Q) Explain Backing up and Restoring Files in Linux.(april 2013)

1. Backup required to secure data from computer failure, some people may harm others property or system if system administrator not perfect to handle it.
2. We 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.
3. Backup can be taken in high capacity tape drive and can be stored in disks..
4. It‘s it more sensible to back up user accounts and system configuration files from the distribution CDs. CDs would be quicker and easier than getting the basics off a tape archive.
5. System Administrator must decide what to back up and how frequently backup.
6. System Administrator may maintain a series of incremental backups i.e. adding only the files that have changed since the last backup or multiple full backups.
7. System Administrator may use RAID (redundant array of independent disks) for Backup , which is multiple hard drives all containing the same data as insurance against the failure of any one of them, in addition to other backup systems.
8. System administrator should carry restore at least once in non critical time.
9. Need to formulate a plan for bringing the system back up in the event of a failure.

Q) Explain Monitoring and Turning Performance in Linux.

1. System Administrator brings improvement in system performance.
2. System Administrator takes some performance decision at installation time while others are added later.
3. A good example is the use of the hdparm utility, to squeeze the best performance from your equipment, monitor your system carefully and use Linux built-in configuration wisely.
4. Proper monitoring allows you to detect a misbehaving application that might be consuming more resources than it should or failing to exit completely on close.
5. careful system monitoring and diagnostic practices give you an early heads-up when a system component is showing early signs of failure, so that any potential downtime can be minimized.
6. 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.

- **Performance Monitoring**
  - ✓ Memory
  - ✓ Disk Sub System
  - ✓ Network Traffic
  - ✓ CPU

- **What to Monitor?**
  - ✓ Overall Performance
  - ✓ Active Process
  - ✓ Memory Usage
  - ✓ Disk Usage & Performance
  - ✓ CPU Utilization
  - ✓ Network Usage

Q) Explain How to configure a secure system.

S) GRUB (nov 12, april 13)

- 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).
- It contains kernel and initrd image.
- So, in simple terms GRUB just loads and executes Kernel and initrd images.

T) 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 OR /sbin/init

- /sbin/init is the very first process to be executed by kernel in RAM.
- The first script init runs is /etc/rc.d/rc.sysinit. This script starts system swap, checks the file systems, and performs other system initialization.
- Next it looks at the /etc/inittab file to decide the Linux run level.
- The default runlevel is set to 5.
- 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)*
- Next it reads /etc/rc.d/init.d/ functions to determine the procedure to use to set the default system path, start and stop programs, find the process ID(PID) of running process and how to log the success or failure of starting a program.
- The next script to run is /etc/rc.d/rc, which is responsible for starting and stopping services when the runlevel changes and determine the new runlevel.
- In the /etc/rc.d directory are additional directories rc0.d, rc1.d, rc2.d, rc3.d, rc4.d, rc5.d, and rc6.d.

Q) Explain the system SHUTDOWN process with the help of shutdown command and example.

- The shutdown procedure for GNOME GUI desktop is as follows:

1)Choose Actions ? Logout from the top panel menu. A screen appears asking if you want to logout, restart or shutdown.

2) Click the shutdown radio button.

3) Click OK, and your system begins to shutdown.

- If not GUI then the command to shut down is /sbin/shutdown. Any user who are logged into system will receive a notice that the system is going down and anyone trying to log in after the shutdown command has been given will not be allowed to do so.
- Following things will happen after shutdown command is executed.
  i) Runnings processes on the system are sent the SIGTERM signal, which attempts to stop the running process.
  ii) The shutdown program signals the init program to change to a different runlevel,
  i.e. runlevel 0 is used to shutdown the system thereby calling the **rc scripts** of the directory **/etc/rc0.d/***
- In most of the cases only root can use to shutdown the system.

SHUTDOWN Command : The shutdown command has options to alert users on the system when it is going down and to set up a shutdown at a particular time.

Syntex : *shutdown [option] time*

Options:

-k Doesn't shutdown, just warns

-h Halts the system after shutdown

-r reboots instead of turning off the system

-F Forces a file system check on reboot

-n Kills all processes quickly (not recommended)

-t SESC sends a shutdown message but delays shutdown by x seconds

For ex: to turn off the system, do the following:

 # shutdown –h now

This time instead of a complete power off, reboot the system as follows:

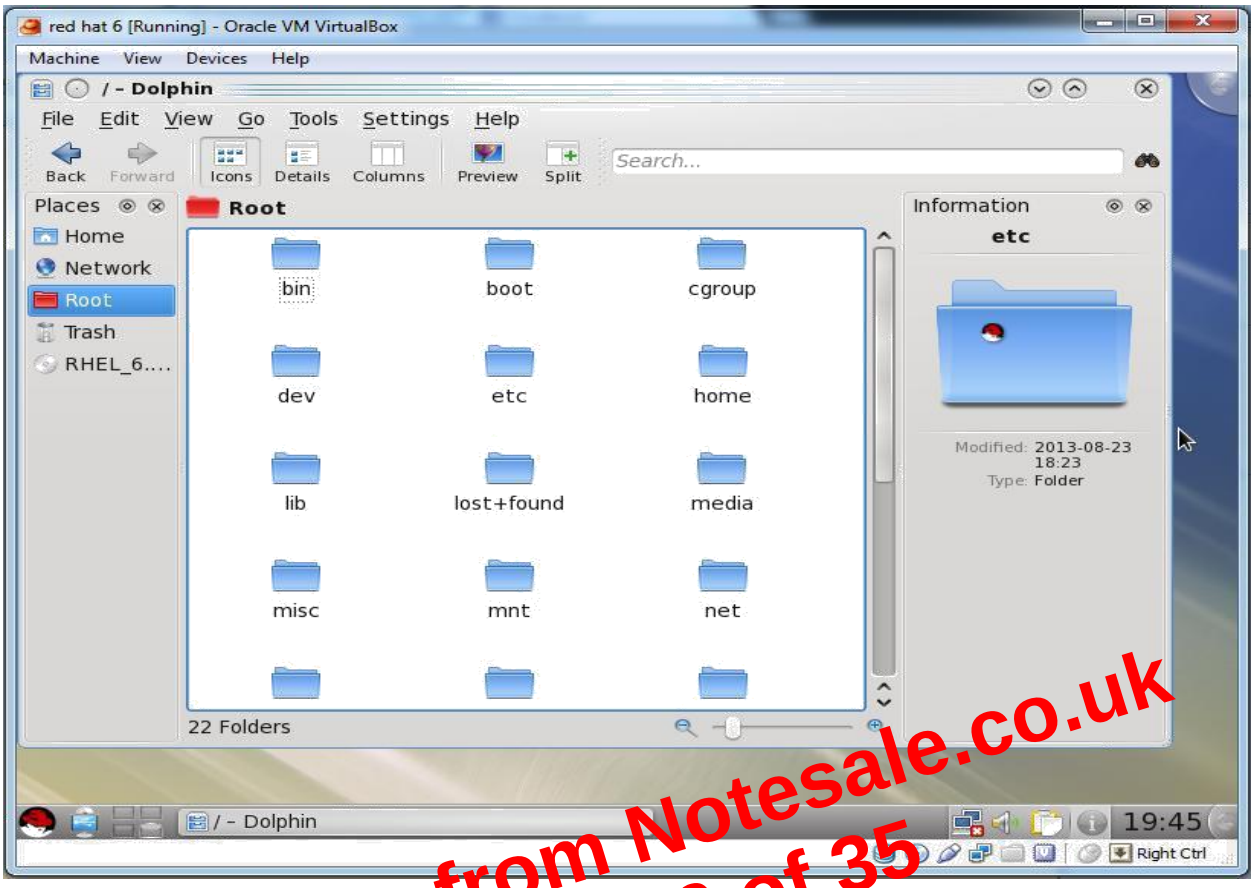
 # shutdown –r now

Delay the shutdown by 2 minutes:

 # shutdown –h 2

Q) What is Boot Strapping? (nov'12, april'13)

- A process that starts operating systems when the user turns on a computer system.
- A boot sequence is the set of operations the computer performs when it is switched on that load an operating system.
- Sequence of operations are as follows:
  i) Locate and start boot loader program
  ii) ROM BIOS



Q) ACCORDING TO THE FHS, THE '/' DIRECTORY MUST CONTAIN OR HAVE LINKS TO THE FOLLOWING DIRECTORIES: (NOV'12)

➢ **The '/' directory**

- The _/' directory is called the *root* directory and is typically at the top of the file system structure. In many systems, the / directory is the only partition on the system and all other directories are mounted under it.
- Figure shows a file system with the / directory mounted as the only partition, with all other directories mounted beneath it.
- The primary purpose of the / directory is booting the system, and to correct any problems that might be preventing the system from booting.
- According to the FHS, the / directory must contain, or have links to, the following directories:

_ **bin** — This directory contains command files for use by the system administrator or other users. The bin directory can not contain subdirectories.

_ **boot** — On Red Hat systems, this is the directory containing the kernel, the

core of the operating system. Also in this directory are files related to booting the system, such as the bootloader.

_ **dev** — This directory contains files with information about devices, either hardware or software devices, on the system.

- **cp :** cp is Unix‘s copy command. The syntax of cp is pretty basic and is just like any other operating system‘s file copy command.
  To copy filename-a to filename-b, type: **cp filename-a filename-b**.
- **rm** : rm is Unix‘s remove command. To remove a file, type rm filename.
- **mv :** mv is the move command. If moving files within a file system, mv operates more like a rename than a move, because just the logical name and logical location are changed. If a file is moved across file systems, mv copies the file over first and then removes it from the old location. To move filename-a to filename-b, type mv filename-a filename-b.
- **chown :** chown changes the user or group ownership of a file or directory. Maintaining proper file ownership helps ensure that only the people who own the file have access to it, since world accessible permissions can be severely limited without inhibiting the use of the file to its rightful owner.To change ownership of a file to username ben, type chown ben filename.
- **chgrp :** chgrp changes only the group ownership of a file or directory. To set the group ownership of a file to group admin, type chgrp admin filename.
- **chmod :** This command changes file access permissions, handy when you want to limit who can and cannot read, write or execute your files. Permissions that can be modified with this command are write access, read access, and executable access for the user owner, the group owner, and all users on the system (world access).
  chmod also determines if the program text should be saved on the swap device and whether or not the command should be run under the user ID of the owner or under the group ID of the group owner.chmod has two methods of changing filename permissions. One way is the numeric method, which sets the user, group and world permissions at once. In this method, 4 represents read, 2 represents write, and 1 represents execute. So if you set a file‘s permissions to 444, you are setting it to be only readable to user, group, and world owners. If you add the numbers together you can add more file permissions. So to make a file writeable (4), readable (2), and executable (1) by user, and not accessible in any way to anybody else, you set it to permissions number 700. That breaks down to (4+2+1) for user, and no permissions for group or world, hence the two 0‘s. To change a file‘s permissions to 700, type **chmod 700 filename**. The other way to specify chmod
  permissions is the character flag method, which changes only the attributes you specify. The letters used to change owner permissions are u for user, g for group, and o for other. To indicate the permissions mode,r is for read, w is for write, and x stands for execute. The + and – signs are used to indicate if the permission is being added or removed from the file. For example, to add readable and writeable permissions to filename-a for the user who owns it, type **chmod u+rw filename**.

Q) Standard and Non-standard Linux File System.

Working With Linux—Supported File Systems

Linux is a very flexible operating system that has a long history of interoperability with other systems on a number of different hardware platforms. A consequence of this friendliness to other operating systems is that Linux can read and write to several different file systems that originated with other operating systems much different from Linux. This section details the different file systems supported and where they originated. One reason that Linux supports so many file systems