Q.1a **What are different Linux distribution? Explain each in brief**

Ans. A Linux distribution is a collection of software applications built on top of the Linux kernel and operating system.

A typical Linux distribution comprises

- a Linux kernel,
- GNU tools and libraries,
- documentation,
- a window system (the most common being the X Window System),
- a window manager,
- And a desktop environment.

Most distributions come ready to use and pre-compiled, while some distributions (such as Gentoo) are distributed in source code form and compiled during installation.

Some of them are:

1. Red Hat Enterprise Linux (RHEL):

RHEL is a universal distribution, since it contains all the features of Linux. It is targeted towards commercial market. Red Hat invented the RPM package system used on several distributions which is meant to automatically install and uninstall softwares.

2. Debian:

This was used as a non-commercial O.S. Debian has been used as a platform for other Linux distributions. Debian is an operating system composed only of free, open-source software.

3. Slackware:

Slackware aims for design stability and simplicity and to be the most "Unix-like" Linux distribution. The Official Release of Slackware Linux is an advanced Linux operating system, designed with the twin goals of ease of use and stability as top priorities.

4. **SUSE:**

SUSE Linux is of German origin, an acronym of “Software und System-Entwicklung”. The first version appeared in 1994, making SUSE one of the oldest existing commercial distributions.

5. **Turbolinux:**

The Turbolinux distribution is a Japanese Linux distribution targeting Asian users.

6. **Ubuntu:**

It is a Debian-based Linux O.S. and distribution. It is based on free software. It’s developer is Canonical Ltd.

7. **Fedora:**

It is a project with a strong focus on free software. Fedora is sponsored by Red Hat. It is the foundation for Red Hat Enterprise Linux.

8. **CentOS:**

The CentOS Linux distribution is a reproducible platform derived from the sources of Red Hat Enterprise Linux (RHEL). It aims to be functionally compatible with its upstream source, Red Hat Enterprise Linux (RHEL).

9. **Gentoo:**

Gentoo is a free operating system based on either Linux or FreeBSD that can be automatically optimized and customized for just about any application or need. It is designed for advanced Linux users.

10. Mandriva Linux:

Mandriva Linux is a Linux distribution by Mandriva. It uses the RPM Package Manager.

Q.1b **Explain the duty of the Linux system administrator in backing up and restoring files.**

Ans.

1) Until equipment becomes absolutely infallible, and until people lose their desire to harm the property of others (and, truth be known, until system administrators become perfect), there is a need to back up important files so that in the event of a failure of hardware, security, or administration, the system can be up and running again with minimal disruption.
2) Only the system administrator may do this.
3) Because of its built-in security features, Linux may not allow users to be able even to back up their own files to floppy disks.
4) Again, knowing that file backup is our job is not enough. We need to formulate a strategy for making sure our system is not vulnerable to catastrophic disruption and it’s not always obvious.
5) If we have a high-capacity tape drive and several good sets of restore diskettes, we might make a full system backup every few days.
6) Once we’ve decided what to back up, we need to decide how frequently we want to perform backups and whether we wish to maintain a series of incremental backups ,adding only the files that have changed since the last backup or multiple full backups, and when these backups are to be performed or, if we have input as to the equipment used, do we want to use a redundant array of independent disks, or RAID, which is to say multiple hard drives all containing the same data as insurance against the failure of any one of them, in addition to other backup systems.
7) A RAID is not enough, because hard drive failure is not the only means by which a system can be brought to a halt.
8) Part of our strategy should be the maintenance of perfect backups without ever needing to resort to them.
9) This means encouraging users to keep multiple copies of their own important files, all in their home directories, so that you are not being asked to mount a backup so as to restore a file that a user has corrupted.

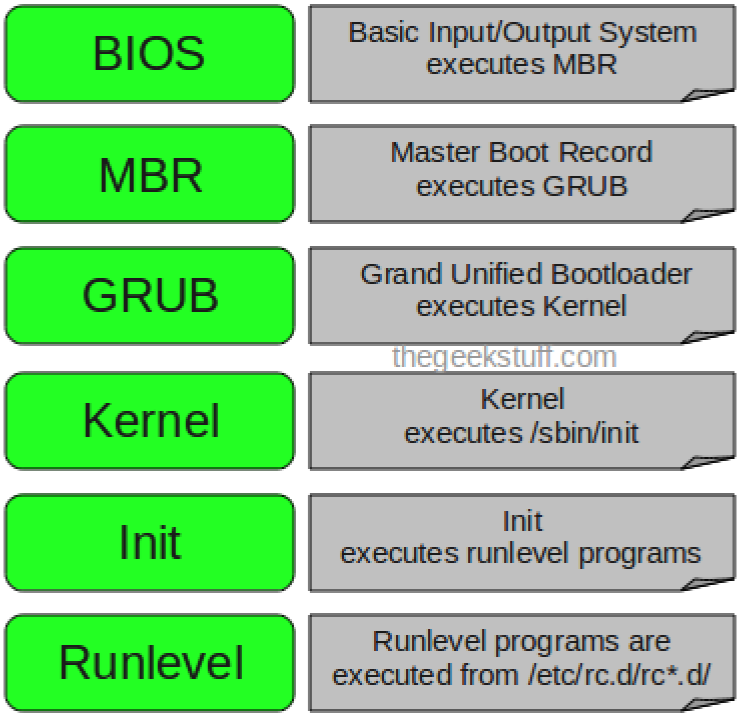
10) Backing up is only half the story, too. You need to formulate a plan for bringing the system back up in the event of a failure.

Sometimes hardware failures are so severe that the only solution is replacing the hard drive, replacing everything except the hard drive, or even restoring from backup to a whole new machine.

1 c **Explain the booting process of the Linux operating system in detail.**

Ans Have you ever wondered what happens behind the scenes from the time you press the power button until the Linux login prompt appears?

The following are the 6 high level stages of a typical Linux boot process.


BIOS
Basic Input/Output System executes MBR
MBR
Master Boot Record executes GRUB
GRUB
Grand Unified Bootloader executes Kernel
thegeekstuff.com
Kernel
Kernel executes /sbin/init
Init
Init executes runlevel programs
Runlevel
Runlevel programs are executed from /etc/rc.d/rc*.d/

1. BIOS

- BIOS stands for Basic Input/Output System
- Performs some system integrity checks
- Searches, loads, and executes the boot loader program.
- It looks for boot loader in floppy, cd-rom, or hard drive. You can press a key (typically F12, but it depends on your system) during the BIOS startup to change the boot sequence.
- Once the boot loader program is detected and loaded into the memory, BIOS gives the control to it.
- So, in simple terms BIOS loads and executes the MBR boot loader.

2. MBR

- MBR stands for Master Boot Record.
- It is located in the 1st sector of the bootable disk. Typically /dev/hda, or /dev/sda
- MBR is less than 512 bytes in size. This has three components 1) primary boot loader info in 1st 446 bytes 2) partition table info in next 64 bytes 3) mbr validation check in last 2 bytes.
- It contains information about GRUB (or LILO in old systems).
- So, in simple terms MBR loads and executes the GRUB boot loader.

3. GRUB

- GRUB stands for Grand Unified Bootloader.
- If you have multiple kernel images installed on your system, you can choose which one to be executed.
- GRUB displays a splash screen, waits for few seconds, if you don't enter anything, it loads the default kernel image as specified in the grub configuration file.
- GRUB has the knowledge of the filesystem (the older Linux loader LILO didn't understand filesystem).
- Grub configuration file is /boot/grub/grub.conf (/etc/grub.conf is a link to this). The following is sample grub.conf of CentOS.

```
#boot=/dev/sda
default=0
timeout=5
splashimage=(hd0,0)/boot/grub/splash.xpm.gz
hiddenmenu
title CentOS (2.6.18-194.el5PAE)
          root (hd0,0)
          kernel /boot/vmlinuz-2.6.18-194.el5PAE ro root=LABEL=/
          initrd /boot/initrd-2.6.18-194.el5PAE.img
```

- As you notice from the above info, it contains kernel and initrd image.
- So, in simple terms GRUB just loads and executes Kernel and initrd images.

4. Kernel

- Mounts the root file system as specified in the "root=" in grub.conf
- Kernel executes the /sbin/init program
- Since init was the 1st program to be executed by Linux Kernel, it has the process id (PID) of 1. Do a 'ps -ef | grep init' and check the pid.
- initrd stands for Initial RAM Disk.
- initrd is used by kernel as temporary root file system until kernel is booted and the real root file system is mounted. It also contains necessary drivers compiled inside, which helps it to access the hard drive partitions, and other hardware.

5. Init

- Looks at the /etc/inittab file to decide the Linux run level.
- Following are the available run levels

  - 0 – halt
  - 1 – Single user mode
  - 2 – Multiuser, without NFS
  - 3 – Full multiuser mode
  - 4 – unused
  - 5 – X11
  - 6 – reboot
- Init identifies the default initlevel from /etc/inittab and uses that to load all appropriate program.
- Execute 'grep initdefault /etc/inittab' on your system to identify the default run level
- If you want to get into trouble, you can set the default run level to 0 or 6. Since you know what 0 and 6 means, probably you might not do that.
- Typically you would set the default run level to either 3 or 5.

6. Runlevel programs

- When the Linux system is booting up, you might see various services getting started. For example, it might say "starting sendmail …. OK". Those are the runlevel programs, executed from the run level directory as defined by your run level.
- Depending on your default init level setting, the system will execute the programs from one of the following directories.
  - Run level 0 – /etc/rc.d/rc0.d/
  - Run level 1 – /etc/rc.d/rc1.d/
  - Run level 2 – /etc/rc.d/rc2.d/
  - Run level 3 – /etc/rc.d/rc3.d/
  - Run level 4 – /etc/rc.d/rc4.d/
  - Run level 5 – /etc/rc.d/rc5.d/
  - Run level 6 – /etc/rc.d/rc6.d/
- Please note that there are also symbolic links available for these directory under /etc directly. So, /etc/rc0.d is linked to /etc/rc.d/rc0.d.
- Under the /etc/rc.d/rc*.d/ directories, you would see programs that start with S and K.
- Programs starts with S are used during startup. S for startup.
- Programs starts with K are used during shutdown. K for kill.
- There are numbers right next to S and K in the program names. Those are the sequence number in which the programs should be started or killed.
- For example, S12syslog is to start the syslog deamon, which has the sequence number of 12. S80sendmail is to start the sendmail daemon, which has the sequence number of 80. So, syslog program will be started before sendmail.

Q.1d	**Explain the Extended 3 file system.**
Ans.	1) The extended 3 file system is a new file system introduced in Red Hat 7.2. ext3 provides all the features of ext2, and also features journaling and backward compatibility with ext2. 2) The backward compatibility enables to still run kernels that are only ext2 aware with ext3 partitions. 3) We can upgrade an ext2 file system to an ext3 file system without losing any of your data. 4) ext3's journaling feature speeds up the amount of time it takes to bring the file system back to a sane state if it's not been cleanly unmounted (that is, in the event of a power outage or a system crash). 5) Under ext2, when a file system is uncleanly mounted, the whole file system must be checked. This takes a long time on large file systems. 6) ext3 keeps a record of uncommitted file transactions and applies only those transactions when the system is brought back up. 7) A cleanly unmounted ext3 file system can be mounted and used as an ext2 file system. The kernel sees the ext3 file system as an ext2 file system. ext3's journaling feature involves a small performance hit to maintain the file system transaction journal. Therefore, it's recommended to use ext3 mostly for larger file systems, where the ext3 journaling performance hit is made up for in time saved by not having to run fsck on a huge ext2 file system.
2a	**What are the different types of shells in Linux? Explain.**
Ans	1. In addition to graphical user interfaces like Gnome, KDE and MATE, the Linux operating system also offers several shells. 2. These command-line interfaces provide powerful environments for software development and system maintenance. 3. Some of the shells available in Linux are: sh, bash, csh, tcsh, etc. 4. **sh**: The Bourne shell, called "sh," is one of the original shells, developed for Unix computers by Stephen Bourne at AT&T's Bell Labs in 1977. 5. It offers features such as input and output redirection, shell scripting with string and integer variables, and condition testing and looping. 6. **bash**: "Bash" -- the "Bourne-again Shell," based on sh -- has become the new default standard. 7. One attractive feature of bash is its ability to run sh shell scripts unchanged. Conveniences include command completion and a command history.

8. **zsh** is improved version of bash.

9. **csh and tcsh**: Using C syntax as a model, Bill Joy at Berkeley University developed the "C-shell," csh, in 1978.

10. Ken Greer, working at Carnegie-Mellon University, developed tcsh. Tcsh fixed problems in csh and added command completion, in which the shell makes educated "guesses" as you type.

11. **ksh:** David Korn developed the Korn shell, or ksh. Ksh is compatible with sh and bash.

12. Ksh improves on the Bourne shell by adding floating-point arithmetic, job control, command aliasing and command completion.

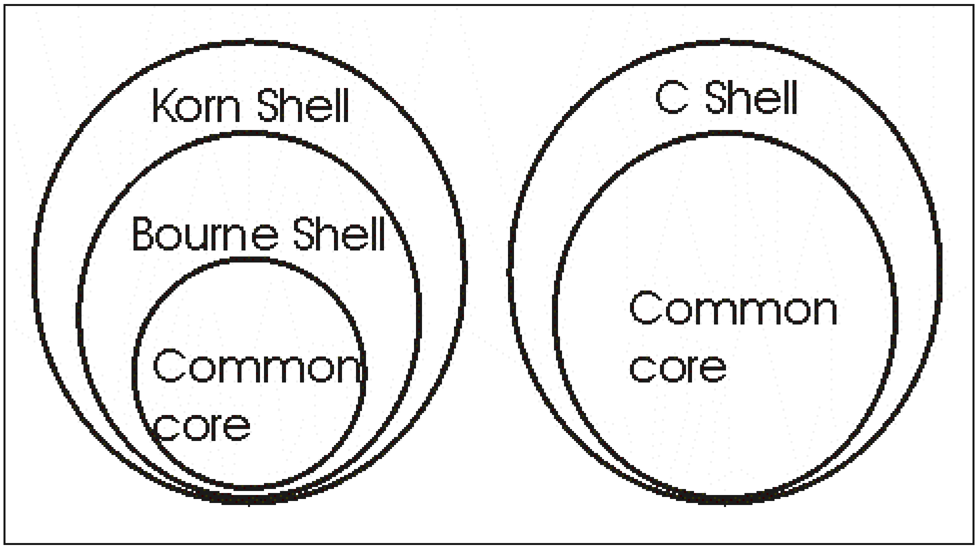


Fig.: Types of shells.

2b **Which are the network configuration files that are required to be changed while setting up a system or moving the system? Explain.**

Ans

The following are the network configuration files that are required to be changed while setting up a system or moving the system:

1. **/etc/resolv.conf** - host name resolver configuration file

   This configures Linux so that it knows which DNS server will be resolving domain names into IP addresses.

2. If using DHCP client, this will automatically be sent to you by the ISP and loaded into this file.

	3. If using a static IP address, ask the ISP or check another machine on your network 4. **/etc/hosts** - locally resolve node names to IP addresses When adding hosts to this file, place the fully qualified name first. 5. This informs Linux of local systems on the network which are not handled by the DNS server. 6. **/etc/nsswitch.conf** - system databases and name service switch configuration file The /etc/nsswitch.conf file is used to configure which services are to be used to determine information such as hostnames, password files, and group files. 7. **etc/sysconfig/network** The /etc/sysconfig/network file is used to specify information about the desired network configuration. 8. The following values may be used: ■ NETWORKING=value, where value is one of the following Boolean values: yes —Networking should be configured. no —Networking should not be configured. ■ HOSTNAME=value, where value should be the fully qualified domain name. 9. **/etc/sysconfig/network-scripts/ifcfg-eth0** To set up the IP addresses of your network interfaces make changes to this file.
2c	What is Network file system? Explain in detail. Explain the two important NFS configuration files for using the features of NFSv4.
Ans	• NFS is the most common method used to share files across Linux and UNIX networks. • It is a distributed file system that enables local access to remote disks and file systems. • NFS's operation is totally transparent to clients using remote file system if it is designed and carefully implemented. • Through NFS, it is possible to access files and directories that are physically located on another system or even in a different city or country using standard Linux commands. • No special procedures such as passwords are necessary to access the files. • NFS follows standard client/server architectural principles. • The server component of NFS consists of the physical disks that contain the file systems the user wants to share and several daemons that make these shared file systems visible to and available for use by client systems on the network. • When an NFS servers sharing a file system in this manner, it is said to be *exporting a file system*. • Similarly, the shared file system is referred to as an *NFS export*.

- The NFS server daemons provide remote access to the exported file systems, enable file locking over the network, and, optionally, allow the server administrator to set and enforce disk quotas on the NFS exports.

The two important NFS configuration files for using the features of NFSv4:

■ gssapi_mech.conf (new in NFSv4)
■ idmapd.conf (new in NFSv4)

If you intend to use NFSv4-specific features, you need to be familiar with the RPCSEC_GSS configuration files, /etc/gssapi_mech.conf and /etc/idmapd.conf. idmapd.conf is the configuration file for NFSv4's idmapd daemon. idmapd works on the behalf of both NFS servers and clients to translate NFSv4 IDs to user and group IDs and vice versa; idmapd.conf controls idmapd's runtime behavior. The default configuration (with comments and blank lines removed) should resemble:

```
[General]
Verbosity = 0
Pipefs-Directory = /var/lib/nfs/rpc_pipefs
Domain = localdomain
[Mapping]
Nobody-User = nobody
Nobody-Group = nobody
[Translation]
Method = nsswitch
```

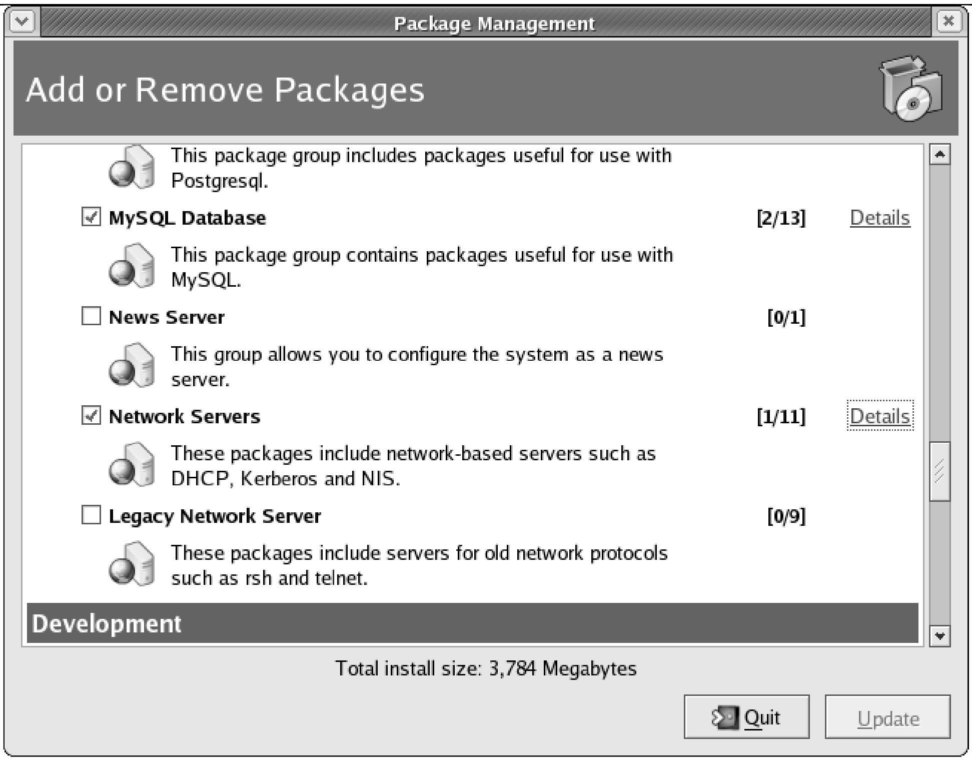
The /etc/gssapi_mech.conf file controls the GSS daemon (rpc .svcgssd). You won't need to modify this file. As provided in Fedora Core and RHEL, gssapi_mech.conf lists the specific function call to use to initialize a given GSS library. Programs (in this case, NFS) need this information if they intend to use secure RPC.

2d What is dynamic host configuration protocol? How are DHCP server and client configured on Linux?

Using Dynamic Host Configuration Protocol (DHCP), you can have an IP address and the other information automatically assigned to the hosts connected to your network. This method is quite efficient and convenient for large networks with many hosts, because the process of manually configuring each host is quite timeconsuming. By using DHCP, you can ensure that every host on your network has a valid IP address, subnet mask, broadcast address, and gateway, with minimum effort.

The program that runs on the server is dhcpd and is included as an RPM on the Fedora Core and Red Hat Enterprise Linux installation CDs. You can install it using the Package Management tool by following these instructions.

1. On Enterprise Linux choose Applications ⇨ System Settings ⇨ Add/Remove Applications from the top panel. The screen shown in Figure below appears.



Package Management

Add or Remove Packages

This package group includes packages useful for use with Postgresql.

☑ **MySQL Database** **[2/13]** Details

This package group contains packages useful for use with MySQL.

□ **News Server** **[0/1]**

This group allows you to configure the system as a news server.

☑ **Network Servers** **[1/11]** Details

These packages include network-based servers such as DHCP, Kerberos and NIS.

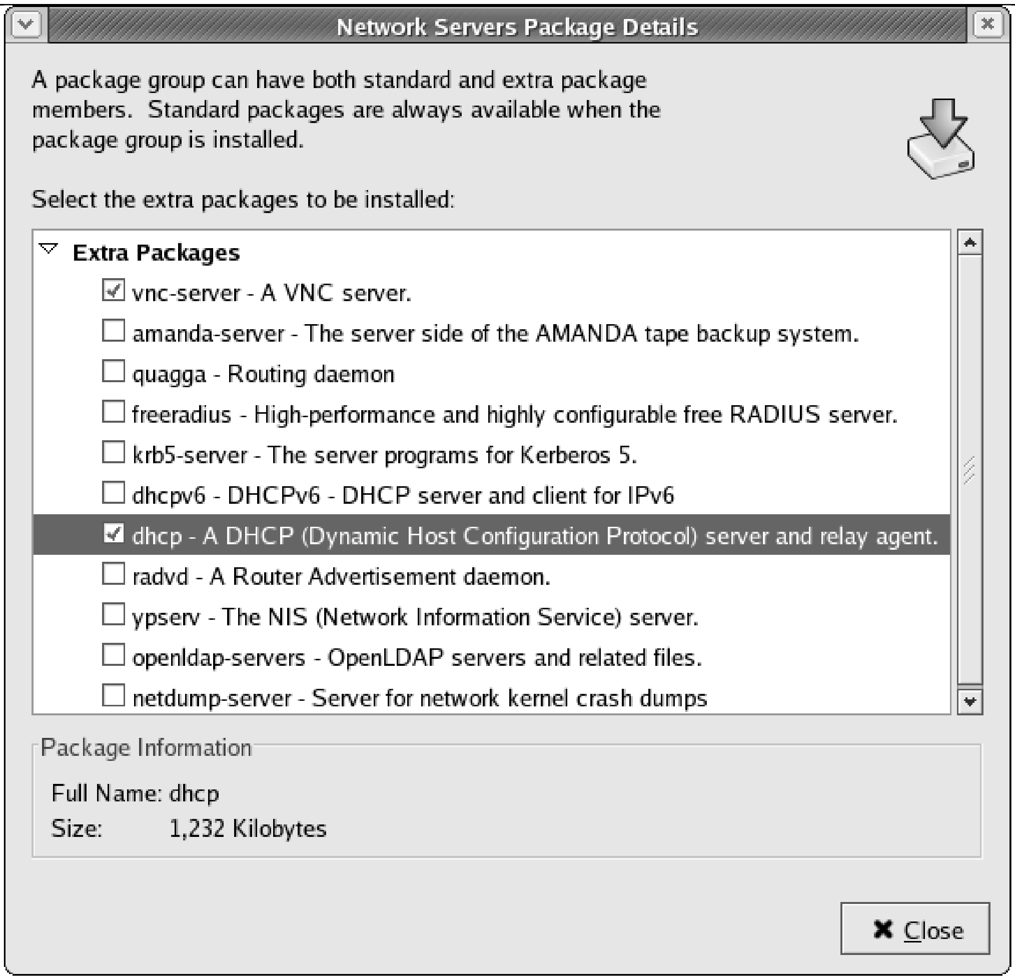
□ **Legacy Network Server** **[0/9]**

These packages include servers for old network protocols such as rsh and telnet.

Development

Total install size: 3,784 Megabytes

2. Scroll down the list until you see a listing for Network Servers.
3. Click the Details link for Network Servers.



A package group can have both standard and extra package members. Standard packages are always available when the package group is installed.

Select the extra packages to be installed:

Extra Packages

- ☑ vnc-server - A VNC server.
- □ amanda-server - The server side of the AMANDA tape backup system.
- □ quagga - Routing daemon
- □ freeradius - High-performance and highly configurable free RADIUS server.
- □ krb5-server - The server programs for Kerberos 5.
- □ dhcpv6 - DHCPv6 - DHCP server and client for IPv6
- ☑ dhcp - A DHCP (Dynamic Host Configuration Protocol) server and relay agent.
- □ radvd - A Router Advertisement daemon.
- □ ypserv - The NIS (Network Information Service) server.
- □ openldap-servers - OpenLDAP servers and related files.
- □ netdump-server - Server for network kernel crash dumps

Package Information

Full Name: dhcp
Size: 1,232 Kilobytes

4. Click Close; then click Update, and finally click Continue.
5. Insert the requested numbered installation CD when prompted and click OK.
6. After the package is installed, click Close to exit the Package Management tool.

In Fedora Core and Red Hat Enterprise Linux the DHCP server is controlled by the text file /etc/dhcpd.conf. Listing below shows the configuration file for my system. Comment lines begin with a # sign.

```
#(The amount of time in seconds that the host can keep the IP
address.)
default-lease-time 36000;
#(The maximum time the host can keep the IP address.)
#domain name
max-lease-time 100000;
# (The domain of the DHCP server.)
#nameserver
option domain-name “tactechnology.com”;
option domain-name-servers 192.168.1.1;
#gateway/routers, can pass more than one:
option routers 1.2.3.4,1.2.3.5;
option routers 192.168.1.1; (IP address of routers.)
#netmask (The subnet mask of the network.)
option subnet-mask 255.255.255.0;
#broadcast address (The broadcast address of the network.)
option broadcast-address 192.168.1.255;
#specify the subnet number gets assigned in
subnet 192.168.1.0 netmask 255.255.255.0
#define which addresses can be used/assigned
range 192.168.1.1 192.168.1.126;
```

Configuring the DHCP Client

First, you need to be sure that you NIC is properly configured and recognized by your system. After that, it is easy to tell your system to use DHCP to obtain its IP information. Follow these steps.

1. Using your favorite text editor, open the /etc/sysconfig/networkscripts/ ifcfg-eth0 file.
2. Find the line bootproto=static.
3. Change static to **dhcp**.
4. Save your changes.
5. Restart the network by issuing the command **service network restart**, and your system will receive its IP information from the DHCP server.

3a What is Samba? What is server message block? Explain

- Samba is an important component to seamlessly integrate Linux/Unix Servers and Desktops into Active Directory environments. It can function both domain controller or as a regular domain member.
- Using Samba, we can emulate the Windows file-sharing protocol and connect your Fedora Core and Red Hat Enterprise network to a Windows network.
- Computers running Windows 95 or greater use a protocol called Server Message Block (SMB) to communicate with each other and to share services such as file and print sharing to share files and printers.
- SMB works through a client-server approach, where a client makes specific requests and the server responds accordingly.
- One section of the SMB protocol specifically deals with access to filesystems, such that clients may make requests to a file server; but some other sections of the SMB protocol specialize in inter-process communication (IPC).
- The Inter-Process Communication (IPC) share, or ipc$, is a network share on computers running Microsoft Windows.
- This virtual share is used to facilitate communication between processes and computers over SMB, often to exchange data between computers that have been authenticated.

3b How are samba users created? Explain with examples.

Creating Samba Users:

1.Creating Samba users means, assigning a Linux user account to each person using the Linux file system and printers from windows. We also need to provide a SMB password for each user. To add or create a new Samba user, we need to perform the following commands:

a) Type the following command in the terminal window as a root user:

```
useradd –m bob
```

This will add a user with name “bob”.

b) Add a Linux password for the new user as follows:

```
passwd bob
```

The terminal will prompt you to change password for the user bob:

Changing password for user bob

```
New password: ******
Retype new password: ******
```

Repeat the above commands to add user accounts for all the users from windows workstation to give access to your Linux System.

c) Type the following command to create Samba password file (smbpasswd) on Fedora Linux system:

```
cat/etc/passwd | mksmbpasswd.sh >/etc/samba/smbpasswd
```

This command creates only the user’s account, not their passwords. You need to create password for your users by using the smbpasswd commands, written as follows:

```
smbpassword bob
New password: ******
Retype new password: ******
```

2.Starting the Samba Server

The last step is to start the Samba daemon. The command to start Samba is:

```
[root@terry terry]# /sbin/service smb start
Starting SMB services: [OK]
```

At this point, we have a functioning Samba server running on our system. It is configured to allow users who have accounts on your Red Hat Enterprise Linux system to access their home directories from a Windows PC. Logged-in users are also able to use the printers configures with the Red Hat system.

3c **What is timeserver? Why is it required? What are the different types of time servers? How is it configured?**

A time server is a daemon that runs on one machine and to which other systems synchronize their system clocks.

The motivation for a time server is to keep the system time consistent throughout the LAN(Local Area Network) so the time-sensitive operations work reliably.

The development environments source code version control systems often rely on file and directory timestamps to track changes to files maintained in the source code repository. NFS is also sensitive to timekeeping irregularities .If the system clock on client machine is significantly faster or slower than system clock on the NFS server, we run into problem saving files.

There are three categories of time server: Hardware, software, both.

Hardware solution involves installing a high resolution clock and then configures the client system to synchronize their system clocks against the dedicated device.

	The most common time server solution especially for small networks and organizations is software based. The simplest approach is to use the date program to set your system clock to the time broadcast by another system. Preferred method is to use Network Time Protocol or NTP. NTP is an open standard that defines how internet time server work and how clients can communicate with this time server to maintain accurate time. NTP consist of daemon (ntpd), a small set of utility programs (ntpq , ntpdc , ntpdate , ntptrace, tickadj, ntptime, ntp-kegen and ntpdsim) and the configuration file /etc/ntp.conf. The NTP daemon is dual purpose. 1. It act as a server listening for time synchronization request and provides the time in response and 2. As a client communicating with other time servers to get the correct time and adjust the local system accordingly. Configuring Time Server 1. Install the NTP software. 2. Local suitable time servers to serve as reference clocks. 3. Configuring local time server . 4. Start the NTP daemon on the local time server . 5. Make sure that the NTP daemon responds to request. 1. Install the NTP software • To make sure that the NTP packages is installed run following command #rpmquery ntp 2. Selecting reference clock • For your time server to keep and thus to serve accurate time, your local time server needs to synchronize its time against one or more master or reference clicks NTP is distributed application meaning that servers and clients are spread, that any given client can request a time check from any given server and application.
3d	What is a caching proxy? What are the uses of a caching proxy server?
	A cache server is also called a "cache engine." A cache server is almost a proxy server, which is a server that represents users by intercepting their Internet requests and managing them for users. Uses of Caching Proxy Server 1. Monitoring and Filtering:- a. A content-filtering web proxy server provides administrative control over the content that may be relayed in one or both directions through the proxy. It is commonly used in both commercial and non-commercial organizations (especially schools) to ensure that Internet usage conforms to acceptable use policy. b. Web filtering proxies are not able to peer inside secure sockets HTTP transactions, assuming the chain-of-trust of SSL/TLS has not been tampered with.

c. Proxies can be installed in order to eavesdrop upon the data-flow between client machines and the web. All content sent or accessed – including passwords submitted and cookies used – can be captured and analyzed by the proxy operator.
d. For this reason, passwords to online services (such as webmail and banking) should always be exchanged over a cryptographically secured connection, such as SSL.

2. Improving Performance:-
a. A caching proxy server accelerates service requests by retrieving content saved from a previous request made by the same client or even other clients.
b. A proxy that is designed to mitigate specific link related issues or degradations is a Performance Enhancing Proxy (PEPs).
c. These typically are used to improve TCP performance in the presence of high round-trip times or high packet loss (such as wireless or mobile phone networks)

3. Translation:-
a. A translation proxy is a proxy server that is used to localize a website experience for different markets.
b. Traffic from global audiences is routed through the translation proxy to the source website.
c. Original language content in the response is replaced by translated content as it passes back through the proxy. The translations used in a translation proxy can be either machine translation, human translation, or a combination of machine and human translation.

4. Accessing services anonymously:-
a. An anonymous proxy server(sometimes called a web proxy) generally attempts to anonymize web surfing. There are different varieties of anonymizers.
b. The destination server (the server that ultimately satisfies the web request) receives requests from the anonymizing proxy server, and thus does not receive information about the end user's address.
c. The requests are not anonymous to the anonymizing proxy server, however, and so a degree of trust is present between the proxy server and the user.
5. Security:-
a. A proxy can keep the internal network structure of a company secret by using network address translation, which can help the security of the internal network.]
b. This makes requests from machines and users on the local network anonymous. Proxies can also be combined with firewalls.
c. An incorrectly configured proxy can provide access to a network otherwise isolated from the Internet.

4a Discuss the http protocol and following daemons available in Linux: sshd, ftpd.

HTTP:

1. The Hypertext Transfer Protocol (HTTP) is the most widely used protocol of the TCP/IP suite.
2. HTTP is based in the TCP/IP Application Layer and is used to transfer Hypertext Markup Language (HTML) files, image files, query results and other communications.
3. HTTP can allow any media type as long as the client system has information to handle the data type properly.
4. The media types are controlled by the Multi-purpose Internet Mail Extensions (MIME) type list on a system. MIME information is in the /user/share/mime/ folder.

SSHD:

1. The SSHD stands for Secure Shell Daemon.
2. SSHD is the daemon program for SSH (Secure Shell).
3. It provides secure encrypted communications between two untrusted hosts over an insecure network.
4. It is started out of the system's rc scripts.
5. Its global system configuration files are in /etc/ssh and user's SSH configuration files are in $HOME/.ssh/.
6. The daemons handle key exchange, encryption, authentication, command execution, and data exchange.
7. The SSH server listens on port 22.

FTPD:

1. The FTPD (File Transfer Protocol Daemon) uses ports 20 and 21 to listen for and initiate FTP requests.
2. The configuration files ftpaccess, ftpconversions, ftpgroups, ftphosts and ftpusers, are located in the /etc directory.
3. File Transfer Protocol (FTP) allows file sharing across varying platforms and systems.
4. FTP is used to allow a file to be copied from one host to another over a TCP/IP network.
5. FTP guarantees delivery by using the Transport Control Protocol (TCP).
6. There are two connections required between the hosts.
    1. **Data Connection** – used to transfer the data between the two hosts (port 21).
    2. **Control Connection** – used to transfer control commands between the hosts (port 20).

4b **How is xinetd server configured? Explain.**

1. It is the extended Internet Daemon. It is an open-source daemon which runs on many Linux and Unix systems and manages Internet-based connectivity. It offers a more secure extension to or version of inetd, the Internet daemon.
2. xinetd is a replacement for inetd that adds more security and functionality.
3. xinetd starts at system boot time, and waits and listens for connections to come in on the ports to which they are assigned in their conf files.
4. When a request comes in, xinetd starts the appropriate server.
5. With inetd, only root can start a network service, and that restriction leads to a host of security problems. In xinetd anyone can start the network services.
6. xinetd supports encrypting plain-text services, enables access control on all services based on different criteria.
7. xinetd also takes the extra security step of killing servers that aren't in the configuration file and those that violate the configuration's access criteria.
8. Log capabilities have also been improved in xinetd.
9. For each service it runs, it can log the remote user and host address, the length of time the service has been running, and failed access control attempts.
10. Following are important configuration files for xinetd:
    a. /etc/xinetd.conf - The global xinetd configuration file.
    b. /etc/xinetd.d directory - The directory containing all service-specific files such as ftp.
11. We can view default configuration file with less or cat command:
    a. # less /etc/xinetd.conf or
    b. # cat /etc/xinetd.conf
12. Some defaults, and include /etc/xinetd.d/ are:

    a. instances = 60 :-
    Determines the number of servers that can be simultaneously active for a service. So 60 is the maximum number of requests xinetd can handle at once.

    b. log_type = SYSLOG authpriv:-
    Determines where the service log output is sent. You can send it to SYSLOG at the specified facility.

    c. log_on_success = HOST PID:-
    Force xinetd to log if the connection is successful. It will log HOST name and Process ID to /var/log/secure file.

    d. log_on_failure = HOST:-
    Force xinetd to log if there is a connection dropped or if the connection is not allowed to /var/log/secure file

    e. cps = 25 30:-
    Limits the rate of incoming connections. Takes two arguments. The first argument is the number of connections per second to handle.

    f. **includedir /etc/xinetd.d**:-
    Read other service specific configuration file this directory.

13. If you use a Telnet server on your system, you should be aware that by default telnetd does not start up with xinetd.

4c Explain the nsswitch.conf and resolv.conf files. What are their uses?

nsswitch.conf-

1) nsswitch means Name Server Switch.
2) When we type in a host name, our system uses its resources to resolve names into IP addresses. One of these files is /etc/nsswitch.conf, which contains a line telling the system where to look for host information. It points to domain name server used to resolve IP address.
3) A system administrator usually configures the operating system's name services using the file /etc/nsswitch.conf.
4) Using this file, we can tell Linux server which method it should use first to look up hostnames. For this we must edit the following line

```
hosts:       files nisplus dns
```

   The order of words files, nisplus and dns determines which method is checked first. Files refer to /etc/hosts files, nisplus refers to nisplus servers we may have on network and dns refers to DNS servers we have set up our machine to reference. We can also specify what action the system takes based on whether a method works or fails.
5) This file contains some other useful settings such as below 2 lines which specify whether the server should authenticate users off the local password or off the network's NIS plus service:

```
passwd: files nisplus

shadow: files nisplus
```

6) The /etc/nsswitch.conf file is used to configure which services are to be used to determine information such as hostnames, password files, and group files.
7) Each line in nsswitch.conf specifies how to search for a piece of information, such as a user's password. A line in nsswitch.conf has the following format:

```
info: method [[action]] [method [[action]]...]
```

   where info is the type of information that the line describes, method is the method used to find the information, and action is the response to the return status of the preceding method. The action is enclosed within square brackets

resolv.conf-resolver configuration file

1) resolv.conf is the name of a computer file used in various operating systems to configure the system's Domain Name System (DNS) resolver.

2) The file is a plain-text file usually created by the network administrator or by applications that manage the configuration tasks of the system.
3) The resolv.conf configuration file contains information that determines the operational parameters of the DNS resolver. The DNS resolver allows applications running in the operating system to translate human-friendly domain names into the numeric IP addresses that are required for access to resources on the local area network or the Internet. The process of determining IP addresses from domain names is called resolving.
4) After the system looks in hosts file and fails to find the address, the next file checked is /etc/resolv.conf.
5) This file contains IP address of computers that are known as Domain Name Serves and these are listed in /etc/resolv.conf as just name servers. To set up name servers for reference we need to make changes to this file.
6) The program that resolves host names to IP address reads a file called resolv.conf, so we need to put your DNS server IP address here. We can specify up to three nameserver. Specifying more than one is important if the first one is not responding.
7) Edit /etc/resolv.conf to contain a list of nameservers like:

```
nameserver 172.16.1.254

nameserver 172.16.2.254
```

8) resolv.conf is usually located in the /etc directory of the file system.
9) The file is either maintained manually, or when DHCP is used, it is usually updated with the utility resolv.conf.

4d **Explain the seven configuration statements used in named.ca file.**

The seven configuration statements used in named.ca file are as follows:

1. options
2. include
3. acl
4. logging
5. server
6. zone
7. key

1. option:

a. The options statement is typically the first section of named.conf, and it contains information about the location of the files used by named.
b. We can use only one options statement, but can have more than one value for that statement.
c. The options statement shows the path to the location where additional configuration files used by named are located.

    d. By specifying the directory where other files are located, it unnecessary to list the entire path to the file

2. **include:**
    a. The include statement lists the path and name of any files that you want to be included with the named.conf file.
    b. Use the same syntax used in the options statement to specify the path.
3. **acl**
    a. This option lets you specify a list of IPaddresses in an access control list.
    b. Only hosts on this list have access to the server.

4. **logging**
    a. The logging statement is where we specify your server’s logging options.
    b. The logging statement contains two additional items, the channel and the category.
    c. The channel is where we specify the location of the logged information.
    d. Logged information can be written to a file, sent to the syslog, or thrown away by specifying the appropriate command.
    e. Choosing to send the information to a file gives us several additional choices for handling the information.
    f. We can set the number of versions to keep, the size of the files, and whether the severity of the information and the time and category are included with the other information in the file.
    g. The syntax for the logging statement is similar to the syntax for the option statement.

5. **server**
    a. In the server statement we can set the properties of a remote server.
    b. We can specify whether to send queries to the remote server from the local server, and we can set the method used for transferring information.
    c. The syntax for this statement is the same as for other statements.
    d. The following are valid values:
        i. bogus — Specify yes or no. No is the default and indicates that queries are sent to the remote server. Yes means that the remote server is not queried.
        ii. transfer — Specify the number of transfers you want to allow.
        iii. transfer-format — Specify whether you want one-answer or manyanswers
        iv. keys — Specify key ID (currently not implemented).

6. **zone**
    a. These zone statements refer to files that are called zone files.
    b. Additional options for zone statements exist, of course.
    c. Each zone statement begins with the word zone followed by the domain name and the data class.
    d. The four data classes are in, hs, hesiod, and chaos.
    e. If no type is specified, the default is in, for Internet.

7. **key**
    a. Specifies security keys used for authentication

5a **What is POP3? What is IMAP4? Explain and compare them.**

Ans

<u>POP3 →</u> <u>Post Office Protocol</u>

1. It is a way of retrieving email information that dates back to a very different Internet than we use today.

2. Computers only had limited, low bandwidth access to remote computers, so engineers created POP in an effort to create a dead simple way to download copies of emails for offline reading, then remove those mails from the remote server.

3. Since POP3 creates local copies of emails and deletes the originals from the server, the emails are tied to that specific machine, and cannot be accessed via any webmail or any separate client on other computers. At least, not without doing a lot of email forwarding or porting around mailbox files.

4. While POP3 is based on an older model of offline email, there's no reason to call it obsolete technology, as it does have its uses.

<u>IMAP4</u> → <u>Internet Message Access Protocol Version 4</u>

1. The Internet Message Access Protocol version 4 (**IMAP4**) provides much more sophisticated email-handling functionality than SMTP or POP3 do.
2. IMAP4 has more features.
3. IMAP4 enables you to store email on a networked mail server, just as POP3 does.

<u>Comparing POP3 and IMAP4:-</u>

Sr.no.	POP3	IMAP4

1.	POP3 downloads email from a server to a single computer, then deletes it from the server.	In IMAP all messages from mail clients and servers are synced with each other.
2.	Once mail is deleted from server, if we try to check our mail from a different computer, it can appear that mail is missing or disappeared from our Inbox.	IMAP allows users to log into many different email clients or webmail interfaces and view the same emails.
3.	You can use only one computer to check your email (no other devices).	You can use multiple computers and devices to check your email.
4.	Messages may be reloaded into PC several times due to the corruption of system files.	The occurrence of reloading messages from the server to PC is much less when compared to POP3.
5.	Our mails are stored on the computer that we use.	Our mails are stored on the server.

5b **How is vsftpd run over SSL? What are the SSL-related configuration directives for vsftpd over SSL? Explain.**

Vsftpd’s Secure Sockets Layer (SSL) support passing authentication information in clear text.

In fact, vsftpd can use SSL to encrypt FTP’s control channel, over which authentication information is passed, and FTP’s data channel, over which file transfers occur. To use SSL, with vsftpd, you need to set at least the ssl_enable=YES in /etc/vsftpd/vsftpd.conf. If you want to fine-tune vsftpd’s SSL-related behavior, become familiar with vsftpd’s SSL-related configuration directives, listed below.

1. Add the following entries to /etc/vsftpd/vsftpd.conf:

```
ssl_enable=YES
allow_anon_ssl=YES
force_local_data_ssl=YES
```

```
force_local_logins_ssl=YES
ssl_tlsv1=YES
```

2. Create a self-signed RSA certificate file:

```
# cd /usr/share/ssl/certs
# make vsftpd.pem
umask 77 ; \
PEM1=`/bin/mktemp /tmp/openssl.XXXXXX` ; \
PEM2=`/bin/mktemp /tmp/openssl.XXXXXX` ; \
/usr/bin/openssl req -newkey rsa:1024 -keyout $PEM1 -nodes -x509 -
days 365 -out
$PEM2 ; \
cat $PEM1 > vsftpd.pem ; \
echo “” >> vsftpd.pem ; \
cat $PEM2 >> vsftpd.pem ; \
rm -f $PEM1 $PEM2
Generating a 1024 bit RSA private key
.........................++++++
...++++++
writing new private key to ‘/tmp/openssl.yYS512’
-----
You are about to be asked to enter information that will be
incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or
a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter ‘.’, the field will be left blank.
-----
Country Name (2 letter code) [GB]:US
State or Province Name (full name) [Berkshire]:Allegheny
Locality Name (eg, city) [Newbury]:Pittsburgh
Organization Name (eg, company) [My Company Ltd]:KurtWerks
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server’s hostname) []:localhost
Email Address []:root@localhost
```

3. Start or restart the vsftpd service:

```
# service vsftpd start
```

The trick here is to test the connection using an FTP client that supports SSL. Not many do, certainly not the command line ftp program that is part of Fedora Core and RHEL.

However, the lftp program does support SSL, so you can use it to verify that your SSL configuration is correct.

5c **How can anonymous ftp access be disabled in vsftpd? Explain.**

The easiest way is to remove the ftp user from /etc /passwd and /etc/group:

```
# cp -p /etc/passwd /etc/passwd.ftp
# cp -p /etc/group /etc/group.ftp
# userdel -r ftp
userdel: /var/ftp not owned by ftp, not removing
# find / -user 50 | xargs rm -r
```

Ordinarily, userdel's -r option removes files in ftp's home directory (/var/ftp), but it doesn't work in this case because the ftp user doesn't own /var/ftp, root does. userdel also removes the ftp user from /etc/group, so you needn't execute the groupdel command. The find command locates all the files owned by the ftp user and deletes them. You have to use the numeric UID (50) instead of the username (ftp) because the username no longer exists. You might not want to execute the command if you have populated the FTP server with files that you can't easily replace.

The problem with this method is that if you later decide to permit anonymous FTP, you have to recreate the ftp user and group because, as configured, vsftpd doesn't allow *any* FTP login if the user ftp is not present in the password file. That's why we made backup copies of /etc/passwd and /etc/group before executing userdel.

A more flexible approach is to add ftp to /etc/vsftpd/user_list and set userlist_deny=YES and anonymous_enable=NO in /etc/vsftpd /vsftpd.conf. It is *not* sufficient to comment out anonymous_enable=YES, because that will default to permitting anonymous FTP. This approach disables anonymous FTP while permitting regular FTP. However, if you use this method, remove any other users from /etc/vsftpd/user_list that you do want to be able to log in via FTP.

5d **Explain the configuration process of Postfix mail server?**

Postfix's primary configuration file is /etc/postfix/main.conf. We can edit the following variables :

- The my domain variable specifies your domain name :-
  mydomain = example.com
- The myhostname variable identifies the local machine's fully qualified domain name :-
  myhostname = computerName.example.com

- The myorigin variable identifies the domain name appended to unqualified addresses that is username without the @example.com attached.

```
myorigin =$mydomain
```

  This causes all mail going out to have your domain name appended. Thus, if the value of mydomain is ***Mavericksky.com*** and your username is ***vagabond***, then your outgoing mail will appear to come from ***vagabond@Mavericksky.com.***
- The mydestination variable tells Postfix what addresses it should deliver locally .For a standalone workstation which is a system that is connected directly to the internet and that has some sort of domain name resolution running, you want mail to that machine and to localhost(and/or ***localhost.$mydomain*** and /or ***localhost,localhost.localdomain***)delivered locally:

```
mydestination=$myhostname,localhost,localhost.$mydomain
```

- Create or modify /etc/aliases. You need aliases for Postfix,postmaster,and root in order for mail sent to those addresses to get to a real person. Here are the contents of my initial/etc/aliases file :-

```
Postfix: root
Postmaster: root
Root: nilofer
```

- After creating or modifying the aliases file , regenerate the aliases database using Postfix's newaliases command:-

```
#/user/sbin/newaliases
```

- Now you are finally ready to start Postfix by following command:-

```
# service postfix start
```

- Finally ,modify your syslog configuration to handle Postfix log messages approximately use the following entries in ***/etc/syslog.conf*** , which controls the system log :-

```
*.info;*.!warn;authpriv.none;cron.none;mail.none; -/var/log/messages
*.warn;authpriv.none;cron.none;mail.none;          -/var/log/syslog
 mail.*;mail.!err                                  -/var/log/mail.log
mail.err                                           -/var/log/mail.err
```

- The first two lines keep any mail-related messages from being logged to *<u>/var/log/messages</u>* and *<u>/var/log/syslog</u>* .
- The 3rd line logs everything but errors to */var/log/mail.org* .
- The last line drops all error messages from Postfix into /var/log/mail.err.
- Restart syslogd syslogd to cause these changes to take effect:-

```
# service syslog restart .
```

6a **What are the changes in Apache 2.0 compared to Apache 1.x? Explain.**

When Apache 2 was released, it included some fundamental architectural changes. One of the most far-reaching changes made in the Apache 2 affected how httpd servers are created. Apache 1.x followed a standard master/child server creation model. In the 1.x model, single master server process spawned a number of child server processes to perform the actual work of serving Web pages. As HTTP traffic increased, the master server would spawn additional child server processes to meet the demand, up to a limit set by the MaxClients directive. As HTTP traffic decreased, the master server would kill some of the child server processes so that the number of idle child httpd server processes stayed between the values set by MinSpareServers and MaxSpareServers directives.

While this model worked well, it imposed significant overhead and limited the scalability of Web servers. The overhead resulted from the time involved in and system resources consumed by creating, maintaining, and killing large numbers of server processes. The scalability issue resulted from having to recompile Apache to override the hard-coded limit of 256 clients (for the MaxClients directive).

Apache 2 redesigned the server generation process, replacing the monotonic process model with multiprocessing modules, or MPMs. Each MPM implements a different server control and process generation method, giving system administrators greater flexibility in deciding how Apache will create and destroy child server processes. Some MPMs exist only to take advantage of the underlying operating system's particularities. Examples of OS-specific MPMs include beos, mpm_netware, mpmt_os2, and mpm_winnt, for BeOS, Netware, OS/2, and Windows NT, respectively. You cannot use these MPMs on Linux systems, so we have nothing more to say about them. The other MPMs implement various approaches to process creation and control. These MPMs include:

■■ **perchild** — Provides a hybrid server that is both a multiprocess and multithreaded server, meaning that the Apache master server starts multiple child server processes, and each child server process can spawn a number of server threads. The advantage of this model is that threads are much faster and consume very few resources compared to full-fledged processes. An additional feature makes it possible to assign different user IDs to the child processes.

■■ **prefork** — Provides an unthreaded, process-based server that resembles the server creation model from Apache 1.3*x* in which the master server maintains a pool of idle spare servers that are started before they are needed (that is, are preforked).

■■ **worker** — Provides a hybrid server that is both multiprocess and multithreaded. Like the perchild MPM, the worker MPM starts multiple child server processes, and each child server process can spawn a number of server threads. The difference between the worker and perchild MPMs is that the worker MPM cannot assign child processes to multiple user IDs.

Other notable changes in Apache 2 include the following:

■■ The build system was rewritten to use the standard GNU build tools, namely, autoconf and libtool.

■■ A new API for modules removes most requirements for loading modules in a certain order.

■■ Apache 2 supports IPv6 on those systems that have IPv6. Rather, if the Apache runtime supports IPv6 on a given OS, Apache 2 listens on IPv6 ports for HTTP connections.

■■ Apache's regular expression support has been replaced using Perl-Compatible Regular Expressions (PCRE).

■■ HTTP error messages generated by the server can be displayed in multiple languages using server-side includes (SSI).

■■ Filters to modify the httpd input or output stream can be implemented as modules, significantly improving the performance of those filters.

■■ The configuration language has been simplified and rationalized. For example, confusing directives such as Port and BindAddress

6b **Explain the userdel and usermod command with options and arguments.**

1) The userdel command deletes a user account and, optionally, related files.
2) Its syntax is: userdel [-r] *username*
3) username identifies the user account to delete.
4) Using -r deletes the corresponding home directory and mail spool. Without -r, userdel removes only the account references in the user and group database files.
5) We cannot delete the account of a logged in user, so userdel fails if username is logged in.
6) The usermod command modifies an existing user account. Its syntax is:
usermod [-c *comment*] [-d *dir* [-m]] [-e *date*]
[-f *inactive*] [-g *group*] [-G *group*[,...]]
[-l *new_username*] [-p *passwd*]
[-s *shell*] [-u *uid* [-o]] [-L|-U] *username*
7) usermod accepts the options and arguments.
8) It adds three new ones,
9) -l new_username, -L and -U.
10) -l new_username changes the account name from username to new_username.
11) -L disables (locks) username's account by placing a ! in front of the user's encrypted password in /etc/shadow.
12) -U enables (unlocks) the account by removing the !.
13) At least one option must be specified, but -p, -U, and -L may not be used together
14) in any combination.
15) If username is logged in, usermod fails because we cannot change the login name of a logged-in user.

6c **How can we build packages from source rpms? Explain the various valid build stages.**

1. In the simplest case, building and installing software from SRPMs requires one or possibly two commands.
2. The same unpack/configure/build/install procedure described in the previous section takes place, but RPM handles each of these steps for we.
3. In this section, we will learn how to use the two command cases (building and installing an RPM), and how to invoke each step of the RPM build process.
4. The general form of the command to build a binary RPM from a source RPM is: rpmbuild -b[*stage*] *spec_file* [...]

- a- All Builds both binary and source RPMs
- b-Binary Builds only a binary RPM
- c- Compile Compiles the source code
- i-Install Installs the files
- l-List Makes sure that all the package files exist
- p-Prep Unpacks the source code and applies any patches
- s-Source Builds only a source RPM

5. Stages are executed in the order listed, and later stages require preceding ones, with one exception. That is, the l (list) step, for example, cannot be performed before the p (prep) stage, and the b (binary) stage happens after the p, l, c, and i (prep, list, compile, and install) stages have been completed.
6. The exception is that building a source RPM (the s stage) does not require first building a binary RPM.
7. Note that the install stage of the RPM build process does not mean that files are moved into the working file system.
8. Rather, files are “installed” in their proper paths underneath RPM’s build directory.
9. For example, if RPM’s build directory is /var/tmp/myrpm, the files /usr/bin/foo and /usr/man/man1 /foo.1 would be installed underneath /var/tmp/myrpm, so their complete
10. paths would be /var/tmp/myrpm/usr/bin/foo and /var/tmp/myrpm /usr/man/man1/foo.1.
11. This step is necessary because of the way binary RPMs are built and how RPM installs them.
12. The following two commands illustrate building the util-linux-2.12a-16 binary RPM from its corresponding SRPM (and assume that the SRPM is already installed using the instructions in the “Installing RPMs” section earlier in the chapter).

```
# cd /usr/src/redhat/SPECS
# rpmbuild -bb util-linux.spec
```

13. The build process generates quite a bit of output, most of which was deleted in the output listing. The SPECS directory contains the spec (presumably, short for specification) files that control RPM’s build process.
14. The rpm command uses -bb to build a binary RPM using the instructions in the utillinux spec file.
15. As the initial few lines of output shows, RPM first decompresses the archive file, using gzip, and unpacks the archived files using tar.

16. Additional steps apply any necessary patches, configure the package as necessary, invoke the build process, and then “install” the files as explained previously.
17. The following two lines appear near the end of the process:
    ```
    Wrote: /usr/src/redhat/RPMS/i386/util-linux-2.12a-16.i386.rpm
    Wrote:/usr/src/redhat/RPMS/i386/util-linux-debuginfo-2.12a-16.i386.rpm
    ```
18. They indicate that RPM created two binary RPMs, util-linux-2.12a- 16.i386.rpm and util-linux-debuginfo-2.12a-16.i386.rpm, in the /usr/src/redhat/RPMS/i386 directory**.**
19. Once the packages are built, we can install them as we would any other binary RPM, as the following command illustrates:
    ```
    # cd ../RPMS/i386
    ```
20. We can also build an RPM in stages by specifying one of the earlier stages.
21. For example, the following command executes only the prep (p) stage against the mount SRPM:
    ```
    # rpmbuild -bp util-linux.spec
    ```
22. The next command stops the process after the compile (c) stage:
    ```
    # rpmbuild -bc util-linux.spec
    ```
23. Again, using the build stages in this manner is not something end users usually need to do, but the capability is there.
24. The reason we might want to build in stages is to monitor each step of the build process to track down problems.
25. The results of one incomplete build invocation overwrite the results of a previous one.
26. Thus, if we execute rpmbuild -bp foo.spec, somehow change the unpacked files, and then execute another rpmbuild –bp foo.spec, we will lose wer changes.

6d **What are the advantages of shadow passwords over traditional password system?**

1. Red Hat Linux, like most, if not all, Linux and UNIX systems, uses shadow passwords because they offer enhanced protection for wer system’s authentication files.
2. During the installation of Red Hat, shadow password protection for wer system is enabled by default, as are MD5 passwords, which are an alternative and far more secure method of encrypting passwords because they are longer and use a encryption method stronger than the standard DES encryption used by the standard authentication utilities.
3. Shadow passwords offer a number of distinct advantages over the traditional password system, including:
    a. Improved system security by moving the encrypted passwords (normally found in /etc/passwd) to /etc/shadow, which is readable only by root

    b. Information concerning *password aging*, how long it has been since a password was last changed

    c. Control over how long a password can remain unchanged before the user is required to change it
4. Settings in /etc/login.defs, particularly concerning passwords to give

we the ability to set and enforce a security policy

5. The shadow password suite contains a number of utilities that simplify working with shadow passwords and, if we wish, also simplify reverting to traditional password management.
6. These utilities include:
   i) pwconv and pwunconv for switching from normal to shadow passwords and back, respectively.
   ii) pwck and grpck for verifying the contents and consistency of the password and group files, respectively, against their shadowed counterparts useradd, usermod, and userdel, for adding, deleting, and modifying user accounts groupadd, groupmod, and groupdel, for adding, deleting, and modifying group accounts gpasswd, for administering the groups file /etc/group.

Q7a **Explain RAID levels 0,1,5,6 and 10**

As a business owner, you have many features to consider when choosing the right system and infrastructure for your critical online applications. One of the features you have to consider when choosing the right server for your business is whether to enable RAID on your system, but more importantly, what type of RAID to choose to fit your technical needs. Below we will go through all the pros and cons of each RAID level and give suggestions on which type to choose for your set up.

RAID, short for redundant array of independent (originally inexpensive) disks is a disk subsystem that stores your data across multiple disks to either increase the performance or provide fault tolerance to your system (some levels provide both).

There are two ways of implementing the system. Software raid and hardware raid.

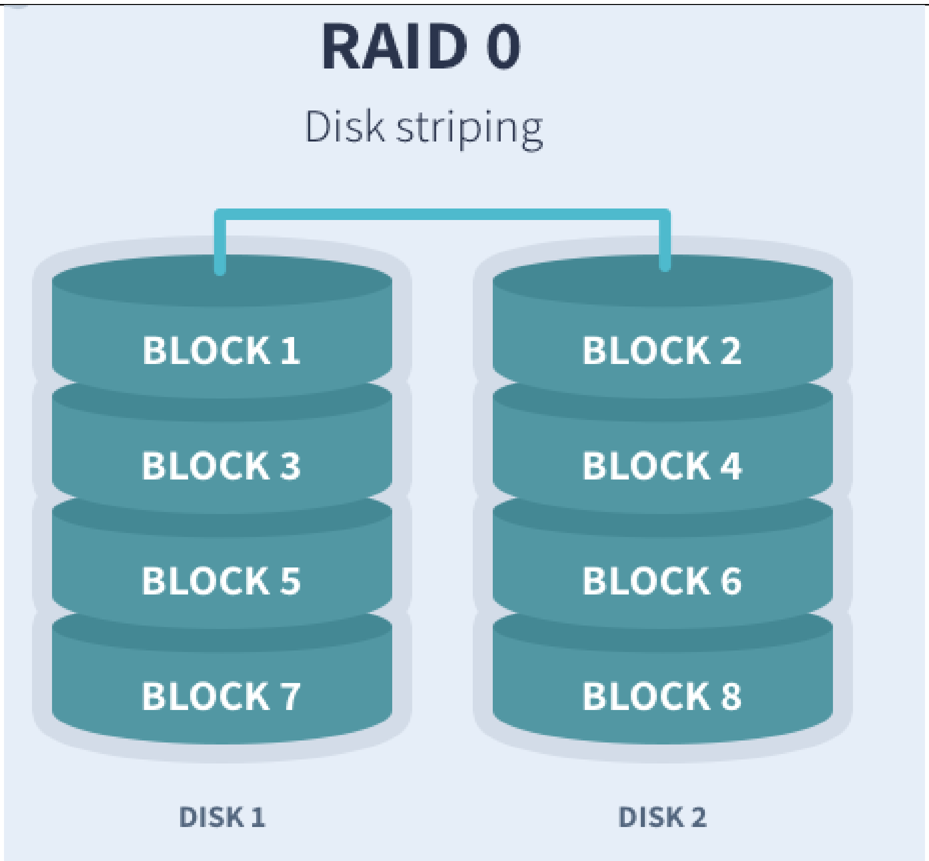
Hardware raid is directly managed by a dedicated hardware controller to which the disks are connected. The raid calculations are managed by an on-board processor which offloads the strain on the host processor CPU. However, the performance of today's CPUs has increased so much, that this advantage has become more or less obsolete. HW controllers do provide an extra failsafe element with its BBU (Battery Backup Unit) that protects your data in case of an unexpected power loss to the server.

Software RAID is part of the OS and is the easiest and most cost effective implementation. It does not require the use of an additional (often costly) piece of hardware and the proprietary firmware.

Here is a list of the most used RAID levels:

RAID 0 (Disk striping):

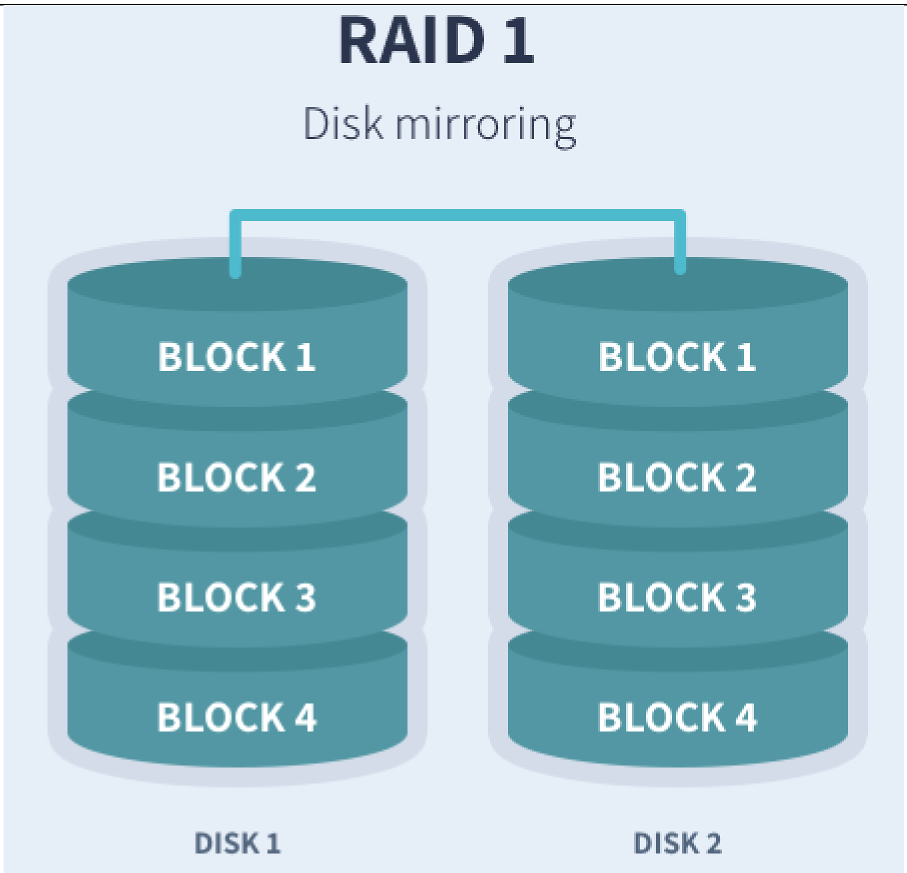
RAID 0 splits data across any number of disks allowing higher data throughput. An individual file is read from multiple disks giving it access to the speed and capacity of all of them. This RAID level is often referred to as striping and has the benefit of increased performance. However, it does not facilitate any kind of redundancy and fault tolerance as it does not duplicate data or store any parity information (more on parity later). Both disks appear as a single partition, so when one of them fails, it breaks the array and results in data loss. RAID 0 is usually implemented for caching live streams and other files where speed is important and reliability/data loss is secondary.



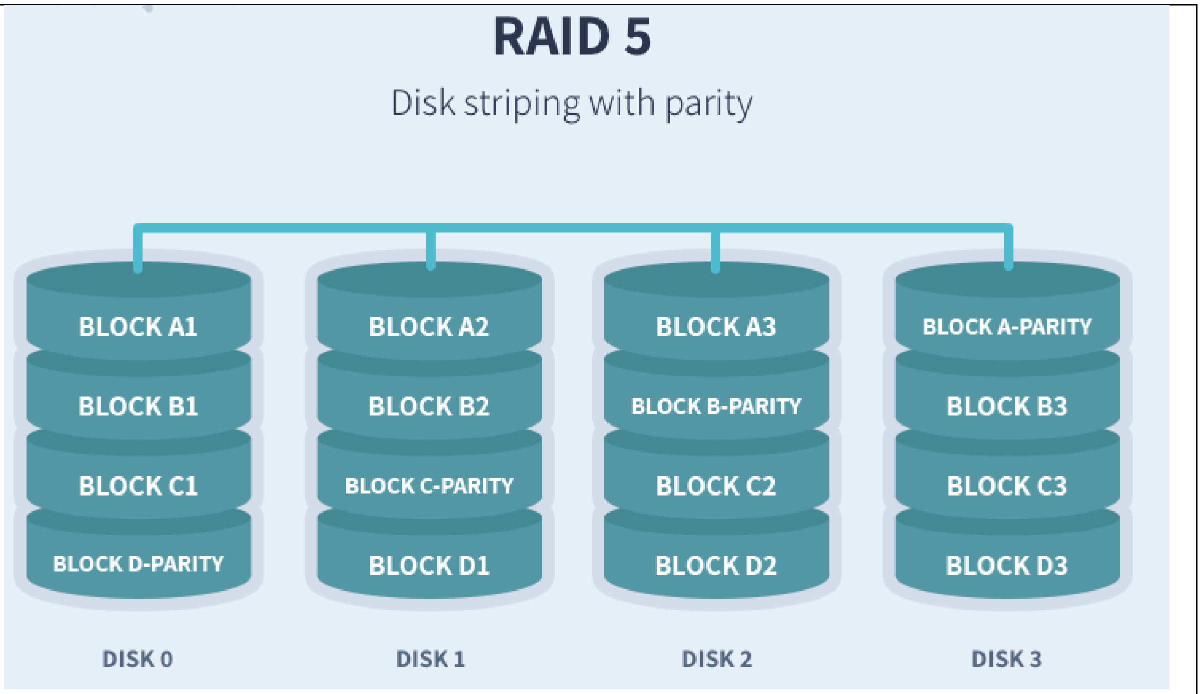
Minimum number of disks: 2
Pros: Increased performance (Write and read speeds).
Cons: No redundancy.
Business use: Live streaming, IPTV, VOD Edge Server

RAID 1 (Disk Mirroring):

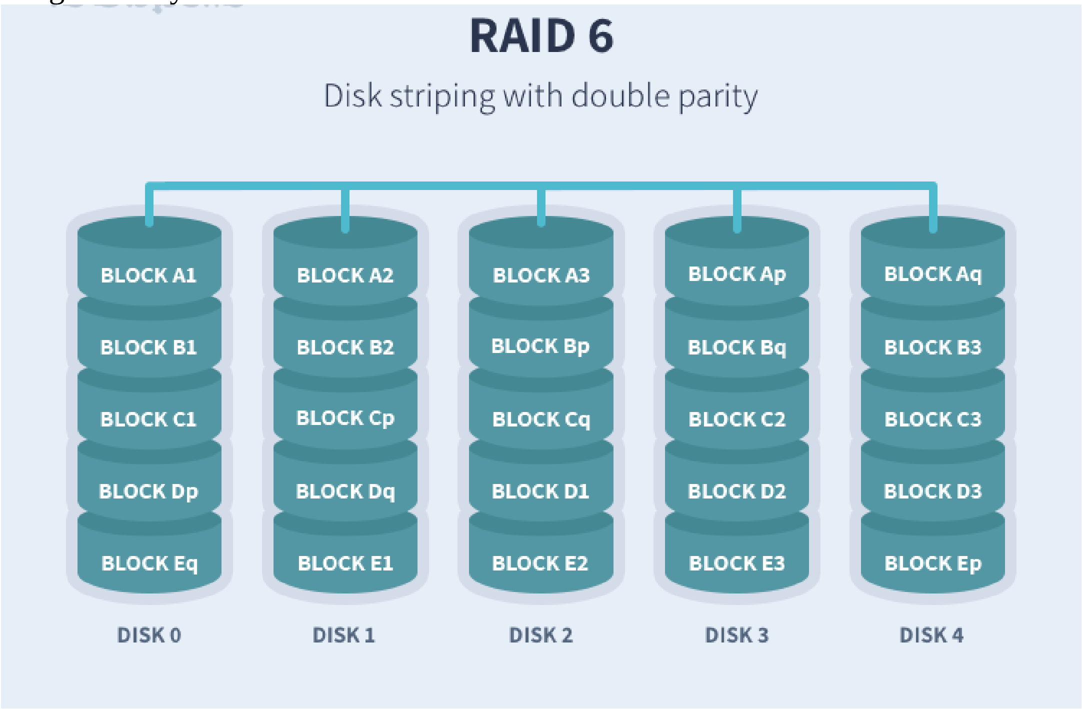
RAID 1 writes and reads identical data to pairs of drives. This process is often called data mirroring and it’s primary function is to provide redundancy. If any of the disks in the array fails, the system can still access data from the remaining disk(s). Once you replace the faulty disk with a new one, the data is copied to it from the functioning disk(s) to rebuild the array. RAID 1 is the easiest way to create failover storage.



Minimum number of disks: 2
Pros: Fault tolerance and easy data recovery. Increased read performance.
Cons: Lower usable capacity. Higher cost per megabyte (double the amounts of drives is required to achieve desired capacity).
Business use: Standard application servers where data redundancy and availability is important.
See more details on our dedicated server hardware.
RAID 5 (Striping with parity):
RAID 5 stripes data blocks across multiple disks like RAID 0, however, it also stores parity information (Small amount of data that can accurately describe larger amounts of data) which is used to recover the data in case of disk failure. This level offers both speed (data is accessed from multiple disks) and redundancy as parity data is stored across all of the disks. If any of the disks in the array fails, data is recreated from the remaining distributed data and parity blocks. It uses approximately one-third of the available disk capacity for storing parity information.



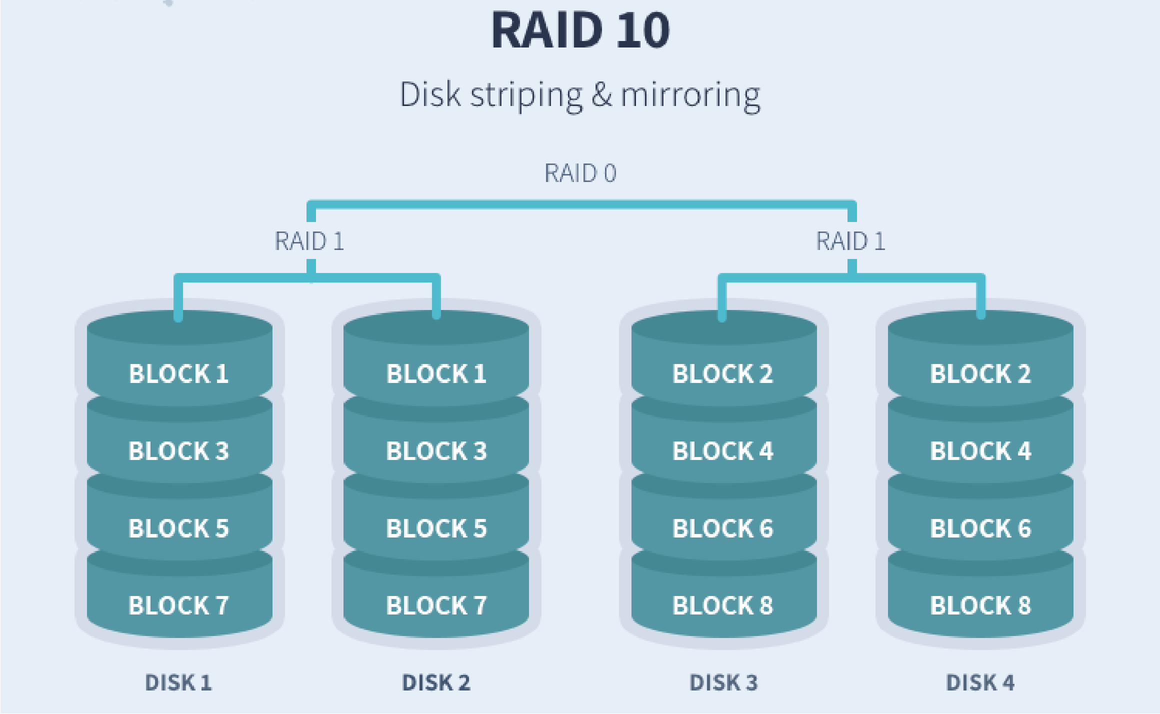
Minimum number of disks: 3
Pros: Fault tolerance and increased performance (lower than RAID 0)
Cons: Lower performance with servers performing large amounts of write operations because of parity overhead.
Ideal use: File storage servers and application servers.
RAID 6 (Striping with double parity):
Raid 6 is similar to RAID 5, however, it provides increased reliability as it stores an extra parity block. That effectively means that it is possible for two drives to fail at once without breaking the array.



Minimum number of disks: 4
Pros: Even higher redundancy than RAID 5. Increased read performance.
Cons: Lower performance with servers performing large amounts of write operations because of parity overhead.
Ideal use: Large file storage servers and application servers.

RAID 10 (Striping + Mirroring):

RAID 10 combines the mirroring of RAID 1 with the striping of RAID 0. Or in other words, it combines the redundancy of RAID 1 with the increased performance of RAID 0. It is best suitable for environments where both high performance and security is required.



Minimum number of disks: 4
Pros: Very high performance. Fault tolerance.
Cons: Lower usable capacity/High cost. Limited scalability
Ideal use: Highly utilized database servers/ servers performing a lot of write operations.

7b **Explain the showmount command with options and example.**

1. Showmount queries the mount daemon on a remote host for information about the state of the NFS server on that machine.
2. With no options showmount lists the set of the client who are Mounting from that host.
3. The output from showmount is designed to appear as though it were processed through "sort -u".
4. Syntax:
   showmount [-adehv] [--all]
   [--directories] [--exports] [--help]
   [--version] [host]

OPTIONS

Tag	Description
-a or **--all**	
	List both the client hostname or IP address and mounted directory in host:dir format. This info should not be considered reliable.
-d or **--directories**	
	List only the directories mounted by some client.
-e or **--exports**	
	Show the NFS server’s export list.
-h or **--help**	
	Provide a short help summary.
-v or **--version**	
	Report the current version number of the program.
--no-headers	
	Suppress the descriptive headings from the output.

The Completeness and accuracy of the information that showmount displays various according to the NFS Severs implementation

Examples:-

a) $ Showmount -e naso1 or showmount –e 192.168.1.12

```
        Exports list on naso1 :
        /Web
        /usb
        /Recordings
        /Public
        /Network  Recycle  Bin 1
        /Mutimedia
        /Mp3
        /Downloads
```

b) All mount points on mynfsserv01:

```
#showmount -a mynfsserv01
10.6.55.33:/umdata
10.6.55.34:/umdata
10.6.55.69:/umdata
```

```
10.6.55.3:/umdata
10.6.55.6:/umdata
10.6.55.16:/umdata
```

Where,
-a: Lists both the client hostname or IP address and mounted directory in host:dir format. This info should not be considered reliable.

c) The following example shows the output from the showmount command with no option specified. To only display the hostnames of all remote mounts from the hostname mvshost.

```
# showmount mvshost
mvshost.sanjose.ibm.com
usera.sanjose.ibm.com
```

7c Explain in detail the configuration process of squid.

To start squid configuration, first check squid is installed
#rpmquery squid
If squid is not installed then install it, the configuration process includes following steps

1. Verifying the kernel configuration
2. Configuring squid
3. Modifying the net filter configuration
4. Starting squid
5. Testing the configuration

1.)Verifying the kernel configuration

The most important kernel feature is net filter support as net filter handles the actual proxing of browser request.
Specifically, enable net filter and modulus that support

- Connection tracking
- IP tables
- Full NAT(Network Address Translator)
- Support for REDIRECT target

Enable IP Forwarding:

- IP Forwarding enables the kernel to send, or forward, packets that arrive on one network interface to another.
- sysctl command will show if IP forwarding is enabled:
  ```
  #sysctl-n net.ipv4.ip_forward
  ```
- If the displayed value is one, IP forwarding is enabled.
- If the output is zero, IP forwarding is not enabled , enable it using the following command:
  ```
  #sysctl –w net.ipv4.ip_forward=1
  net.ipv4.ip_forward=1
  ```

- To enable IP forwarding edit the start Squid file at boot time, edit the file etc/sysctl.conf ,
  Edit the line :.

  ```
  net.ipv4.ip_forward=0
  ```

  so that it reads

  ```
  net.ipv4.ip_forward=1
  ```

2.)Configuring Squid

The Squid Configuration file on fedora core and RHEL systems is etc/squid/squid.conf . The Initialization script that controls Squid is /etc/rc.d/init.d/squid , which reads default values from /etc/sysconfig/squid.

PARAMETER	DEFAULT VALUE	DESCRIPTION
cache_effective_group	Squid	Identifies the group Squid runs as
cache_effective_user	Squid	Identifies the user Squid runs as
httpd_accel_host	None	Defines the hostname of the real HTTP server(if using acceleration)
httpd_accel_with_proxy	Off	Controls whether Squid runs as both an accelerator and a proxy
httpd_accel_port	80	Defines the port number of the real HTTP server(if using acceleration)
httpd_accel_uses_host_ header	Off	Enables Squid to function as a transparent proxy
http_access	Deny all	Defines who can access the Squid server

- cache_effetive_user and cache_effective_group identifies User ID (UID) and Group ID (GID) respectively under which Squid runs.
- http_accel_with_proxy which defaults to off, controls whether Squid runs as cache(or accelerator) and proxy or just as proxy.
  When set to off , Squid function only as proxy.
  If set to on, Squid works as both as a cache and a proxy .
- http_accel_port which defaults to 80 to use Squid's caching functionality .
- http_accel_host to define the name of the host running Squid.
- For Transparent proxy server, set httpd _accel_uses_host_header on. The default value off means that the client has to configure their web clients to use a proxy server.
- httpd_access controls who can access the Squid server and therefore who can surf the web through the proxy. The default configuration is deny all which prevents any user from accessing the proxy.

```
cache_effective_user Squid
cache_effective _group Squid
httpd_accel_host squid.example.com
httpd_accel_with_proxy on
httpd_accel_port 80
httpd_accel_uses_host_header on
httpd_access allow all
```

- Replace squid.example.com with the name of the system on which you are running Squid.
- Initialize Squid’s cache using #squid -z

7d **What are zone statements? Explain the different values for the zone statement.**

Zone Statements:

a. These zone statements refer to files that are called zone files.
b. Additional options for zone statements exist, of course.
c. Each zone statement begins with the word zone followed by the domain name and the data class.
d. The four data classes are in, hs, hesiod, and chaos.
e. If no type is specified, the default is in, for Internet

<u>The different values for zone statements are:</u>

1. **allow-query** — Accepts queries only from hosts in the address list (by default queries are accepted from any host).

2. **allow-transfer** — Zone transfers are accepted only by hosts in the address list (by default transfers
are allowed to all hosts).

3. **allow-update** — Hosts in the address list are allowed to update the database.

4. **also-notify** — Servers in the address list are sent a notify message when the zone is updated.

5. **check-names** — Hostnames are checked for compliance with the RFC.

6. **max-transfer-time-in** — Specifies the time the slave waits for a zone transfer.

7. **notify** — When zone files are updated, this option, when set to yes, sends DNS NOTIFY messages (default is yes).

7e	**What is vsftpd? Enumerate and explain its features and usability enhancements. What are the distinguishing features of ProFtpd?**
	The default FTP server daemon on Fedora Core and RHEL systems is vsftpd, the Very Secure FTP Daemon, which has a project Web site at http://vsftpd.beasts.org/. Red Hat Software feels confident enough about vsftpd, in fact, to use it to power their own FTP site. So does the OpenBSD FTP site. So do a lot of other sites. vsftpd is extremely lightweight in that it makes sparing use of system resources and does not rely on system binaries for parts of its functionality. It can be tuned, to some degree, to use even fewer resources if need be. **The text vsftpd refers to the name of the product. When vsftpd is used to refer to the actual daemon binary, it appears in monospaced text, for example,** vsftpd**.** To the standard FTP services defined in RFC 959, the core RFC (Request for Comment) that defines the FTP protocol, vsftpd offers the additional security features and usability enhancements listed here: ■■ Support for virtual IP configurations ■■ Support for so-called virtual users ■■ Can run as a standalone daemon or from inetd or xinetd ■■ Configurable on a per-user or per-IP basis ■■ Bandwidth throttling ■■ IPv6-ready Unlike older versions of products formerly known as Red Hat Linux, you no longer have to install a special RPM to provide anonymous FTP services. A couple of tweaks to the vsftpd configuration file and you are set to go. One of the reasons a special RPM is no longer required is that vsftpd is self-contained—that is, it doesn't need access to system binaries, such as a statically linked /bin/ls, to operate. Rather, vsftpd provides internal implementations of commands other FTP daemons (such as the venerable wu-ftpd) expect the host operating system to provide. Although Fedora Core and RHEL prefer and install vsftpd, ProFTPD and NcFTPd deserve mention because they are widely used at busy FTP sites. ProFTPD (www.proftpd.org) is a free FTP server licensed under the GPL. Roughly modeled on the Apache Web server, ProFTPD was designed to be more configurable and more secure than vsftpd. ProFTPD was written from scratch. Other Linux FTP servers, including vsftpd, evolved from the original BSD ftpd server. The following key features distinguish ProFTPD: ■■ Per-directory access configuration using .ftpaccess files, much like Apache's .htaccess file

	■■ An anonymous FTP root directory unencumbered by required directory structures and system binaries ■■ Support for hidden files and directories ■■ Self-contained and does not need to use system binaries or libraries, reducing the likelihood of exploits that take advantage of external programs ■■ Runs as an unprivileged user in standalone mode, decreasing exposure to security attacks that attempt to exploit its root privileges NcFTPd (http://www.ncftp.com) is a commercial FTP server that, like ProFTPD, was written from scratch, optimized for anonymous FTP service, and designed for high performance. Its primary architectural features are its self-described “no-forks” design — not spawning child processes to handle incoming connections and individual directory listings — and its independence from inetd and xinetd. It runs as a standalone server. It is *not* free software, but its features, security, and performance make it a popular FTP server.
7f	**Explain the AddHandler, AddLanguage, AddDefaultCharset and AddCharset directives.**
	AddHandler’s purpose is to define a content handler for specific MIME types. Its syntax is comparable to AddType’s: AddHandler *handler extension* [...] This directive associates any file that has the extension extension with the content handler named by handler. The following statement, for instance, instructs Apache to use the image map handler imap-file with files whose name ends with map: AddHandler imap-file map The AddLanguage directives map filenames to language encodings. So, for example, files ending with .en are treated as English documents, and files ending with .en.gz or .en.tgz are treated as gzip compressed English documents. The LanguagePriority directive, similarly, determines which file the server returns if the browser does not indicate a preference. For example, if the files index.en.html and index.fr.html both exist and a client does not specify a preferred content language, the server returns index.en.html. AddDefaultCharset and AddCharset load Apache’s support for various character sets. AddDefaultCharset specifies the default character set Apache uses to serve content if a browser does not specifically request one. The default character set in Fedora Core and RHEL is UTF-8.