How to create self-signed certificate? **[5]**

Ans.:



* Managing certificates with open ssl

Imp 1. Creating self signed certificate

step1: Install the following package

```
# yum install crypto-utils mod_ssl ←┘
```

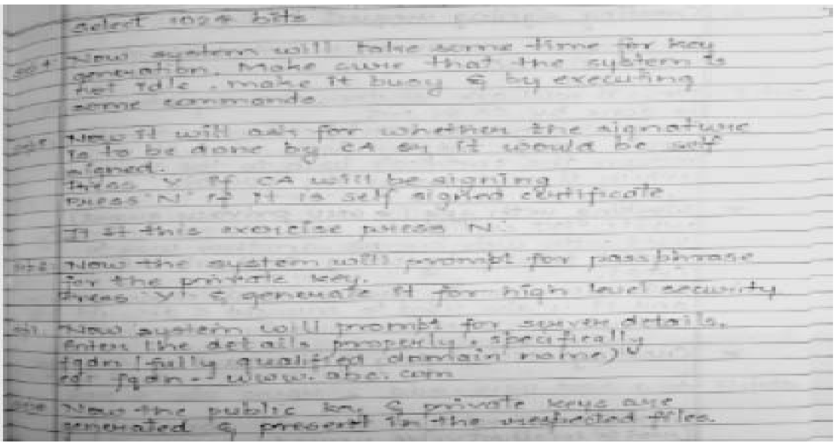
step2: Now generate the pair of keys, by default the pair of keys are created for 30 days, one can use option days to specify no of days the key should be valid.
One should also make sure that the server details are properly entered

```
# genkey --days 365 www.abc.com ←┘
```

for web server abc.com, public key & private key would be generated which would be valid for 365 days and the keys would be stored in
(1) public key = /etc/pki/tls/certs
(2) private key = /etc/pki/tls/private

step3: Now the system will prompt to select the size of the keys.

(1) 512 bits = low security, fast generation
(2) 1024 bits = moderate security, moderate generation
(3) 2048 bits = High security, slow generation



select 1024 bits

Now system will take some time for key generation. Make sure that the system is not idle, make it busy & by executing some commands

Now it will ask for whether the signature is to be done by CA or it would be self signed.
Press 'Y' if CA will be signing
Press 'N' if it is self signed certificate

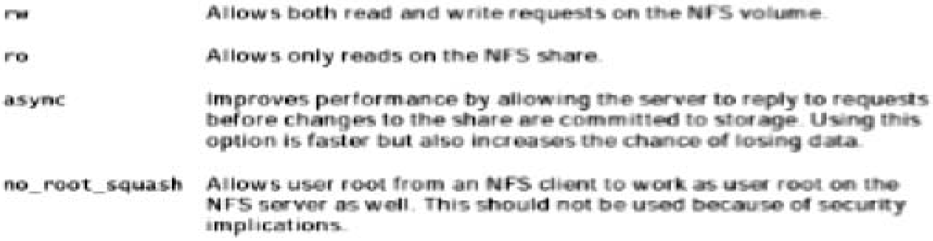
In this exercise press 'N'

Now the system will prompt for passphrase for the private key.
Press 'Y' & generate it for high level security

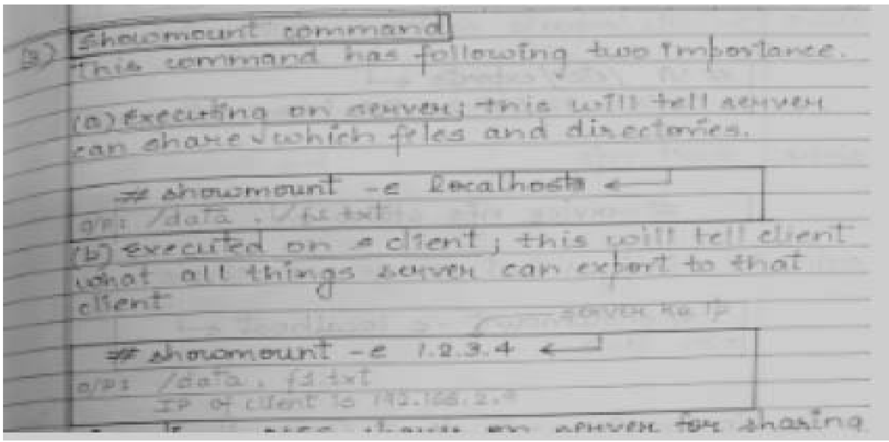
Now system will prompt for server details.
Enter the details properly, specifically fqdn (fully qualified domain name)
eg: fqdn - www.abc.com

Now the public key & private keys are generated & present in the respected files.

Q.3(d) What are different NFS server permissions and also explain showmount command. [5]
Ans.:



rw	Allows both read and write requests on the NFS volume.
ro	Allows only reads on the NFS share.
async	Improves performance by allowing the server to reply to requests before changes to the share are committed to storage. Using this option is faster but also increases the chance of losing data.
no_root_squash	Allows user root from an NFS client to work as user root on the NFS server as well. This should not be used because of security implications.



3) Showmount command
This command has following two importance.

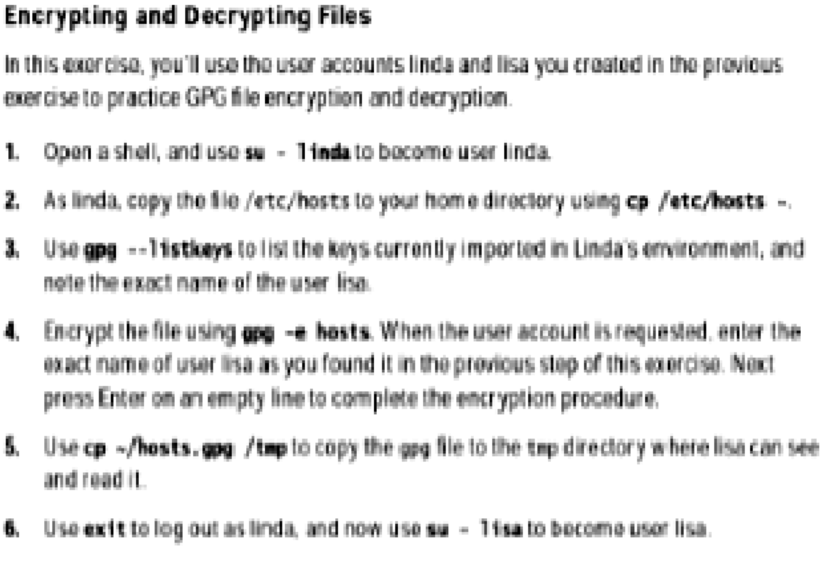
(a) Executing on server; this will tell server can share which files and directories.

```
# showmount -e localhost ←
o/p: /data , f1.txt
```

(b) executed on client; this will tell client what all things server can export to that client

```
# showmount -e 1.2.3.4 ←
o/p: /data , f1.txt
IP of client is 192.168.2.4
```

Q.3(e) How to encrypt and decrypt files in Linux using GPG? [5]
Ans.:



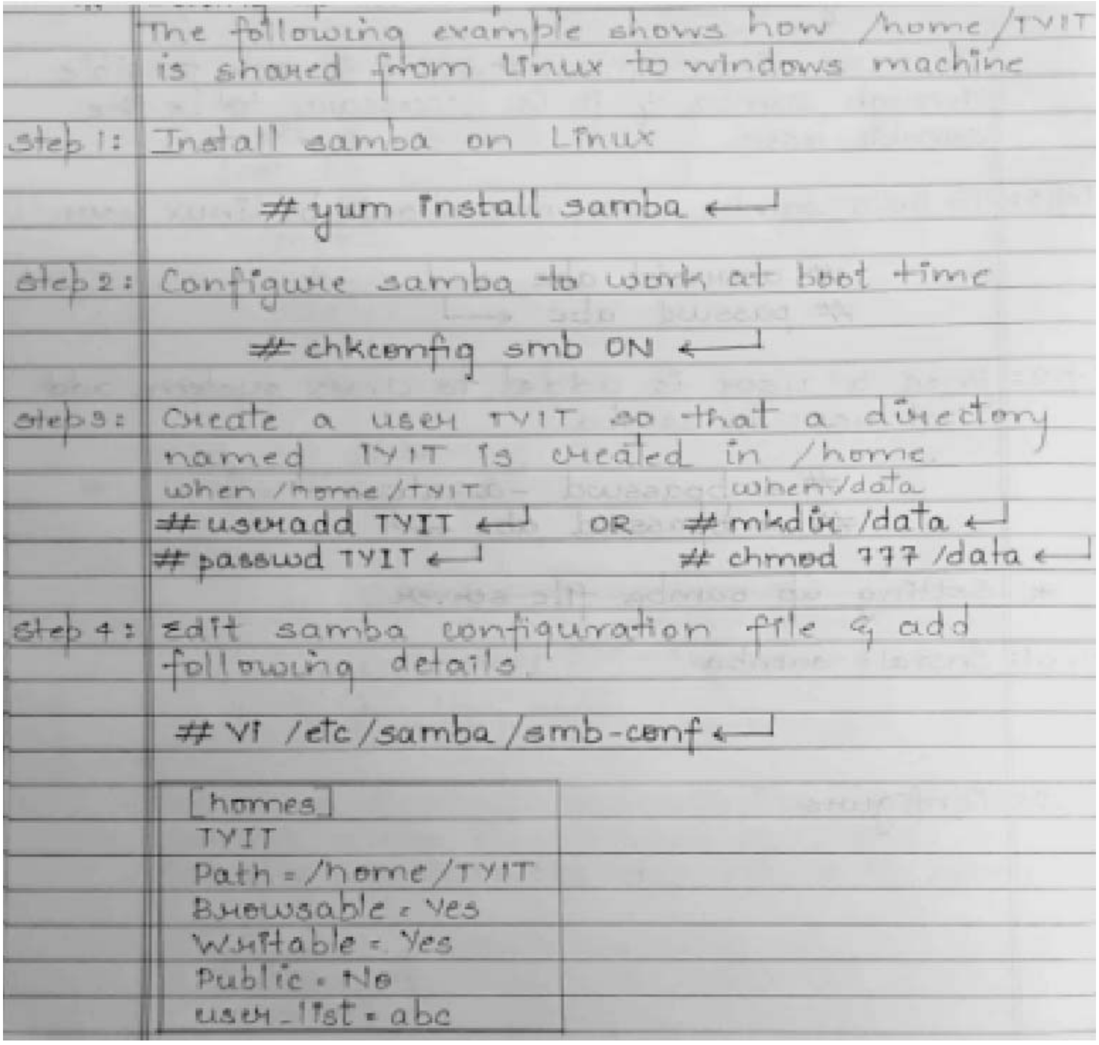
Encrypting and Decrypting Files

In this exercise, you'll use the user accounts linda and lisa you created in the previous exercise to practice GPG file encryption and decryption.

1. Open a shell, and use **su - linda** to become user linda.
2. As linda, copy the file /etc/hosts to your home directory using **cp /etc/hosts ~**.
3. Use **gpg --listkeys** to list the keys currently imported in Linda's environment, and note the exact name of the user lisa.
4. Encrypt the file using **gpg -e hosts**. When the user account is requested, enter the exact name of user lisa as you found it in the previous step of this exercise. Next press Enter on an empty line to complete the encryption procedure.
5. Use **cp ~/hosts.gpg /tmp** to copy the gpg file to the tmp directory where lisa can see and read it.
6. Use **exit** to log out as linda, and now use **su - lisa** to become user lisa.
7. As lisa, use **gpg -d /tmp/hosts.gpg** to decrypt the hosts file.

Q.3(f) Show the configuration for sharing / home / TYIT from Linux to Windows. [5]

Ans.:



The following example shows how /home/TYIT is shared from Linux to windows machine

step 1: Install samba on Linux

```
# yum install samba ←┘
```

step 2: Configure samba to work at boot time

```
# chkconfig smb ON ←┘
```

step 3: Create a user TYIT so that a directory named TYIT is created in /home

when /home/TYIT when /data

```
# useradd TYIT ←┘      OR      # mkdir /data ←┘
# passwd TYIT ←┘               # chmod 777 /data ←┘
```

step 4: Edit samba configuration file & add following details

```
# vi /etc/samba/smb-conf ←┘
```

```
[homes]
TYIT
Path = /home/TYIT
Browsable = Yes
Writable = Yes
Public = No
user_list = abc
```

Q.4Attempt the following (any THREE) [15]

Q.4(a) How to configure DHCP server? [5]

Ans.:

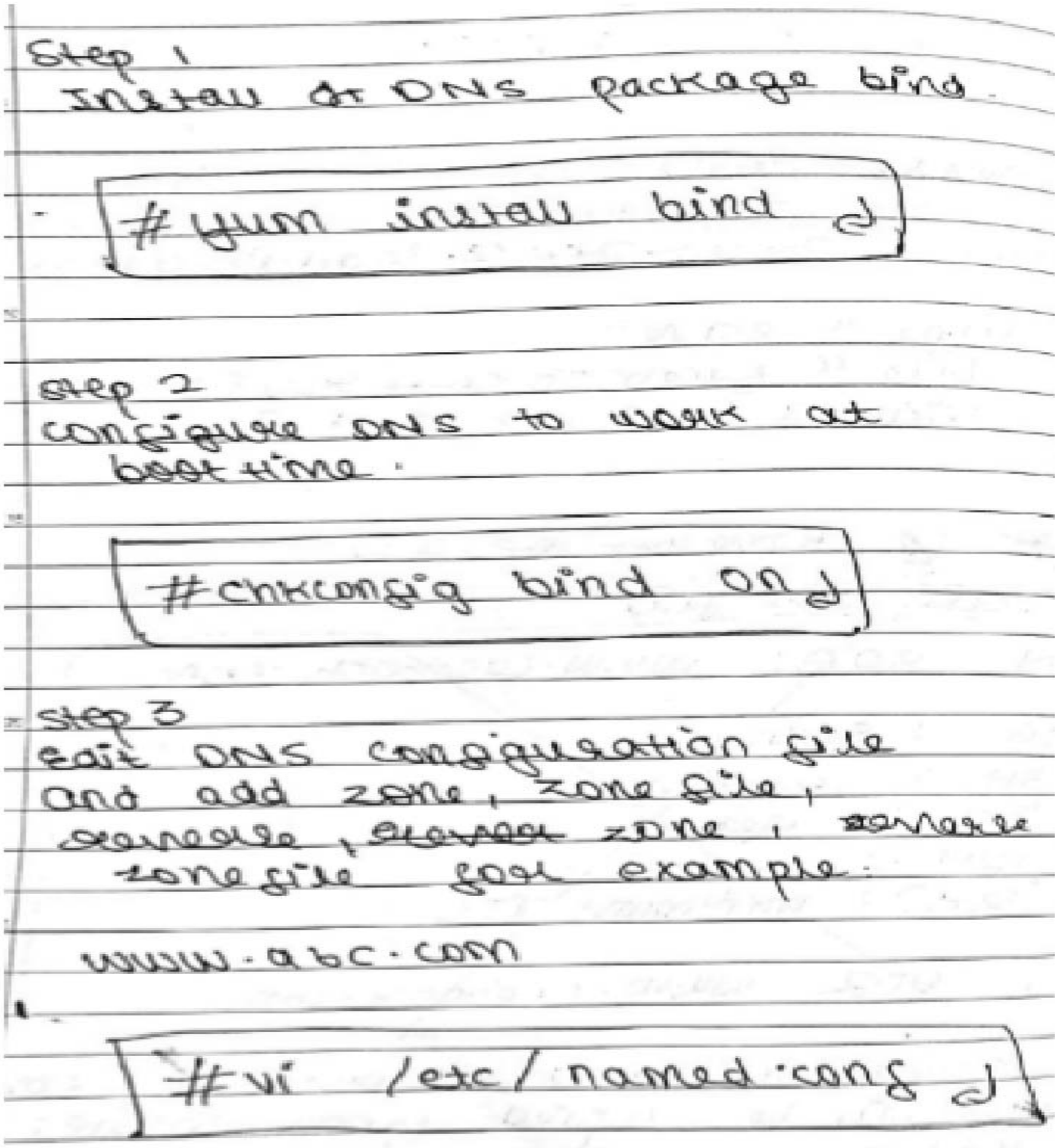
1. Start the virtual machine, and open a root shell. From the root shell, use the command yum -y dhcp to install the DHCP server.

2. Open the fi le /etc/dhcp/dhcpd.conf with an editor, and give it the following contents. Make sure that the names and IP addresses used in this example match your network:

```
option domain-name "example.com";

 option domain-name-servers YOUR.DNS.SERVERNAME.HERE;

default-lease-time 600;

max-lease-time 1800;
 subnet 192.168.100.0

 netmask 255.255.255.0

 {

 range 192.168.100.10 192.168.100.20;

  options routers 192.168.100.1;

 }
```

3. Start the DHCP server by using the command service dhcpd start, and enable it using chkconfig dhcpd on.

4. Start the second virtual machine. Make sure that the network card is set to get an IP address from a DHCP server. After starting it, verify that the DHCP server has indeed handed out an IP address.



Step 1
Install ~~or~~ DNS package bind.

```
# yum install bind ↵
```

Step 2
configure DNS to work at boot time.

```
#chkconfig bind on ↵
```

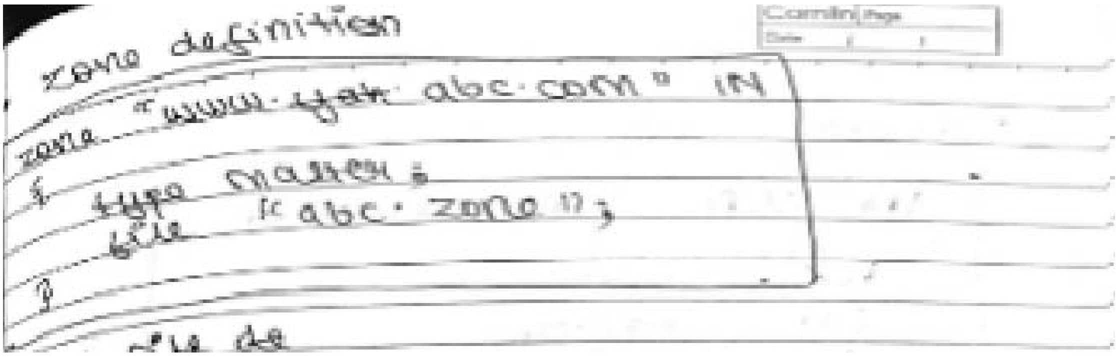
Step 3
edit DNS configuration file and add zone, zone file, reverse, reverse zone, reverse zone file for example.

www.abc.com

```
#vi /etc/named.conf ↵
```

Q.4(b) Show the configuration of Primary DNS server. **[5]**

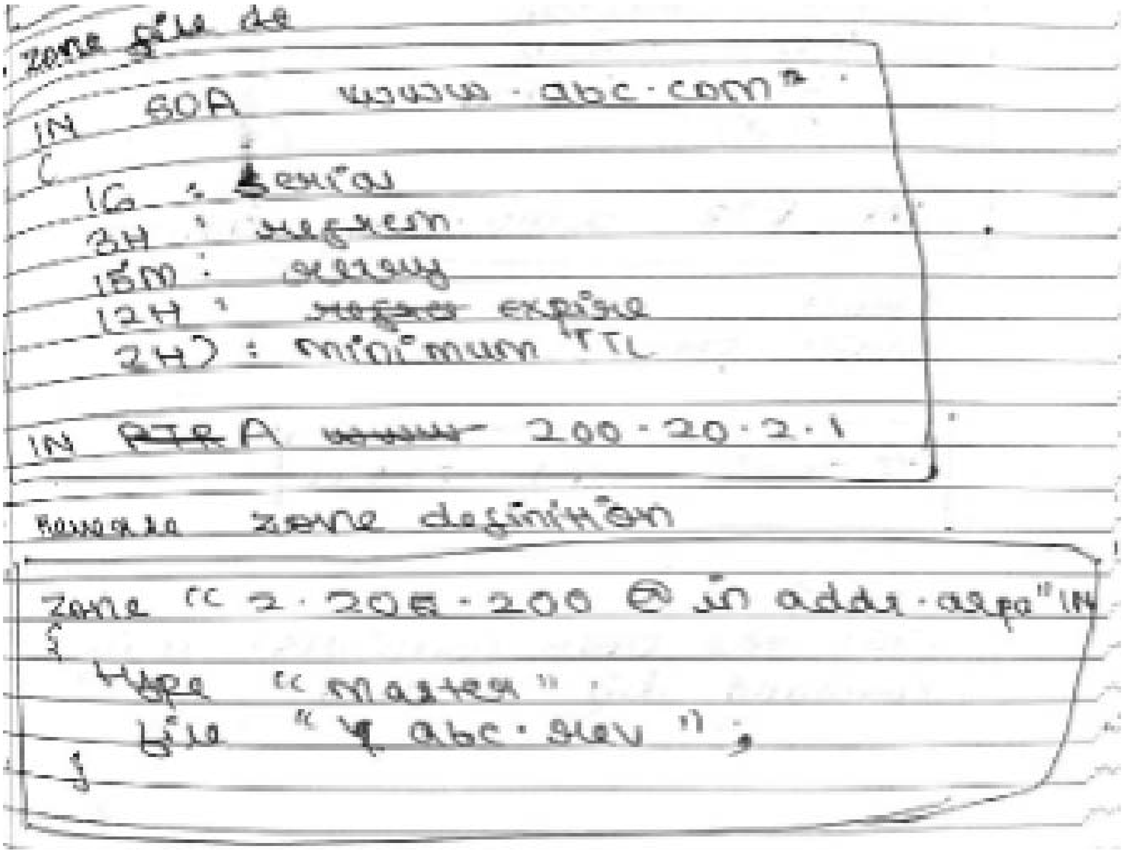
Ans.:



zone definition

```
zone "www.tyit abc.com" IN
{
  type master;
  file "abc.zone";
}
```

file de



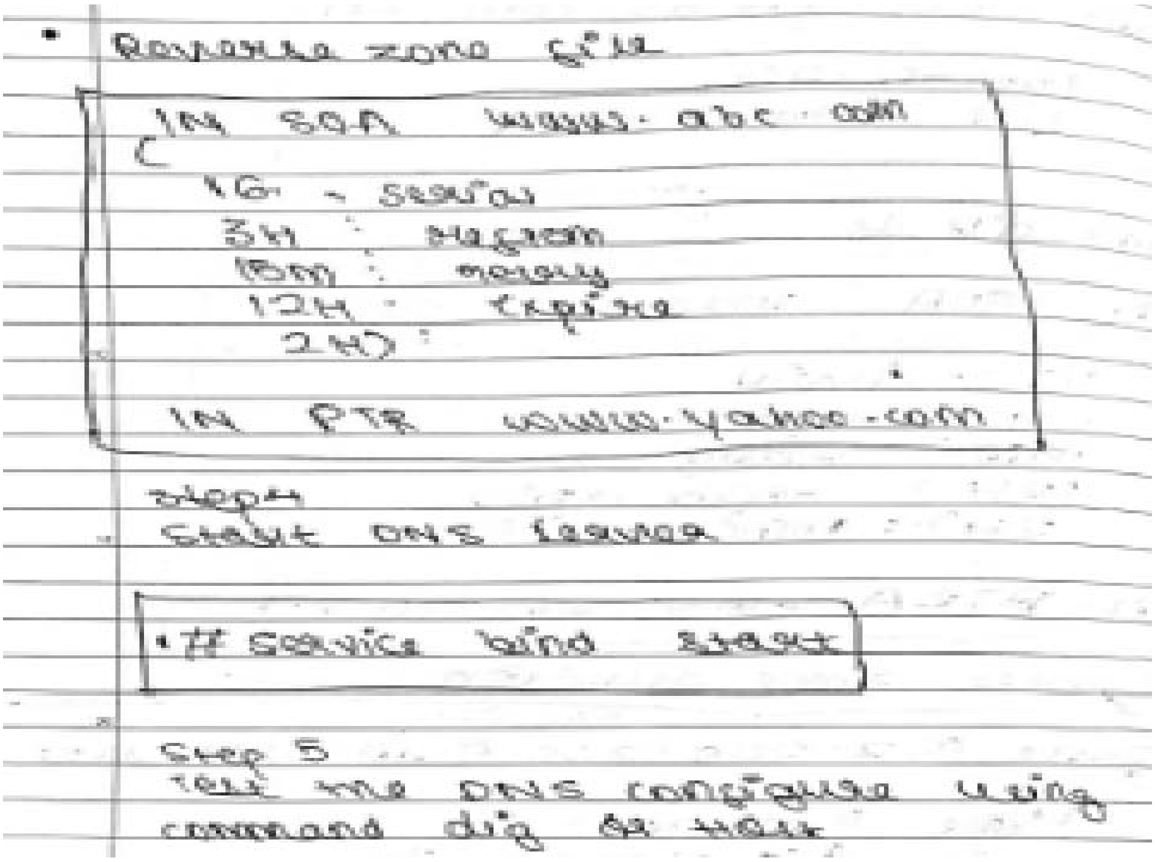
zone file de

IN SOA www.abc.com.
(
1G : serial
3H : refresh
15M : retry
12H : ~~refres~~ expire
2H) : minimum TTL

IN ~~PTR~~ A ~~www~~ 200.20.2.1

Reverse zone definition

```
zone "2.208.200 @ in addr.arpa" IN
{
 type "master";
 file "r abc.rev";
}
```



• Reverse zone file

```
IN SOA www.abc.com
(
  1G - serial
  3H : refresh
  15M : retry
  12H : expire
  2H) :

IN PTR www.yahoo.com.
```

step 4
Start DNS server

```
• # service bind start
```

Step 5
test the DNS configure using command dig or host

Q.4(c) How to configure Virtual Host? [5]

Ans.:

Virtual host :

Theortically one webserver can host only one website

To host multiple webserve sites, multiple web-servers are required which will not only increase the cost of implementation but also make the architecture complex.

Solution to above problem is :

Create server process of web servers inside physical web server, which are known as virtual hosts

Every virtual host will host single web site. The virtual host are created using following 2 approaches

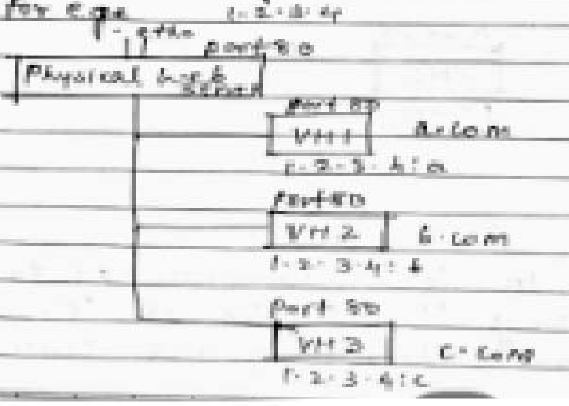
(1) Name (host) based VH.

(2) IP based VH.

Name based VH :

(1) Most commonly used architecture as for the outside world all VH would be identified by same IP.

for eg. 1.2.3.4



URL	VIP
a.com	1.2.3.4:a
b.com	1.2.3.4:b
c.com	1.2.3.4:c

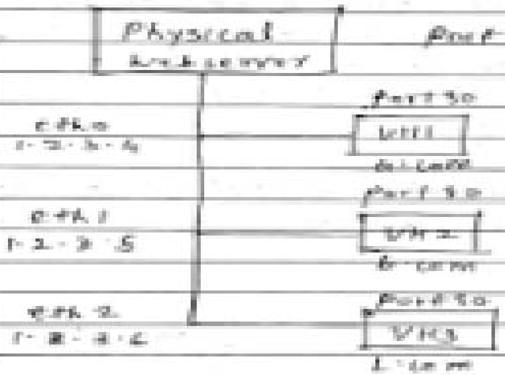
whether it is a.com, b.com or c.com, it would be identified by outside world through 1.2.3.4 only.

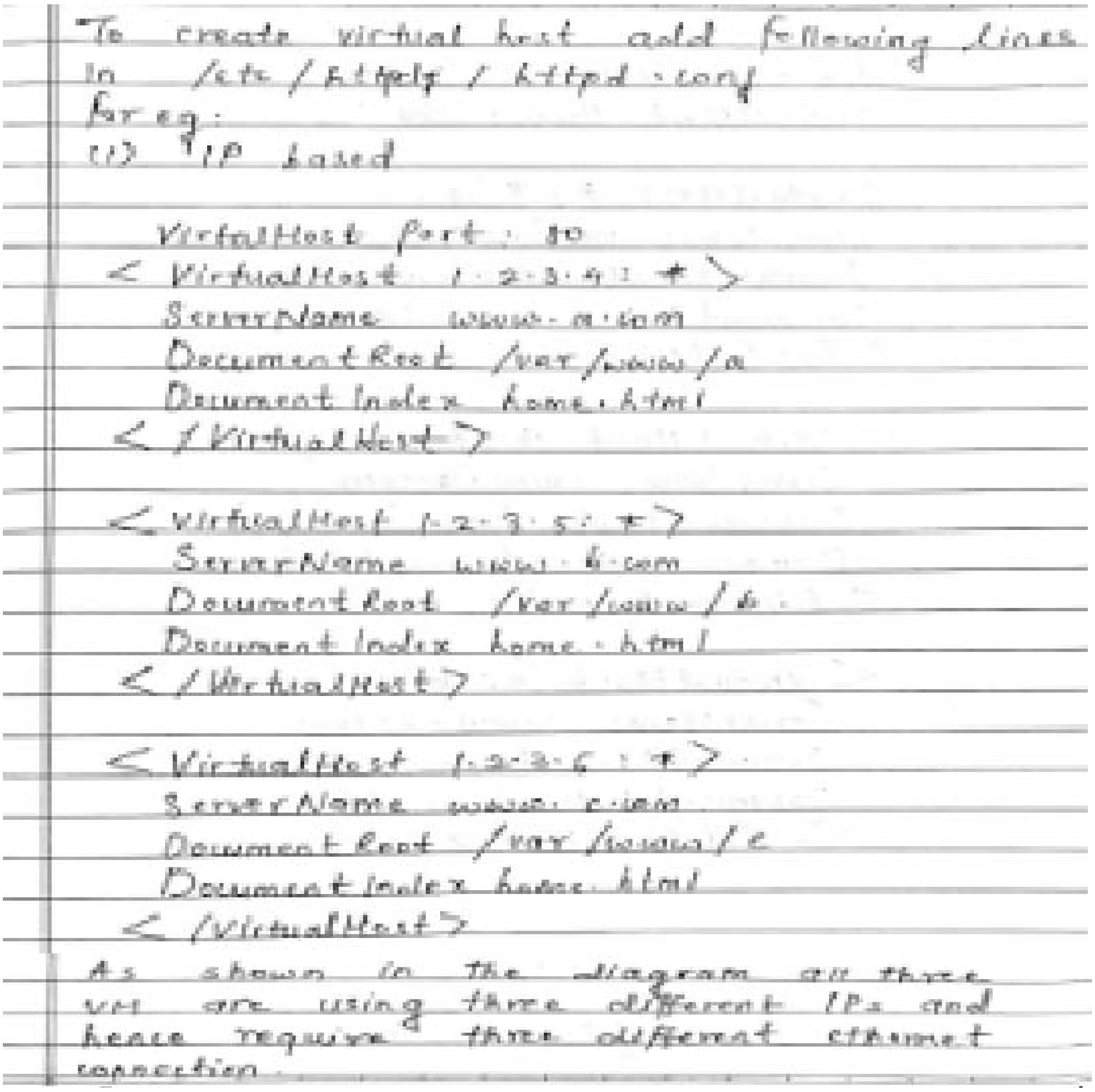
For physical server to search VH it assigns private IP (VIP)

for eg : 1.2.3.4 : a For VH1

(2) IP based VH :

This approach is used for security purpose where every VH is identified by different IP





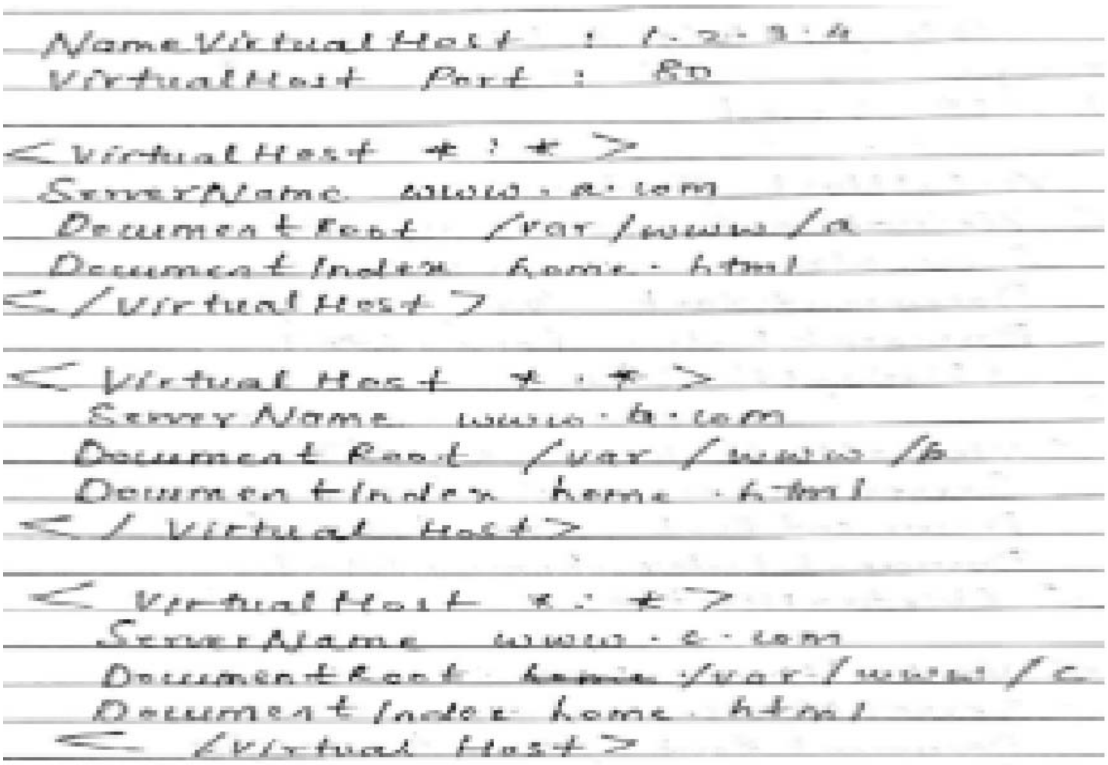
To create virtual host add following lines in /etc/httpd/httpd.conf

for eg:

(1) IP based

```
    VirtualHost port : 80
  <VirtualHost 1.2.3.4:*>
    ServerName www.a.com
    DocumentRoot /var/www/a
    Document Index home.html
  </VirtualHost>

  <VirtualHost 1.2.3.5:*>
    ServerName www.b.com
    DocumentRoot /var/www/b.
    Document Index home.html
  </VirtualHost>

  <VirtualHost 1.2.3.6:*>.
    ServerName www.c.com
    DocumentRoot /var/www/c
    DocumentIndex home.html
  </VirtualHost>
```

As shown in the diagram all three vh are using three different IPs and hence require three different ethernet connection.



```
 NameVirtualHost : 1.2.3.4
 VirtualHost Port : 80

<VirtualHost *:*>
 ServerName www.a.com
  DocumentRoot /var/www/a
  DocumentIndex home.html
</VirtualHost>

<Virtual Host *:*>
   Server Name www.b.com
  Document Root /var/www/b
  DocumentIndex home.html
 </Virtual Host>

 <VirtualHost *:*>
  ServerName www.c.com
  DocumentRoot ~~home~~ /var/www/c
  DocumentIndex home.html
 </Virtual Host>
```

Q.4(d) Show how Linux send a mail to External User. **[5]**

Ans.: Sending a Message to an External User

In this exercise, you're going to send an email message to a user on another computer. This means that two instances of Postfix will need to communicate with one another. Make sure to follow all of the instructions so that all prerequisites have been set properly,

1. On the host computer, use virsh list. If you performed all previous exercises, you'll see a virtual machine with the name testvm. Use virsh start testvmto start this virtual machine.
2. On the host computer, start the Virtual Machine Manager using the virt-manager command. Open a console on the virtual machine, and note its IP address.
3. On both the host and the virtual machine, edit the /etc/hosts file and include a line for the host computer and the virtual machine. The purpose for doing this is that these two computers can then resolve one another.
4. On the virtual machine, use useradd lisa to create a user with the name lisa.
5. On the host computer, use su - linda to become linda and start Mutt, From the Mutt interface, type m to start composing a new mail message. Enter lisa@testvm.example.com in the to field. In the Subject field, enter test message l. Enter some text in the mail message, and press y to send it.
6. Open a shell on testvm, and as root, use yum -y install mutt to install Mutt. Next, use su - lisa to log in as lisa and start Mutt. You'll notice that lisa's mailbox is empty, and the message that user linda sent from the other machine has not yet been sent.

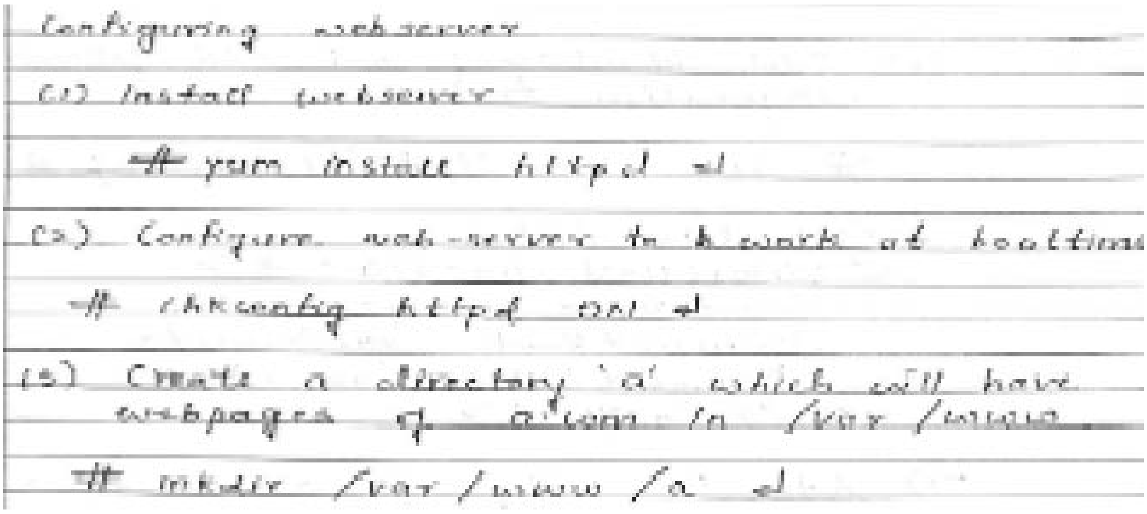
Q.4(e) Show the basic configuration of dovecot. **[5]**

Ans.:

1. Install dovecot.
2. Run /usr/libexec/dovecot/mkcert.sh to create some self-signed certificates for Dovecot, and install them in the right locations.
3. Use service dovecot start to start Dovecot.
4. As root, make sure that user linda has a message in her mailbox by entering the following command: mail-s hello linda <. This sends an empty message to user linda that has only a subject line.
5. Use su-to become linda, and as linda, start Mutt. Download from Wow! eBook <www.wowebook.com>
6. From Mutt, hit c to change the mailbox you're accessing, and enter the URL pop:// linda@localhost. This should give you access to your mailbox on the local computer.
7. Use c once more, and enter the URL pop3://linda@localhost. You'll now have access to the mailbox using the TLS version of POP3.

Q.4(f) Show the configuration of how web server will host a website? **[5]**

Ans.:



Configuring web server

(1) Install webserver

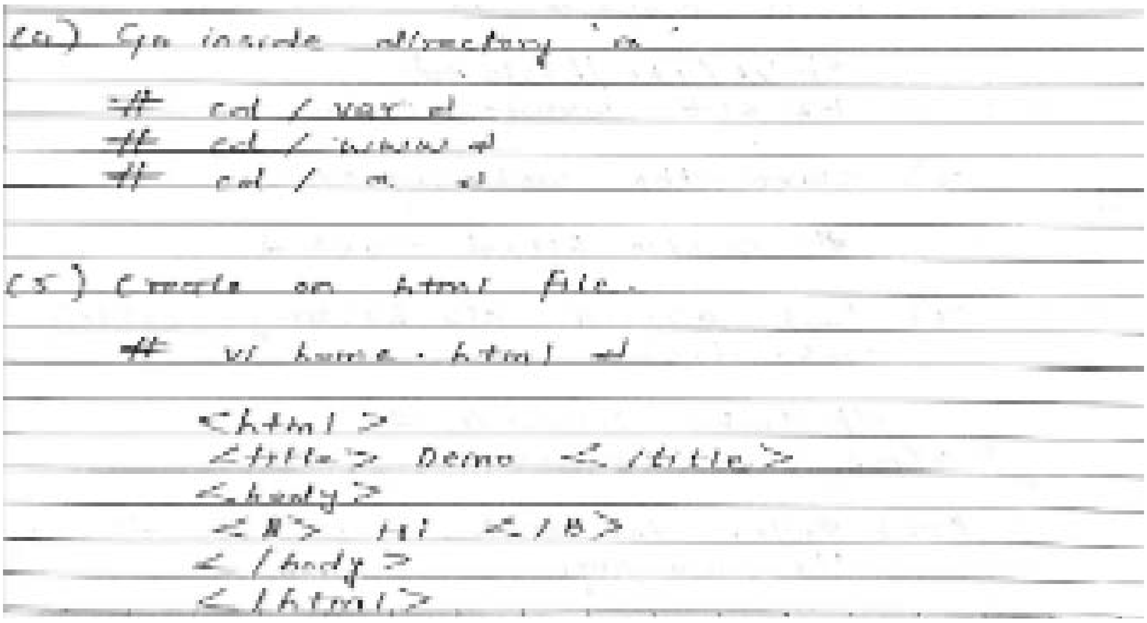
yum install httpd ↵

(2) Configure web-server to b work at booting

chkconfig httpd on ↵

(3) Create a directory 'a' which will have webpages of a.com in /var/www

mkdir /var/www/a ↵

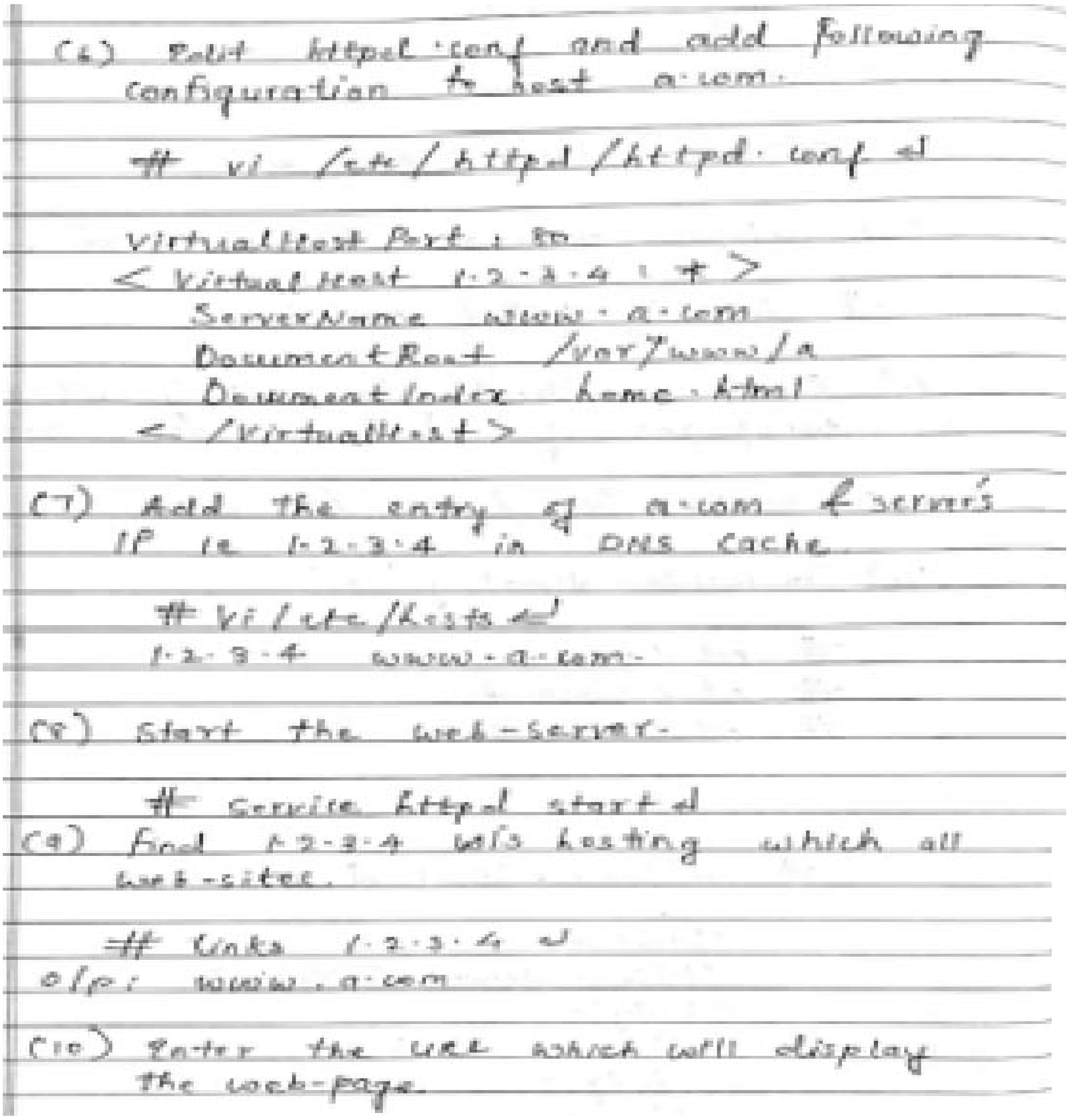


(4) Go inside directory 'a'

```
# cd / var ↵
# cd / www ↵
# cd / a ↵
```

(5) Create an html file.

```
# vi home.html ↵

    <html>
    <title> Demo </title>
    <body>
     <B> Hi </B>
    </body>
    </html>
```



(6) Edit httpd.conf and add following configuration to host a.com.

```
# vi /etc/httpd/httpd.conf ↵

VirtualHost Port : 80
< VirtualHost 1.2.3.4 : * >
    ServerName www.a.com
    DocumentRoot /var/www/a
    DocumentIndex home.html
</VirtualHost>
```

(7) Add the entry of a.com & server's IP ie 1.2.3.4 in DNS cache.

```
# vi /etc/hosts ↵
1.2.3.4   www.a.com.
```

(8) Start the web-server.

```
# service httpd start ↵
```

(9) Find 1.2.3.4 web's hosting which all web-sites.

```
# links 1.2.3.4 ↵
o/p: www.a.com
```

(10) Enter the URL which will display the web-page.

Q.5Attempt the following (any THREE) **[15]**

Q.5(a) What are the requirement of HA Clustering. **[5]**

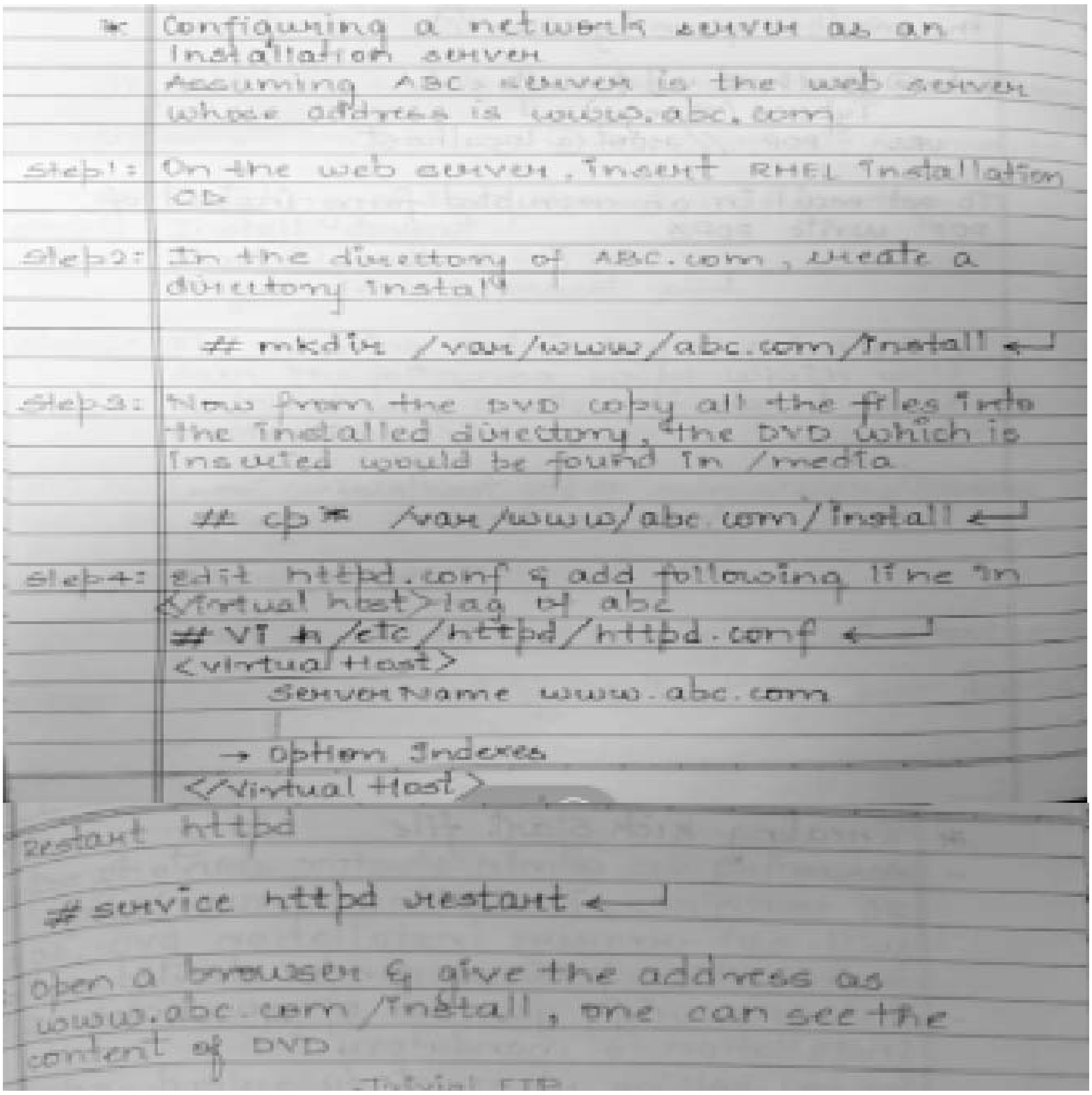
Ans.: High-Availability Requirements Several components are involved in setting up an HA cluster. In its simplest form, two servers are connected to each other by a network cable. But a cluster that is configured in a simple, nonredundant way like this can create some unpleasant surprises. Imagine, for instance, that the network cable breaks In that case, both nodes in the cluster would no longer see each other, and there would be no way for one node to find out what the other node is doing. This also means that, in such a scenario, there would be no way to make sure that the cluster service is started on the right machine and that its services would continue to be offered. That is why a typical Red Hat cluster involves more than just two nodes.

The following items are typically used in HA clusters:

- Multiple nodes
- Fence devices
- Ethernet bonding
- Shared storage

Q.5(b) How to configure Network server as an Installation Server? **[5]**

Ans.:



* Configuring a network server as an installation server

Assuming ABC server is the web server whose address is www.abc.com

Step1: On the web server, Insert RHEL Installation CD

Step2: In the directory of ABC.com, create a directory install

```
# mkdir /var/www/abc.com/Install ←
```

Step3: Now from the DVD copy all the files into the Installed directory, the DVD which is Inserted would be found in /media

```
# cp * /var/www/abe com/Install ←
```

Step4: edit httpd.conf & add following line in <Virtual host> lag of abc

```
# VI -n /etc/httpd/httpd.conf ←
<virtual Host>
        ServerName www.abc.com

   → option Indexes
</Virtual Host>
```

Restart httpd

```
# service httpd restart ←
```

open a browser & give the address as www.abc.com/Install, one can see the content of DVD

Trivial FTP

Q.5(c) Write a short note on Kick start file and its Configuration. **[5]**

Ans.: Creating kick start file

- Assuming an administrator wants to set up 20 machines, installing linux today will not require installation DVD, so a installation process can be started online.
- But answering few set of questions while installation is mandatory.

for example, setting up root password, selecting language, selecting timezone, setting up partition, etc.

- So, answering this for every installation is same what complex from administrator point of view.
- So, to handle this automatically there is a kick start file present in every users home directory known as Anaconda–ks.cfg
- This file contains all the answer to the questions asked during installation, so if this file is made available, no need to answer all the questions for remaining 19 installations. If would be done automatically.

Step 1: On the installation server abc.com copy kick start file of root in abc.com directory.

```
#cp anaconda-ks.cfg /var/www/abc.com ↵
```

Step 2: Start the virtual machine manager, click on create virtual machine and select network install.

Step 3: Now give 2 addresses, installation address and kickstart file address :

(a) Installation address :	www.abc.com/install
(b) Kick start address :	www.abc.com/anaconda-ks.cfg

Click on next option, there would be series of messages where user needs to simply click on next. If there is no error message and messages stop appearing it means kick start file is loaded successfully.

Now whenever client tries online installation it should open the browser and enter 2 address.

(1) www.abc.com/install

(2) www.abc.com/anaconda-ks.cfg

Q.5(d) Write a script to print sum of N Natural numbers. [5]

Ans.:

```
# ! /bin/bash

echo "enter the value of n:"
read n
for ((i = 1; i <= n; i++))
do
echo "$i"

done

exit
```

Q.5(e) Write a script to monitor user login, also explain Until Loop. [5]

Ans.: Whereas while works as long as a certain condition is met, until is just the opposite; that is, it runs until the condition is met.

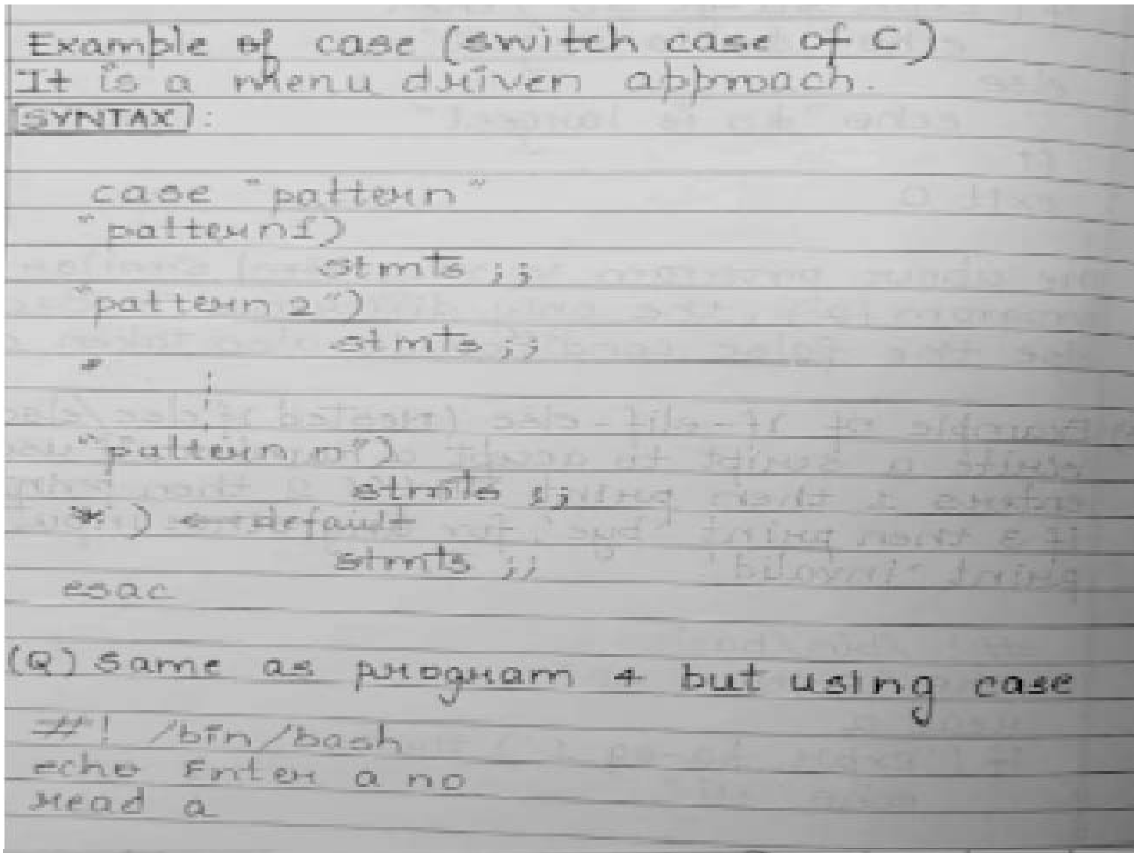
```
#!/bin/bash
#usermon
#script that alerts when a user logs in
#usage: ishere <username>

until who | grep $1 >> /dev/null
do
    echo $1 is not logged in yet
    sleep 5
done
```

echo $1 has just logged in

exit 0

Q.5(f) With example explain case – esac. **[5]**

Ans.: It is a menu driven conditional statement.



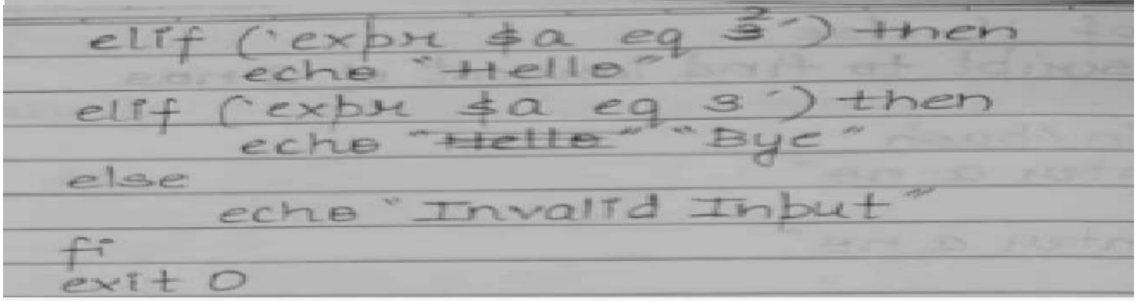
Example ~~of~~ case (switch case of C)
It is a menu driven approach.

SYNTAX :

```
  case "pattern"
  "pattern1)
            stmts ;;
  "pattern 2")
            stmts ;;
  #
      :
      :
  "pattern n")
            stmts ;;
  *) ~~<-default~~
            stmts ;;
  esac
```

(Q) Same as program 4 but using case

```
#! /bin/bash
echo Enter a no
read a
```



```
  elif (`expr $a eq 2`) then
        echo "Hello"
  elif (`expr $a eq 3`) then
        echo "~~Hello~~" "Bye"
  else
        echo "Invalid Input"
  fi
  exit 0
```

❑ ❑ ❑ ❑ ❑