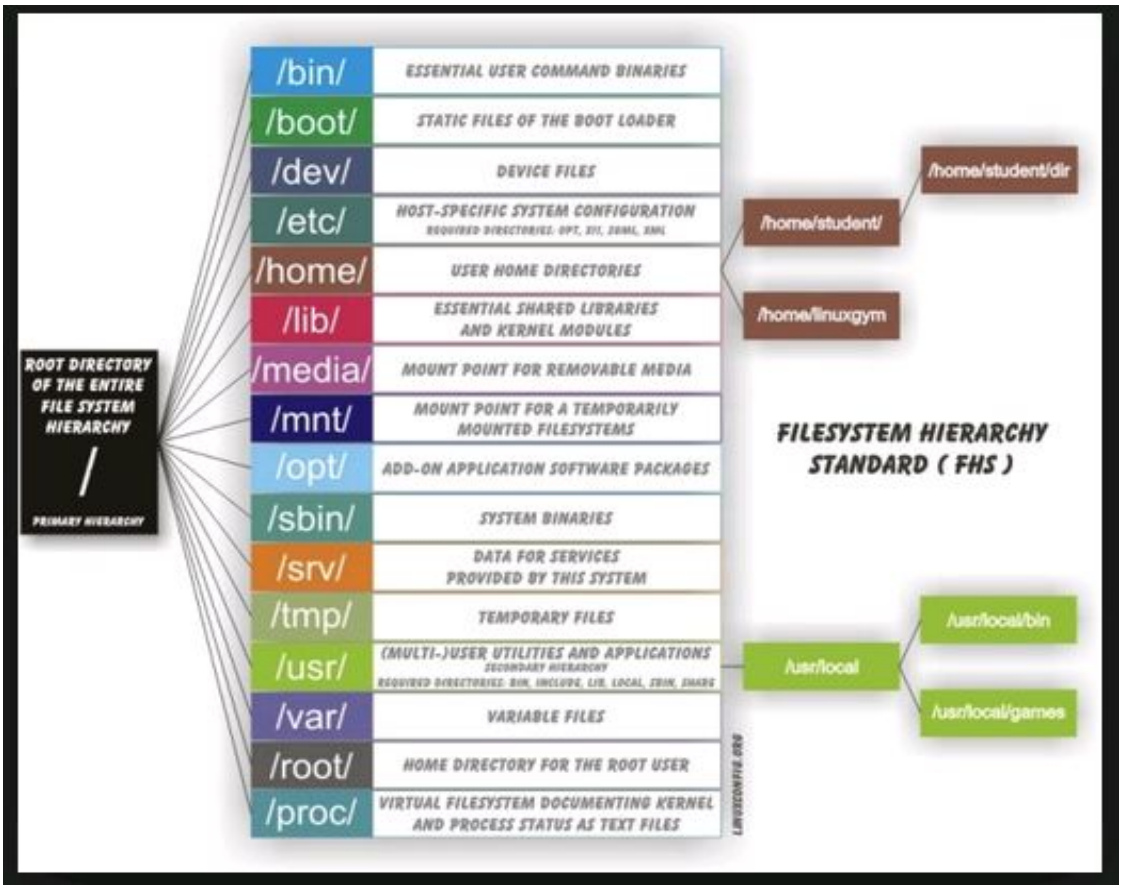
Linux Directory Structure and Important Files Paths Explained

Linux File System, some of the important files, their usability and location.

Linux Directory Structure Diagram



Each of the above directory (which is a file, at the first place) contains important information, required for booting to device drivers, configuration files, etc. Describing briefly the purpose of each directory, we are starting hierarchically.

/bin : All the executable binary programs (file) required during booting, repairing, files required to run into single-user-mode, and other important, basic commands viz., cat, du, df, tar, rpm, wc, history, etc.

/boot : Holds important files during boot-up process, including Linux Kernel.

/dev : Contains device files for all the hardware devices on the machine e.g., cdrom, cpu, etc

/etc : Contains Application's configuration files, startup, shutdown, start, stop script for every

individual program.

/home : Home directory of the users. Every time a new user is created, a directory in the name of user is created within home directory which contains other directories like Desktop, Downloads, Documents, etc.

/lib : The Lib directory contains kernel modules and shared library images required to boot the system and run commands in root file system.

/lost+found : This Directory is installed during installation of Linux, useful for recovering files which may be broken due to unexpected shut-down.

/media : Temporary mount directory is created for removable devices viz., media/cdrom.

/mnt : Temporary mount directory for mounting file system.

/opt : Optional is abbreviated as opt. Contains third party application software. Viz., Java, etc.

/proc : A virtual and pseudo file-system which contains information about running process with a particular Process-id aka pid.

/root : This is the home directory of root user and should never be confused with '/'

/run : This directory is the only clean solution for early-runtime-dir problem.

/sbin : Contains binary executable programs, required by System Administrator, for Maintenance. Viz.,
iptables, fdisk, ifconfig, swapon, reboot, etc.

/srv : Service is abbreviated as 'srv'. This directory contains server specific and service related files.

/sys : Modern Linux distributions include a /sys directory as a virtual filesystem, which stores and allows modification of the devices connected to the system.

/tmp :System's Temporary Directory, Accessible by users and root. Stores temporary files for user and system, till next boot.

/usr : Contains executable binaries, documentation, source code, libraries for second level program.

/var : Stands for variable. The contents of this file is expected to grow. This directory contains log, lock, spool, mail and temp files

Exploring Important file, their location and their Usability

Linux is a complex system which requires a more complex and efficient way to start, stop, maintain and reboot a system unlike Windows.
There is a well defined configuration files, binaries, man pages, info files, etc. for every process in Linux..

/boot/vmlinuz : The Linux Kernel file.

/dev/hda : Device file for the first IDE HDD (Hard Disk Drive)

/dev/hdc : Device file for the IDE Cdrom, commonly

/dev/null : A pseudo device, that don't exist. Sometime garbage output is redirected to /dev/null, so that it gets lost, forever.

/etc/bashrc : Contains system defaults and aliases used by bash shell.

/etc/crontab : A shell script to run specified commands on a predefined time Interval.

/etc/exports : Information of the file system available on network.

/etc/fstab : Information of Disk Drive and their mount point.

/etc/group : Information of Security Group.

/etc/grub.conf : grub bootloader configuration file.

/etc/init.d : Service startup Script.

/etc/lilo.conf : lilo bootloader configuration file.

/etc/hosts : Information of Ip addresses and corresponding host names.

/etc/hosts.allow : List of hosts allowed to access services on the local machine.

/etc/host.deny : List of hosts denied to access services on the local machine.

/etc/inittab : INIT process and their interaction at various run level.

/etc/issue : Allows to edit the pre-login message.

/**etc/modules.conf** : Configuration files for system modules.

/etc/motd : motd stands for Message Of The Day, The Message users gets upon login.

/etc/mtab : Currently mounted blocks information.

/etc/passwd : Contains password of system users in a shadow file, a security implementation.

/etc/printcap : Printer Information

/etc/profile : Bash shell defaults

/etc/profile.d : Application script, executed after login.

/etc/rc.d : Information about run level specific script.

/etc/rc.d/init.d : Run Level Initialisation Script.

/etc/resolv.conf : Domain Name Servers (DNS) being used by System.

/etc/securetty : Terminal List, where root login is possible.

/etc/skel : Script that populates new user home directory.

/etc/termcap : An ASCII file that defines the behaviour of Terminal, console and printers.

/etc/X11 : Configuration files of X-window System.

/usr/bin : Normal user executable commands.

/usr/bin/X11 : Binaries of X windows System.

/usr/include : Contains include files used by ‘c‘ program.

/usr/share : Shared directories of man files, info files, etc.

/usr/lib : Library files which are required during program compilation.

/usr/sbin : Commands for Super User, for System Administration.

/proc/cpuinfo : CPU Information

/**proc/filesystems** : File-system Information being used currently.

/proc/interrupts : Information about the current interrupts being utilised currently
.
/proc/ioports : Contains all the Input/Output addresses used by devices on the server.

/proc/meminfo : Memory Usages Information.

/proc/modules : Currently using kernel module.

/proc/mount : Mounted File-system Information.

/proc/stat : Detailed Statistics of the current System.

/proc/swaps : Swap File Information.

/**version** : Linux Version Information.

/var/log/lastlog : log of last boot process.

/var/log/messages : log of messages produced by syslog daemon at boot.

/var/log/wtmp : list login time and duration of each user on the system currently

Linux File Systems

File System	Max File Size	Max Partition Size	Journaling	Notes
Fat16	2 GiB	2 GiB	No	Legacy
Fat32	4 GiB	8 TiB	No	Legacy
NTFS	2 TiB	256 TiB	Yes	(For Windows Compatibility) NTFS-3g is installed by default in Ubuntu, allowing Read/Write support
ext2	2 TiB	32 TiB	No	Legacy
ext3	2 TiB	32 TiB	Yes	Standard linux filesystem for many years. Best choice for super-standard installation.
ext4	16 TiB	1 EiB	Yes	Modern iteration of ext3. Best choice for new installations where super-standard isn't necessary.
reiserFS	8 TiB	16 TiB	Yes	No longer well-maintained.
JFS	4PiB	32PiB	Yes (metadata)	Created by IBM - Not well maintained.
XFS	8 EiB	8 EiB	Yes (metadata)	Created by SGI. Best choice for a mix of stability and advanced journaling.
GiB = Gibibyte (1024 MiB) :: TiB = Tebibyte (1024 GiB) :: PiB = Pebibyte (1024 TiB) :: EiB = Exbibyte (1024 PiB)				

Linux File Systems: Ext2 vs Ext3 vs Ext4

ext2, ext3 and **ext4** are all filesystems created for Linux. Here I explains the following:

High level difference between these filesystems.
How to create these filesystems.
How to convert from one filesystem type to another.

Ext2

Ext2 stands for second extended file system.
It was introduced in 1993. Developed by Rémy Card.
This was developed to overcome the limitation of the original ext file system.
Ext2 does not have journaling feature.
On flash drives, usb drives, ext2 is recommended, as it doesn't need to do the over head of journaling.
Maximum individual file size can be from 16 GB to 2 TB
Overall ext2 file system size can be from 2 TB to 32 TB

Ext3

Ext3 stands for third extended file system.
It was introduced in 2001. Developed by Stephen Tweedie.
Starting from Linux Kernel 2.4.15 ext3 was available.
The main benefit of ext3 is that it allows journaling.

Journaling has a dedicated area in the file system, where all the changes are tracked. When the system crashes, the possibility of file system corruption is less because of journaling.
Maximum individual file size can be from 16 GB to 2 TB
Overall ext3 file system size can be from 2 TB to 32 TB
There are three types of journaling available in ext3 file system.
Journal – Metadata and content are saved in the journal.
Ordered – Only metadata is saved in the journal. Metadata are journaled only after writing the content to disk. This is the default.
Writeback – Only metadata is saved in the journal. Metadata might be journaled either before or after the content is written to the disk.
You can convert a ext2 file system to ext3 file system directly (without backup/restore).

Ext4

Ext4 stands for fourth extended file system.
It was introduced in 2008.
Starting from Linux Kernel 2.6.19 ext4 was available.
Supports huge individual file size and overall file system size.
Maximum individual file size can be from 16 GB to 16 TB
Overall maximum ext4 file system size is 1 EB (exabyte). 1 EB = 1024 PB (petabyte). 1 PB = 1024 TB (terabyte).
Directory can contain a maximum of 64,000 subdirectories (as opposed to 32,000 in ext3)
You can also mount an existing ext3 fs as ext4 fs (without having to upgrade it).
Several other new features are introduced in ext4: multiblock allocation, delayed allocation, journal checksum. fast fsck, etc. All you need to know is that these new features have improved the performance and reliability of the filesystem when compared to ext3.
In ext4, you also have the option of turning the journaling feature "off"

Creating an ext2, or ext3, or ext4 filesystem

Once you've partitioned your hard disk using fdisk command, use mke2fs to create either ext2, ext3, or ext4 file system.

1.Create an ext2 file system:

```
#mke2fs /dev/sda1
```

2.Create an ext3 file system:

```
#mkfs.ext3 /dev/sda1
(or)
#mke2fs –j /dev/sda1
```

3.Create an ext4 file system:

```
mkfs.ext4 /dev/sda1
(or)
mke2fs -t ext4 /dev/sda1
```

Converting ext2 to ext3

For example, if you are upgrading /dev/sda2 that is mounted as /home, from ext2 to ext3, do the following.

```
umount /dev/sda2

tune2fs -j /dev/sda2

mount /dev/sda2 /home
```

Note: You really don't need to umount and mount it, as ext2 to ext3 conversion can happen on a live file system. But, I feel better doing the conversion offline.

Converting ext3 to ext4

If you are upgrading **/dev/sda2** that is mounted as /home, from **ext3** to **ext4**, do the following.

```
umount /dev/sda2

tune2fs -O extents,uninit_bg,dir_index
/dev/sda2

e2fsck -pf /dev/sda2

mount /dev/sda2 /home
```

try all of the above commands only on a test system, where you can afford to lose all your data.

5 Methods to Identify Your Linux File System Type (Ext2 or Ext3 or Ext4)

here I explain 5 method to find the file system type.

1.**Use df -T Command**

The -T option in the df command displays the file system type.

```
# df -T | awk '{print $1,$2,$NF}' | grep
"^/dev"
/dev/sda1 ext2 /
/dev/sdb1 ext3 /home
/dev/sdc1 ext3 /u01
```

2.Use Mount Command

Use the mount command as shown below.

```
# mount | grep "^/dev"
/dev/sda1 on / type ext2 (rw)
/dev/sdb1 on /home type ext3 (rw)
/dev/sdc1 on /u01 type ext3 (rw)
```

As shown in the above example:
/dev/sda1 is ext2 file system type. (mounted as /)
/dev/sdb1 is ext3 file system type. (mounted as /home)
/dev/sdc1 is ext3 file system type. (mounted as /u01)

3.Use file Command

As root, use the file command as shown below. You need to pass the individual device name to the file command.

```
# file -sL /dev/sda1
/dev/sda1: Linux rev 1.0 ext2 filesystem data (mounted or unclean) (large files)

# file -sL /dev/sdb1
/dev/sda1: Linux rev 1.0 ext3 filesystem data (needs journal recovery)(large files)

# file -sL /dev/sdc1
/dev/sda1: Linux rev 1.0 ext3 filesystem data (needs journal recovery)(large file
```

Note: You should execute the file command as root user. If you execute as non-root user, you'll still get some output. But, that will not display the file system type as shown below.

4.View the /etc/fstab file

If a particular mount point is configured to be mounted automatically during system startup, you can identify its file system type by looking at the /etc/fstab file.

As shown in the example below, / is ext2, /home is ext3, and /u01 is ext3.

```
# cat /etc/fstab
LABEL=/r      /       ext2    defaults    1 1
LABEL=/home   /home   ext3    defaults    0 0
LABEL=/u01    /u01    ext3    defaults    0 0
```

5.Use fsck Command

Execute the fsck command as shown below. This will display the file system type of a given device.

```
# fsck -N /dev/sda1
fsck 1.39 (29-May-2006)
[/sbin/fsck.ext2 (1) -- /] fsck.ext2 /dev/sda1

# fsck -N /dev/sdb1
fsck 1.39 (29-May-2006)
[/sbin/fsck.ext3 (1) -- /home] fsck.ext3 /dev/sdb1

# fsck -N /dev/sdc1
fsck 1.39 (29-May-2006)
[/sbin/fsck.ext3 (1) -- /u01] fsck.ext3 /dev/sdc1
```

Note: If you don't have the root access, but would like to identify your file system type, use /sbin/fsck -N as shown above.

User Management

Here I explained the user management in both ubuntu and centos server.

Centos	Ubuntu
Create a new user adduser or useradd adduser username	Create a new user adduser or useradd adduser username
Delete User userdel username userdel -r username Userdel- is used to delete a user account. If the -r option is used then the user's home directory and mail spool are deleted too.	Delete user deluser username delete the user's home directory when the user is deleted .The deluser command removes a user from the system. To remove the user's files and home directory, you need to add the -remove-home option. deluser -remove-home username
Modify the user name usermod -l new_username old_username	Modify the user name usermod -l new_username old_username
Change the password passwd username	Change the password passwd username

Lock and Unlock user accounts Lock the user usermod -L username Unlock user username -U username	Lock and Unlock user accounts Lock the user passwd -l username Unlock user passwd -u username Users with a locked password are not allowed to change their password
Create a new group groupadd groupname	Create a new group groupadd groupname
Delete group groupdel groupname	Delete group groupdel groupname
View account aging information chage -l username	View account aging information chage -l username
View password status of any user account passwd -S username	View password status of any user account passwd -S username
FILES /etc/passwd User account information /etc/shadow Secure user account information /etc/group /etc/pam.d/passwd PAM configuration for **passwd**.	**FILES** /etc/passwd User account information /etc/shadow Secure user account information /etc/group /etc/pam.d/passwd PAM configuration for **passwd**.
Use chage command to force users to change their password upon First Login chage - d 0 username	Use chage command to force users to change their password upon First Login chage - d 0 username
View current status of the user account chage -l username	View current status of the user account chage -l username
Disable user account Usermod -L -e expire date username or passwd -l username	Disable user account Usermod -L -e expire date username or passwd -l username

File Permissions

Centos	Ubuntu
Change Permission chmod chmod mode filename	Change Permissions chmod chmod mode filename
Change permissions for all files and directories within a directory by using the -R option on the chmod command. chmod -R mode directory-name /	Change permissions for all files and directories within a directory by using the -R option on the chmod command. chmod -R mode directory-name /
Change the Ownership Change Ownership of a file chown newuser filename Change Ownership of a group chown newuser : newgroup filename Skip changing owner and change only the group chown :newgroup filename (`chown` command can only be used by users with root privileges)	Change the Ownership Change Ownership of a file chown newuser filename Change Ownership of a group chown newuser : newgroup filename Skip changing owner and change only the group chown :newgroup filename
Change the group ownership of file chgrp [option]....group file	Change the group ownership of file chgrp [option]....group file
Print user and group information for the specified username or for the current user id id [option]...[username]	Print user and group information for the specified username or for the current user id id [option]...[username]

useradd

When we run ‘useradd‘ command in Linux terminal, it performs following major things:

It edits /etc/passwd, /etc/shadow, /etc/group and /etc/gshadow files for the newly created User account.

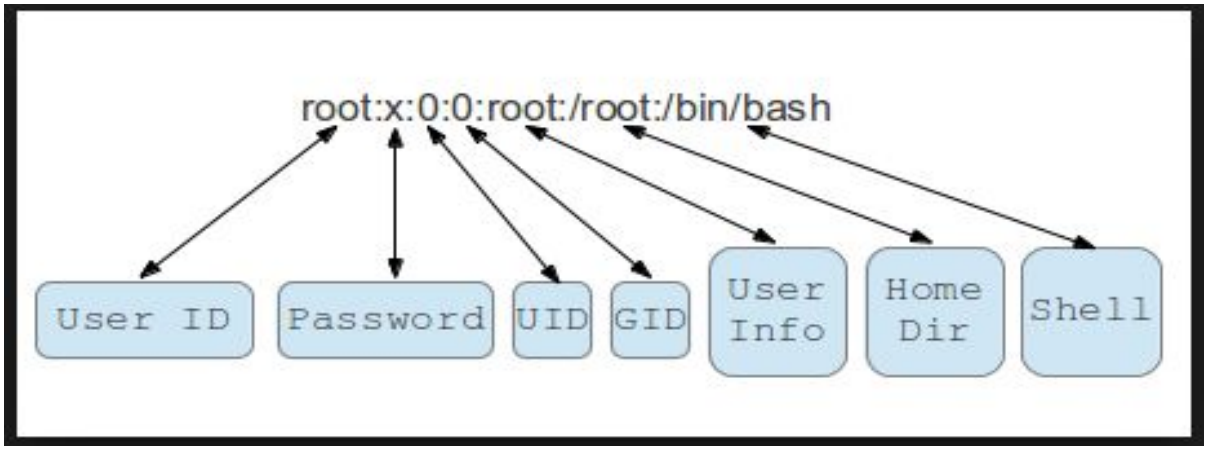
Creates and populate a home directory for the new user.

Sets permissions and ownerships to home directory.

Once a new user created, it’s entry automatically added to the ‘/etc/passwd‘ file.

The file is used to store users information and the entry should be.

tecmint:x:504:504:tecmint:/home/tecmint:/bin/bash



The above entry contains a set of seven colon-separated fields, each field has it's own meaning. Let's see what are these fields:

Username: User login name used to login into system. It should be between 1 to 32 charcters long.

Password: User password (or x character) stored in /etc/shadow file in encrypted format.

User ID (UID): Every user must have a User ID (UID) User Identification Number. By default UID 0 is reserved for root user and UID's ranging from 1-99 are reserved for other predefined accounts. Further UID's ranging from 100-999 are reserved for system accounts and groups.

Group ID (GID): The primary Group ID (GID) Group Identification Number stored in /etc/group file.

User Info: This field is optional and allow you to define extra information about the user. For example, **user full name**. This field is filled by 'finger' command.

Home Directory: The absolute location of user's home directory.

Shell: The absolute location of a user's shell i.e. /bin/bash.

Basic syntax of command is:	useradd [options] username

To Add a New User in Linux When we add a new user in Linux with 'useradd' command it gets created in locked state and to unlock that user account,we need to set a password for that account with 'passwd' command.	**# useradd kiran** **# passwd kiran** **Changing password for user tecmint.** **New UNIX password:** **Retype new UNIX password:** **passwd: all authentication tokens updated successfully.**
Create a User with Different Home Directory You can see the user home directory and other user related information like user id, group id, shell and comments.	**# useradd -d /data/projects anusha** **cat /etc/passwd | grep anusha** **anusha:x:505:505::/data/projects:/bin/bash**
Create a User with Specific User ID	**useradd -u 999 navin** **# cat /etc/passwd | grep kiran** **kiran:x:999:999::/home/kiran:/bin/bash**
Create a User with Specific Group ID	**useradd -u 1000 -g 500 tarunika** **# cat /etc/passwd | grep tarunika** **varun:x:1000:500::/home/varun:/bin/bash**
Add a User to Multiple Groups	**#useradd -G admins,webadmin,developers serji** **# id serji** uid=1001(tecmint) gid=1001(serji) groups=1001(serji),500(admins),501(webadmin),502(developers) context=root:system_r:unconfined_t:SystemLowSystemHigh
Create a User with Account Expiry Date Next, verify the age of account and password with 'chage' command for user 'aparna' after setting account expiry date.	**# useradd -e 2014-03-27 aparna** **# chage -l aparna** **Last password change : Mar 28, 2014** **Password expires : never** **Password inactive : never** **Account expires : Mar 27, 2014** **Minimum number of days between password change : 0** **Maximum number of days between password change : 99999** **Number of days of warning before password expires : 7**
Create a User with Password Expiry Date	**# useradd -e 2014-04-27 -f 45 tecmint**

(The '-f' argument is used to define the number of days after a password expires. A value of 0 inactive the user account as soon as the password has expired. By default, the password expiry value set to -1 means never expire. Here in this example, we will set a account password expiry date i.e. 45 days on a user 'tecmint' using '-e' and '-f' options.)	

CentOS	Ubuntu
CentOS is based on Redhat Enterprise Linux	**Ubuntu Server has its roots in Debian.**

Package Management

In few words, package management is a method of installing and maintaining (which includes updating and probably removing as well) software on the system.

Ubuntu

1. dpkg

dpkg is a package manager for Debian-based systems. It can install, remove, and build packages, but unlike other package management systems, it cannot automatically download and install packages or their dependencies.

2. apt

The apt command is a powerful command-line tool, which works with Ubuntu's Advanced Packaging Tool (APT) performing such functions as installation of new software packages, upgrade of existing software packages, updating of the package list index, and even upgrading the entire Ubuntu system

CentOS

1.rpm

rpm is the package management system used by Linux Standard Base (LSB)-compliant distributions for low-level handling of packages. Just like dpkg, it can query, install, verify, upgrade, and remove packages, and is more frequently used by Fedora-based distributions, such as RHEL and CentOS.

2.yum

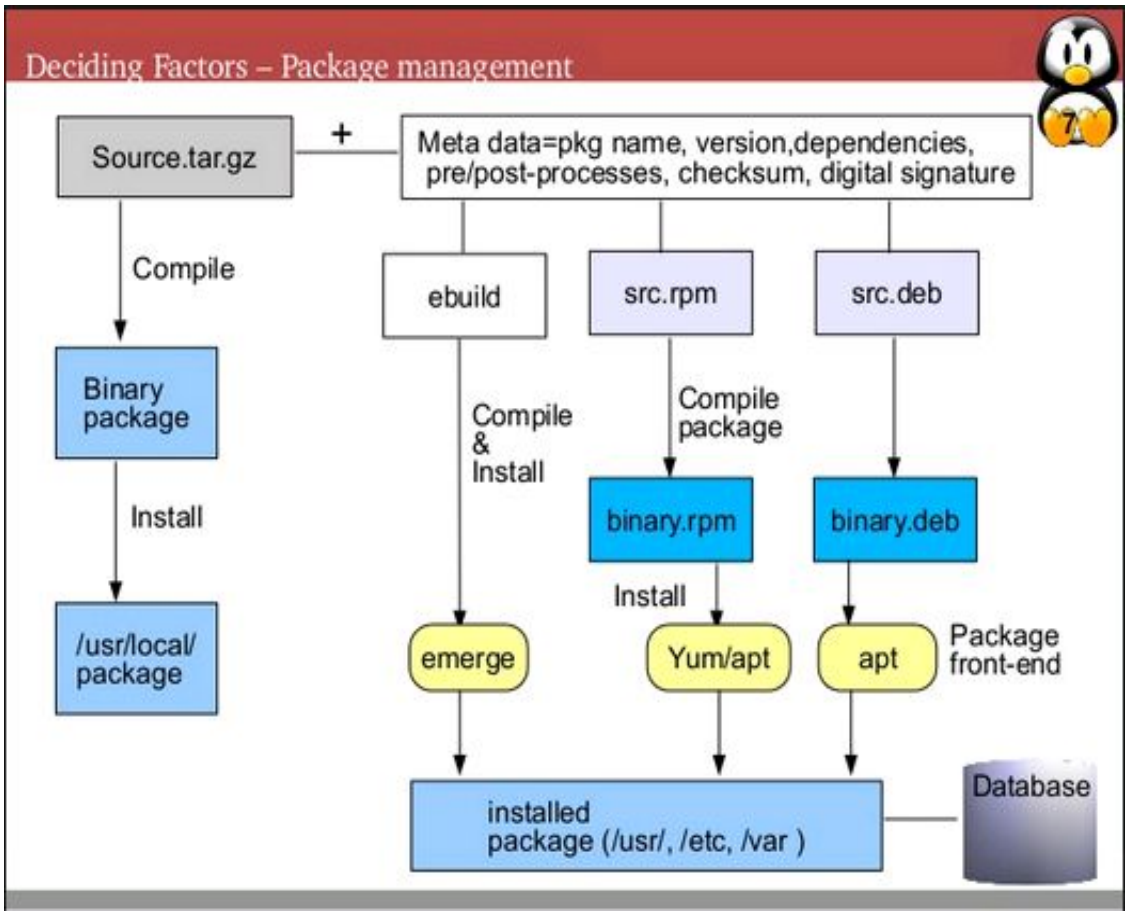
yum adds the functionality of automatic updates and package management with dependency management to RPM-based systems. As a high-level tool, like apt-get or aptitude, yum works with repositories.

Installing packages(.rpm)	Installing Packages (.deb)
# **yum install packagename** # **rpm -ivh packagename.rpm**	# **apt-get install packagename** # **dpkg -i packagename.deb**

Below is a table of equivalent commands for package management on both Ubuntu/Debian and Red Hat/Fedora systems

Task	**CentOS**	**Ubuntu**
Adding, Removing and Upgrading Packages,Services		
Refresh list of available packages	Yum refreshes each time it's used	apt-get update
Install a package from a repository	yum install package_name	apt-get install package_name
Install a package file	yum install package.rpm rpm -i package.rpm	dpkg –install package.deb
Remove a package	rpm -e package_name	apt-get remove package_name
Check for package upgrades	yum check-update	apt-get -s upgrade apt-get -s dist-upgrade
Upgrade packages	yum update rpm -Uvh [args]	apt-get dist-upgrade
Upgrade the entire system	yum upgrade	apt-get dist-upgrade
Package Information		

Get information about an available package	yum search package_name	apt-cache search package_name
Show available packages	yum list available	apt-cache dumpavail
List all installed packages	yum list installed, rpm -qa	dpkg –list
Get information about a package	yum info package_name	apt-cache show package_name
List files in an installed package	rpm -ql package_name	dpkg –listfiles package_name
List documentation files in an installed package	rpm -qd package_name	----
List configuration files in an installed package	rpm -qc package_name	----
Package File Information		
Get information about a package file	rpm -qpi package.rpm	dpkg –info package.deb
List files in a package file	rpm -qpl package.rpm	dpkg –contents package.deb
List documentation files in a package file	rpm -qpd package.rpm	----
List configuration files in a package file	rpm -qpc package.rpm	----
Extact files in a package	rpm2cpio package.rpm | cpio -vid	dpkg-deb –extract package.deb dir-to-extract-to
Find package that installed a file	rpm -qf filename	dpkg –search filename
Find package that provides a particular file	yum provides filename	apt-file search filename
Remove packages from the local cache directory	yum clean packages	apt-get clean
General Packaging System Information		
Package file extension	*.rpm	*.deb
Repository location configuration	/etc/yum.conf	/etc/apt/sources.list



How to Compile and Install Software from Source Code on Linux

While yum, apt-get, rpm are very handy to install a package that is already compiled,you still might encounter some situations where you have to install a software from source code.

1. **First we need to Download the Source Code Package and Unpack it**

The source code for software on Linux comes in the form of compressed tar files, which typically have either **.tar.gz** or **.tar.bz2** extensions. The tools that are used for packing the source code into these tar balls are 'tar'(used for combining multiple files into one), '**gzip**' or **bzip2** (used for compression).
To fetch the source code tarball for a particular software you need to know the URL to the tarball.

Once you have the download link, use '**wget**' to fetch the tarball from command line.

```
$ wget <link to the tarball>
```

The above command will download the tarball into the current directory.

Next you needs to unpack the tarball in order to get access to the source code and other files.Depending on the extension, use one of the following commands:

```
$ tar -xvfz <name of tarball with .tar.gz extension>
(or)
$ tar -xvfj <name of tarball with tar.bz2 extension>
```

2.Read Install Documentation

Once the software source code is downloaded and extracted, the very first thing that one should do is to go through the documentation. This may sound boring to most of us but this is a very important step as doing this step thoroughly would save you from most of the future problems.

The documentation provides information about the software, changes since last version, links to more documentation, information regrading the author of the software, steps for compilation and installation of software etc. So we can see that lots of valuable information is present in the documentation.

This whole information is broadly divided into two files : 'Readme' and 'Install'. While 'Install' covers all the information required for compilation and installation, all the other information is covered in the 'Readme' file. Please note that the name of file and it case may vary.

3.Configuration

Once the above step is over then we can assume that we have sufficient theoretical knowledge about this software and now we can move forward and configure the environment for compiling and installing the software on our system. Most of the packages come along with a configuration script that can be used for configuring the environment.
The file name for configuration file is mostly 'configure'. his script usually accepts parameters that can be used to control some features of this software.

Also this script makes sure that all the tools required for compilation are present in the system.
To learn about the options provided by a specific configuration file, run the following command:

```
$ configure --help
```

To start configuring the build environment, execute the following command :

```
$ ./configure
```

The above command will check and/or create the build environment and if everything goes fine then it produces a file called ‘makefile’.The file ‘makefile’ is used in the compilation of the software.

4.Compilation

Once the makefile is generated, then in the same directory just run the following command:

```
$ make
```

The above command will compile all the source code related to the software. If compilation encounters some problem then error is thrown on the console.

5.Installation

Once the compilation is done successfully then all the required binaries are created. Now is the time to install these binaries in the standard paths so that they can be invoked from anywhere in the system.
To do this run the following command :

```
$ make install
```

Note that some times installing the software may require root privileges, so one may gain the rights and then proceed with the above command.

The above 5 steps show how to fetch, unpack, configure, compile and install the software from source. Additionally one could do some cleanup by removing the directory created while unpacking the software tarball.

Configuration Files

	Ubuntu	**CentOS**
Apache	/etc/apache2.conf	/etc/httpd/conf/httpd.conf
MySql	/etc/mysql/my.cnf	/etc/my.cnf

PHP	/etc/php5/apache/php.ini, /etc/php5/cli/php.ini, /etc/php5/cgi/php.ini	/etc/php.ini
APT(Advanced Packaging Tool)	/etc/apt/sources.list	-----
Ethernet Interface Logical Names	/etc/udev/rules.d/70-persistent-net.rules	
DNS	/etc/resolv.conf	/etc/resolve.conf
Static IP Address Assignment	/etc/network/interfaces	/etc/sysconfig/network-scripts/ifcfg-eth0
Static hostnames	/etc/hosts	/etc/hosts
DHCP	/etc/dhcp/dhcpd.conf	/etc/dhcp/dhcpd.conf
NTP	/etc/ntp.conf	/etc/ntp.conf
dpdk	/etc/dpdk/dpdk.conf	
SSH	/etc/ ssh/ssh.config	/etc/ ssh/ssh.config

Networking

Networks consist of two or more devices, such as computer systems, printers, and related equipment

which are connected by either physical cabling or wireless links for the purpose of sharing and distributing information among the connected devices.

To set IP Address # system-config-network or in the below file: # cat /etc/sysconfig/network-scipts/ifcfg-eth0 To set gateway # system-config-network or in the below file: # cat /etc/sysconfig/network	**To set IP Address & Gatway** # cat /etc/network/interfaces
To set DNS #cat /etc/resolv.conf	**To set DNS** # cat /etc/resolv.conf
To set hostname # hostname yourcomputerName # cat /etc/sysconfig/network	**To set hostname** # cat /etc/hostname
To find version /etc/redhatrelease	**To find version** /etc/issue

ifconfig

The ifconfig command has a variety of options to configure, tune, and debug your system's network interfaces. It's also a quick way to view IP addresses and other network interface information.
Type ifconfig to view the status of all currently active network interfaces, including their names. You can also specify an interface's name to view only information about that interface.

```
#ifconfig
#ifconfig eth0
```

dhclient

The dhclient command can release your computer's IP address and get a new one from your DHCP server. This requires root permissions, so use sudo on Ubuntu. Run dhclient with no options to get a new IP address or use the -r switch to release your current IP address.

```
#sudo dhclient -r
#sudo dhclient
```

ping

ping sends ECHO_REQUEST packets to the address you specify. It's a great way to see whether your computer can communicate with the Internet or a specific IP address.

```
#ping -c 4 google.com
```

tracepath & traceroute

The tracepath command is similar to traceroute, but it doesn't require root privileges.
It's also installed by default on Ubuntu, while traceroute isn't. tracepath traces the network path to a destination you specify and reports each "hop" along the path. If you're having network problems or slowness, tracepath can show you where the network is failing or where the slowness is occurring

```
#tracepath example.com
```

mtr

The mtr command combines ping and tracepath into a single command. mtr will continue to send packets, showing you the ping time to each "hop." This will also show you any problems — in this case,we can see that hop 6 is losing over 20% of the packets

```
#mtr google.com
```

Host

The host command performs DNS lookups. Give it a domain name and you'll see the associated IP address. Give it an IP address and you'll see the associated domain name.

```
#host example.com
#host xx.xx.xx.xx
```

whois

The whois command will show you a website's whois records, so you can view more information about who registered and owns a specific website.

```
#whois example.com
```

Netstat commands

The netstat command can show a lot of different interface statistics, including open sockets and routing tables. Run the netstat command with no options and you'll see a list of open sockets.
There's a lot more you can do with this command. For example, use the netstat -p command to view the programs associated with open sockets

1. Listing all the LISTENING Ports of TCP and UDP connections	**#netstat -a | more**
2. Listing TCP Ports connections	**# netstat -at**
3. Listing UDP Ports connections	**# netstat -au**
4. Listing all LISTENING Connections	**# netstat -l**
5. Listing all TCP Listening Ports	**# netstat -lt**
6. Listing all UDP Listening Ports	**# netstat -lu**
7. Listing all UNIX Listening Ports	**# netstat -lx**
8. Showing Statistics by Protocol	**# netstat -s**
9. Showing Statistics by TCP Protocol	**# netstat -st**
10. Showing Statistics by UDP Protocol	**# netstat -su**
11. Displaying Service name with PID	**# netstat -tp**
12. Displaying Promiscuous Mode	**# netstat -ac 5 | grep tcp**
13. Displaying Kernel IP routing	**# netstat -r**
14. Showing Network Interface Transactions	**# netstat -i**
15. Showing Kernel Interface Table	**# netstat -ie**
16. Displaying IPv4 and IPv6 Information	**# netstat -g**
17. Print Netstat Information Continuously	**# netstat -c**
18. Finding non supportive Address	**# netstat –verbose**
19. Finding Listening Programs	**# netstat -ap | grep http**
20. Displaying RAW Network Statistics	**# netstat --statistics --raw**

DIG Command

Dig (domain information groper) query DNS related information like A Record, CNAME, MX Record etc. This command mainly use to troubleshoot DNS related query.

1.Query Domain "A" Record with +short	**# dig yahoo.com +short**
2. Querying MX Record for Domain	**# dig yahoo.com MX**
3. Querying SOA Record for Domain	**# dig yahoo.com SOA**
4. Querying TTL Record for Domain	**# dig yahoo.com TTL**
5. Querying only answer section	**# dig yahoo.com +nocomments +noquestion +noauthority +noadditional +nostats**
6. Querying ALL DNS Records Types	**# dig yahoo.com ANY +noall +answer**
7. DNS Reverse Look-up	**# dig -x 72.30.38.140 +short**
8. Querying Multiple DNS Records	**# dig yahoo.com mx +noall +answer redhat.com ns +noall +answer**
9. Create .digrc file	**# dig yahoo.com**

ifdown & ifup

The ifdown and ifup commands are the same thing as running ifconfig up or ifconfig down. Given an interface's name, they take the interface down or bring it up. This requires root permissions, so you have to use sudo on Ubuntu.

Enable eth0
ifup eth0

Disable eth0
ifdown eth0

NSLOOKUP Command

nslookup command also use to find out DNS related query.

The following examples shows A Record (IP Address) of vaisakh.com
 # nslookup vaisakh.com

1. Find out "A" record (IP address) of Domain	**# nslookup yahoo.com**
2. Find out Reverse Domain Lookup	**# nslookup 209.191.122.70**
3. Find out specific Domain Lookup.	**# nslookup ir1.fp.vip.mud.yahoo.com.**
4. To Query MX (Mail Exchange) record.	**# nslookup -query=mx www.yahoo.com**
5. To query NS(Name Server) record.	**# nslookup -query=ns www.yahoo.com**
6. To query SOA (Start of Authority) record.	**# nslookup -type=soa www.yahoo.com**
7. To query all Available DNS records.	**# nslookup -query=any yahoo.com**
8. Enable Debug mode	**# nslookup -debug yahoo.com**

Nmap Commands for Linux System/Network Administrators

The Nmap aka Network Mapper is an open source and a very versatile tool for Linux system/network administrators.Nmap is used for exploring networks, perform security scans, network audit and finding open ports on remote machine.
It scans for Live hosts, Operating systems, packet filters and open ports running on remote hosts.

How to Install NMAP in Linux

Most of the today's Linux distributions like Red Hat, CentOS, Fedoro,
Debian and Ubuntu have included Nmap in their default package management repositories called Yum and APT.The both tools are used to install and manage software packages and updates.

To install Nmap on distribution specific use the following command.

 # yum install nmap [on Red Hat based systems]

$ sudo apt-get install nmap [on Debian based systems]

1. Scan a System with Hostname and IP Address	
Scan using Hostname	**[root@server1 ~]# nmap server2.example.com**
Scan using IP Address	**[root@server1 ~]# nmap 192.168.0.101**
2. Scan using "-v" option	**nmap -v server2.example.com**
3.Scan Multiple Hosts	**nmap 192.168.0.101 192.168.0.102 192.168.0.103**
4.Scan a whole Subnet	**[root@server1 ~]# nmap 192.168.0.***
5.Scan Multiple Servers using last octet of IP address	**[root@server1 ~]# nmap 192.168.0.101,102,103**
6.Scan list of Hosts from a File	**[root@server1 ~]# cat > nmaptest.txt** Next, run the following command with "iL" option with nmap command to scan all listed IP address in the file. **[root@server1 ~]# nmap -iL nmaptest.txt**
7.Scan an IP Address Range	**[root@server1 ~]# nmap 192.168.0.101-110**
8.Scan Network Excluding Remote Hosts	**[root@server1 ~]# nmap 192.168.0.* --exclude 192.168.0.100**
9.Scan OS information and Traceroute	**[root@server1 ~]# nmap -A 192.168.0.101**
10. Enable OS Detection with Nmap	**[root@server1 ~]# nmap -O server2.vaisakh.com**
11. Scan a Host to Detect Firewall	**[root@server1 ~]# nmap -sA 192.168.0.101**
12. Scan a Host to check its protected by Firewall	**[root@server1 ~]# nmap -PN 192.168.0.101**
13. Find out Live hosts in a Network	**[root@server1 ~]# nmap -sP 192.168.0.***

14. Perform a Fast Scan	**[root@server1 ~]# nmap -F**
15. Find Nmap version	**192.168.0.101[root@server1 ~]# nmap -V**
16. Scan Ports Consecutively	**root@server1 ~]# nmap -r 192.168.0.101**
17. Print Host interfaces and Routes	**[root@server1 ~]# nmap –iflist**
18. Scan for specific Port	**[root@server1 ~]# nmap -p 80 server2.tecmint.com**
19. Scan a TCP Port	**[root@server1 ~]# nmap -p T:8888,80 server2.tecmint.com**
20. Scan a UDP Port	**[root@server1 ~]# nmap -sU 53 server2.tecmint.com**
21. Scan Multiple Ports	**[root@server1 ~]# nmap -p 80,443 192.168.0.101**
22. Scan Ports by Network Range	**[root@server1 ~]# nmap -p 80-160 192.168.0.101**
23. Find Host Services version Numbers	**[root@server1 ~]# nmap -sV 192.168.0.101**
24. Scan remote hosts using TCP ACK (PA) and TCP Syn (PS).	**[root@server1 ~]# nmap -PS 192.168.0.101**
25. Scan Remote host for specific ports with TCP ACK	**[root@server1 ~]# nmap -PA -p 22,80 192.168.0.101**
26. Scan Remote host for specific ports with TCP Syn	**[root@server1 ~]# nmap -PS -p 22,80 192.168.0.101**
27. Perform a stealthy Scan	**[root@server1 ~]# nmap -sS 192.168.0.101**
28. Check most commonly used Ports with TCP Syn	**[root@server1 ~]# nmap -sT 192.168.0.101**
29. Perform a tcp null scan to fool a firewall	**[root@server1 ~]# nmap -sN 192.168.0.101**

NFS on RHEL/CentOS/Fedora and Debian/Ubuntu

NFS (Network File System) is basically developed for sharing of files and folders between Linux/Unix systems by Sun Microsystems in 1980. It allows you to mount your local file systems over a network and remote hosts to interact with them as they are mounted locally on the same system. With the help of NFS, we can set up file sharing between Unix to Linux system and Linux to Unix system.

Benefits of NFS

NFS allows local access to remote files.
It uses standard client/server architecture for file sharing between all *nix based machines.
With NFS it is not necessary that both machines run on the same OS.
With the help of NFS we can configure centralized storage solutions.
Users get their data irrespective of physical location.
No manual refresh needed for new files.
Newer version of NFS also supports acl, pseudo root mounts.
Can be secured with Firewalls and Kerberos.

Important Files for NFS Configuration

/etc/exports : Its a main configuration file of NFS, all exported files and directories are defined in this file at the NFS Server end.

/etc/fstab : To mount a NFS directory on your system across the reboots, we need to make an entry in **/etc/fstab.**

/etc/sysconfig/nfs : Configuration file of NFS to control on which port rpc and other services are listening.

Setup and Configure NFS Mounts on Linux Server

To setup NFS mounts, we'll be needing at least two Linux/Unix machines.

We need to install NFS packages on our NFS Server as well as on NFS Client machine.
We can install it via "yum" (Red Hat Linux) and "apt-get" (Debian and Ubuntu) package installers.

CentOS
[root@nfsserver ~]# **yum install nfs-utils nfs-utils-lib**
[root@nfsserver ~]# **yum install portmap (not required with NFSv4)**

Ubuntu
[root@nfsserver ~]# **apt-get install nfs-utils nfs-utils-lib**

Now start the services on both machines.

[root@nfsserver ~]# **/etc/init.d/portmap start**
[root@nfsserver ~]# **/etc/init.d/nfs start**
[root@nfsserver ~]# **chkconfig --level 35 portmap on**
[root@nfsserver ~]# **chkconfig --level 35 nfs on**

After installing packages and starting services on both the machines, we need to configure both the machines for file sharing.

Setting Up the NFS Server

Configure Export directory

For sharing a directory with NFS, we need to make an entry in **"/etc/exports"** configuration file.
Here we need to creating a new directory named "**nfsshare**" in **"/"** partition to share with client server, you can also share an already existing directory with NFS.

[root@nfsserver ~]# **mkdir /nfsshare**

Now we need to make an entry in "**/etc/exports**" and restart the services to make our directory shareable in the network.

[root@nfsserver ~]# **vi /etc/exports/nfsshare 192.168.0.101(rw,sync,no_root_squash)**

In the above example, there is a directory in / partition named "nfsshare" is being shared with client IP "192.168.0.101" with read and write (rw) privilege,
you can also use **hostname** of the client in the place of IP in above example.

NFS Options

Some other options we can use in **"/etc/exports**" file for file sharing is as follows.

ro: With the help of this option we can provide read only access to the shared files i.e client will only be able to read.

rw: This option allows the client server to both read and write access within the shared directory.

sync: Sync confirms requests to the shared directory only once the changes have been committed.

no_subtree_check: This option prevents the subtree checking. When a shared directory is the subdirectory of a larger file system, nfs performs scans of every directory above it, in order to verify its permissions and details. Disabling the subtree check may increase the reliability of NFS, but reduce security.

no_root_squash: This phrase allows root to connect to the designated directory.
For more options with "**/etc/exports",** you are recommended to read the man pages for export.

Setting Up the NFS Client

After configuring the NFS server, we need to mount that shared directory or partition in the client server.

Mount Shared Directories on NFS Client

Now at the NFS client end, we need to mount that directory in our server to access it locally.
To do so, first we need to find out that shares available on the remote server or NFS Server.

[root@nfsclient ~]# **showmount -e 192.168.0.100**
Export list for 192.168.0.100:
/nfsshare 192.168.0.101

Above command shows that a directory named "**nfsshare**" is available at "192.168.0.100" to share with your server.

Mount Shared NFS Directory

To mount that shared NFS directory we can use following mount command.

[root@nfsclient ~]# **mount -t nfs 192.168.0.100:/nfsshare /mnt/nfsshare**

The above command will mount that shared directory in **"/mnt/nfsshare"** on the client server.
You can verify it following command.

[root@nfsclient ~]# **mount | grep nfs**
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
nfsd on /proc/fs/nfsd type nfsd (rw)
192.168.0.100:/nfsshare on /mnt type nfs (rw,addr=192.168.0.100)

The above mount command mounted the nfs shared directory on to nfs client temporarily, to mount an NFS directory permanently on your system across the reboots,
we need to make an entry in **"/etc/fstab**".

[root@nfsclient ~]# **vi /etc/fstab**

Add the following new line as shown below.

192.168.0.100:/nfsshare /mnt nfs defaults 0 0

Test the Working of NFS Setup

We can test our NFS server setup by creating a test file on the server end and check its availability at nfs client side or vice-versa.

At the nfsserver end

We have create a new text file named "**nfstest.txt**' in that shared directory.

[root@nfsserver ~]# **cat > /nfsshare/nfstest.txt**

This is a test file to test the working of NFS server setup.

At the nfsclient end

Go to that shared directory in client server and you'll find that shared file without any manual refresh or service restart.

[root@nfsclient]# **ll /mnt/nfsshare**
total 4
-rw-r--r-- 1 root root 61 Sep 21 21:44 nfstest.txt

root@nfsclient ~]# **cat /mnt/nfsshare/nfstest.txt**

This is a test file to test the working of NFS server setup.

Removing the NFS Mount

If you want to unmount that shared directory from your server after you are done with the file sharing, you can simply **unmount** that particular directory with "**umount**" command. See this example below.

root@nfsclient ~]# **umount /mnt/nfsshare**

You can see that the mounts were removed by then looking at the filesystem again.

[root@nfsclient ~]# **df -h -F nfs**

You'll see that those shared directories are not available any more.

Important commands for NFS

showmount -e : Shows the available shares on your local machine

showmount -e <server-ip or hostname>: Lists the available shares at the remote server

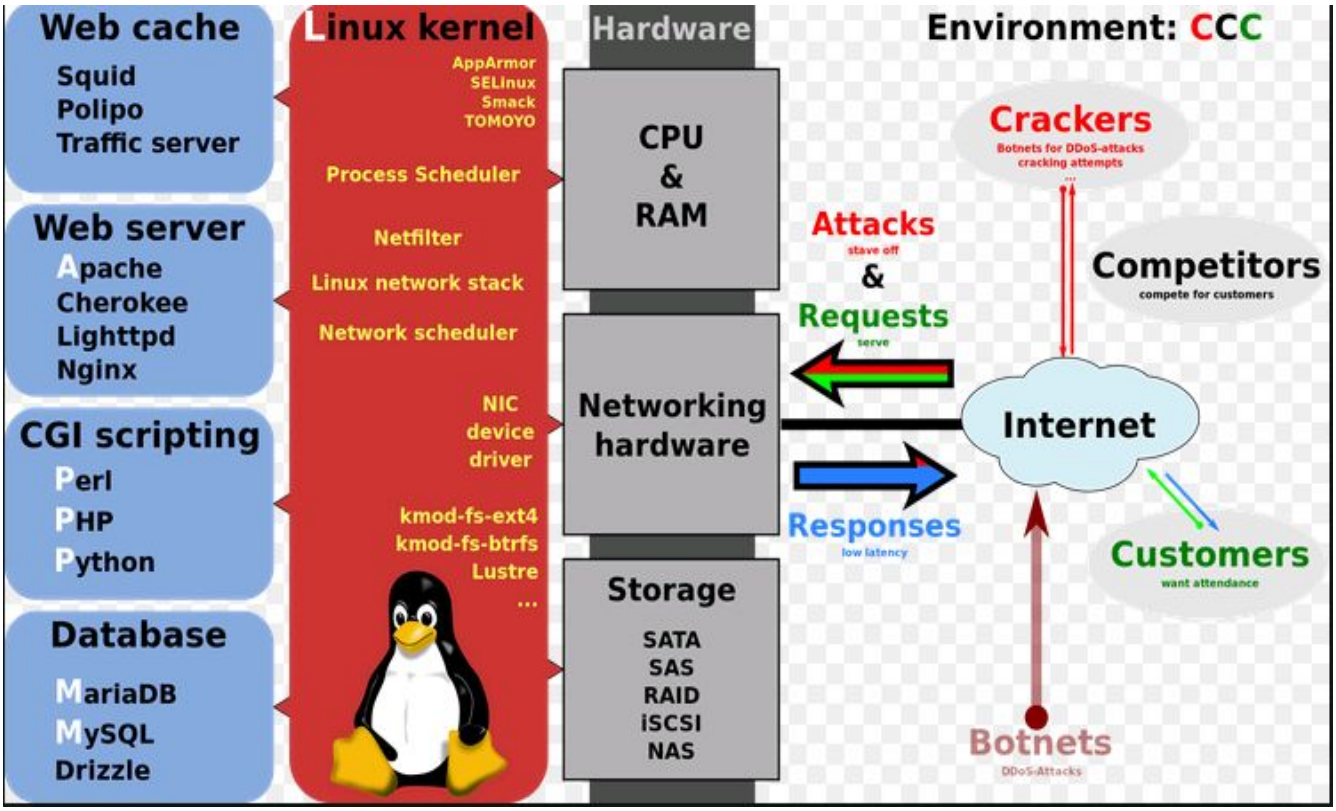
showmount -d : Lists all the sub directories

exportfs -v : Displays a list of shares files and options on a server

exportfs -a : Exports all shares listed in /etc/exports, or given name

exportfs -u : Unexports all shares listed in /etc/exports, or given name
exportfs -r : Refresh the server's list after modifying /etc/exports

System Hardware Information



Disk space of file systems

The '**df**' command stand for "disk filesystem", it is used to get full summary of available and used disk space usage of file system on Linux system.

Check File System Disk Space Usage	# **df**

(information of device name, total blocks, total disk space, used disk space, available disk space and mount points on a file system.) Display Information of all File System Disk Space Usage	**# df -a**
Show Disk Space Usage in Human Readable Format	**#df -h**
Display Information of /home File System	**# df -hT /home**
Display Information of File System in **Bytes**	**# df -k**
Display Information of File System in **MB**	**# df -m**
Display Information of File System in **GB**	**# df -h**
Display File System Inodes	**#df -i**
Display File System Type	**# df -T**
Include Certain File System Type (If you want to display certain file system type use the '-t' option. For example, the following command will only display ext3 file system.)	**# df -t ext3**
Exclude Certain File System Type (If you want to display file system type that doesn't belongs to ext3 type use the option as '-x'. For example, the following command will only display other file systems types other than ext3.)	**# df -x ext3**
Display Information of df Command	**df --help**

du (Disk Usage) Commands

The "du" (Disk Usage) is a standard Unix/Linux command, used to check the information of disk usage of files and directories on a machine.

To check disk usage summary of a /home/vaisakh	**# du /home/vaisakh**
Using "-h" option with "du" command provides results in "Human	**# du -h /home/vaisakh**

Readable Format“. Means you can see sizes in Bytes, Kilobytes, Megabytes, Gigabytes etc.	
To get the summary of a grand total disk usage size of an directory use the option “-s” as follows.	**# du -sh /home/vaisakh**
Using “-a” flag with “du” command displays the disk usage of all the files and directories.	**# du -a /home/vaisakh**
Using “-a” flag along with “-h” displays disk usage of all files and folders in human readeable format. The below output is more easy to understand as it shows the files in Kilobytes, Megabytes etc.	**# du -ah /home/vaisakh**
Find out the disk usage of a directory tree with its subtress in Kilobyte blcoks.Use the “-k” (displays size in 1024 bytes units).	**# du -k /home/vaisakh**
The “-c” flag provides a grand total usage disk space at the last line. If your directory taken 674MB space, then the last last two line of the output would be.	**# du -ch /home/vaisakh**
Display the disk usage based on modification of time, use the flag “–time” as shown below.	**# du -ha --time /home/vaisakh**

‘free’ Commands to Check Memory Usage in Linux

The most important and single way of determining the total available space of the physical memory and swap memory is by using “free” command.

Display System Memory	**# free**
display the size of memory in Bytes.	**# free -b**
Display Memory in Kilo Bytes	**# free -k**
Display Memory in Megabytes	**# free -m**
Display Memory in Gigabytes	**# free -g**

Display Total Line	**# free -t**
Dispaly Memory Status for Regular Intervals (The -s option with number, used to update free command at regular intervals. For example, the below command will update free command every 5 seconds.)	**# free -s 5**
Show Low and High Memory Statistics (The -l switch displays detailed high and low memory size statistics.)	**# free -l**
Check Free Version (The -V option, display free command version information.)	**# free -V**

fdisk Commands to Manage Linux Disk Partitions

fdisk stands (for "fixed disk or format disk") is an most commonly used command-line based disk manipulation utility for a Linux/Unix systems.

View all Disk Partitions in Linux	**# fdisk -l**
View Specific Disk Partition in Linux	**# fdisk -l /dev/sda**
Print all Partition Table in Linux	**# fdisk /dev/sda** From the command mode, enter **'p'** ========== Command (m for help): **p** Disk /dev/sda: 637.8 GB, 637802643456 bytes 255 heads, 63 sectors/track, 77541 cylinders Units = cylinders of 16065 * 512 = 8225280 bytes Device Boot Start End Blocks Id System /dev/sda1 * 1 13 104391 83 Linux /dev/sda2 14 2624 20972857+ 83 Linux /dev/sda3 2625 4582 15727635 83 Linux /dev/sda4 4583 77541 586043167+ 5 Extended /dev/sda5 4583 5887 10482381 83 Linux /dev/sda6 5888 7192 10482381 83 Linux ==========

1. How to Delete a Partition in Linux

If you would like to delete a specific partition (i.e **/dev/sda6**) from the specific hard disk such as **/dev/sda.**
You must be in **fdisk** command mode to do this.

 # fdisk /dev/sda

Next, enter **'d'** to delete any given partition name from the system. As I enter '**d'**,it will prompt me to enter partition number that I want to delete from **/dev/sda hard disk.** Suppose I enter number '**4**' here, then it will delete partition number '**4**' **(i.e. /dev/sda4)** disk and shows free space in partition table. Enter **'w'** to write table to disk and exit after making new alterations to partition table. The new changes would only take place after next reboot of system.

This can be easily understood from the below output.

==========

 # fdisk /dev/sda
WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
switch off the mode (command 'c') and change display units to
sectors (command 'u').
Command (m for help): **d**
Partition number (1-4): **4**
Command (m for help): **w**
The partition table has been altered!
Calling ioctl() to re-read partition table.
WARNING: Re-reading the partition table failed with error 16: Device or resource busy.
The kernel still uses the old table. The new table will be used at
the next reboot or after you run partprobe(8) or kpartx(8)
Syncing disks.
You have new mail in /var/spool/mail/root

==========

Note:
option **'d'** will completely delete partition from system and may lost all data in partition.

2.**How to Create a New Partition in Linux**

If you've free space left on one of your device say **/dev/sda** and would like to create a new partition under it. Then you must be in fdisk command mode of **/dev/sda.** Type the following command to enter into command mode of specific hard disk.

 # fdisk /dev/sda

After entering in command mode, now press "**n**" command to create a new partition under **/dev/sda** with specific size. This can be demonstrated with the help of following given output.

==========

[root@vaisakh ~]# **fdisk /dev/sda**
WARNING: DOS-compatible mode is deprecated. It's strongly recommended toswitch off the mode (command 'c') and change display units to
sectors (command 'u').
Command (m for help): **n**
Command action
e extended
p primary partition (1-4)
e

==========

While creating a new partition, it will ask you two options '**extended**' or '**primary**' partition creation. Press '**e**' for extended partition and '**p**' for primary partition.Then it will ask you to enter following two inputs.

First cylinder number of the partition to be create.
Last cylinder number of the partition to be created (Last cylinder, +cylinders or +size).

You can enter the size of cylinder by adding "+5000M" in last cylinder. Here, '+' means addition and 5000M means size of new partition (i.e 5000MB). Please keep in mind that after creating a new partition, you should run **'w'** command to alter and save new changes to partition table and finally reboot your system to verify newly created partition.

==========

Command (m for help): **w**
The partition table has been altered!
Calling ioctl() to re-read partition table.
WARNING: Re-reading the partition table failed with error 16: Device or resource busy.
The kernel still uses the old table. The new table will be used at
the next reboot or after you run partprobe(8) or kpartx(8)
Syncing disks.

3.How to Format a Partition in Linux

After the new partition is created, don't skip to format the newly created partition using **'mkfs'** command. Type the following command in the terminal to format a partition. Here **/dev/sda4** is my newly created partition.

[root@vaisakh ~]# **mkfs.ext4 /dev/sda4**

4.How to Check Size of a Partition in Linux

After formatting new partition, check the size of that partition using flag '**s'** (displays size in blocks) with fdisk command. This way you can check size of any specific device.

[root@vaisakh ~]# **fdisk -s /dev/sda2**

5.How to Fix Partition Table Order

If you've deleted a logical partition and again recreated it, you might notice '**partition out of order**' problem or error message like '**Partition table entries are not in disk order'**.

For example, when three logical partitions such as **(sda4, sda5 and sda6)** are deleted, and new partition created, you might expect the new partition name would be **sda4.** But,the system would create it as **sda5.** This happens because of, after the partition are deleted, **sda7** partition had been moved .as **sda4** and free space shift to the end.

To fix such partition order problems, and assign sda4 to the newly created partition,issue the '**x**' to enter an extra functionality section and then enter **'f'** expert command to fix the order of partition table as shown below.

After, running **'f'** command, don't forget to run **'w'** command to save and exit from **fdisk** command mode. Once it fixed partition table order, you will no longer get error messages.

-==========

 # fdisk /dev/sda
WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
switch off the mode (command 'c') and change display units to
sectors (command 'u').
Command (m for help): **x**
Expert command (m for help): **f**
Done.
Expert command (m for help): **w**
The partition table has been altered!
Calling ioctl() to re-read partition table.
WARNING: Re-reading the partition table failed with error 16: Device or resource busy.
The kernel still uses the old table. The new table will be used at
the next reboot or after you run partprobe(8) or kpartx(8)
Syncing disks.
==========

To View Linux CPU Information

lscpu

To view information about your CPU, use the **lscpu** command as it shows information about your

CPU architecture such as number of CPU's, cores, CPU family model, CPU caches, threads, etc from sysfs and /proc/cpuinfo.

```
[root@vaisakh ~]#
lscpu
```

To View Linux System Information

Command	O/P
$ uname	Linux
$ uname -n	vaisakh.com
$ uname -v (To get information about kernel-version, use '-v' switch.)	#64-Ubuntu SMP Mon Sep 22 21:28:38 UTC 2014
$ uname -r (To get the information about your kernel release, use '-r' switch.)	3.13.0-37-generic
$ uname -m (To print your machine hardware name, use '-m' switch:)	x86_64
$ uname -a (All this information can be printed at once by running 'uname -a' command)	Linux vaisakh.com 3.13.0-37-generic #64-Ubuntu SMP Mon Sep 22 21:28:38 UTC 2014 x86_64 x86_64 x86_64 GNU/Linux

lshw - List Hardware

When executing lshw without option, you will get detailed information on the hardware configuration such as cpu, disks, memory, usb controllers etc of the machine in text format.

install the lshw

If you are using Debian or any Debian derivative such as Ubuntu or Linux Mint just type.

```
~$ sudo apt-get install lshw
```

If you are using Fedora or Cent Linux use yum.

```
~$ sudo yum install lshw
```

Following is the structure of lshw output.

To print information about your **Linux system hardware,** run this command	**$ sudo lshw**
You can print a summary of your hardware information by using the **-short option.**	**$ sudo lshw -short**
If you wish to generate output as a html file, you can use the option -html.	**$ sudo lshw -html > lshw.html**
Get Information about the **Disks using** lshw	**# lshw -class disk**
Get Information about **Physical Memory (RAM)** of the System	**# lshw -class memory**

Note: lshw must be run as root to get a full report. lshw will display partial report with a warning message as shown below when you execute it from a non-root user.

hwinfo - Hardware Information

Hwinfo is another general purpose hardware probing utility that can report detailed and brief information about multiple different hardware components, and more than what lshw can report.

```
$ hwinfo --short
```

How to Collect Linux Block Device Information

Block devices are storage devices such as hard disks, flash drives etc. lsblk command is used to report information about block devices as follows.

```
$ lsblk
```

To view all block devices on your system then include the -a option.

```
$ lsblk -a
```

Print USB Controllers Information

lsusb - List usb buses and device details

This command shows the USB controllers and details about devices connected to them. By default brief information is printed. Use the verbose option "-v" to print detailed information about each usb port

The lsusb command is used to report information about USB controllers and all the devices that are connected to them.

```
$ lsusb
```

use the **-v** option to generate a detailed information about each USB device.

```
$ lsusb -v
```

How to Print PCI Devices Information

PCI devices may included usb ports, graphics cards, network adapters etc. The **lspci** tool is used to generate information concerning all PCI controllers on your system plus the devices that are connected to them.

To print information about PCI devices run the following command.

```
$ lspci
```

Use the -t option to produce output in a tree format.

```
$ lspci -t
```

Use the -v option to produce detailed information about each connected device.

```
$ lspci -v
```

lspci Command Examples to Get PCI Bus Hardware Device Info

lspci stands for list pci. Think of this command as "ls" + "pci". This will display information about all the PCI bus in your server. it will also display information about all the hardware devices that are connected to your PCI and PCIe bus. It will also display information about all the hardware devices that are connected to your PCI and PCIe bus.

Default Usage (The first field is the slot information in this format: [domain:]bus:device.function)	**# lspci**
lspci Output in Tree Format	**# lspci -t**
Detailed Device Information	**lspci -v**
Display Device Codes in the Output	**lspci -n**
want to display both the description and the number,	**# lspci -nn**
Display Kernel Drivers	**# lspci -k**

/proc files

Many of the virtual files in the /proc directory contain information about hardware and configurations. Here are some of them

CPU/Memory information

 # cpu information

```
$ cat /proc/cpuinfo
```

 # memory information

```
$ cat /proc/meminfo
```

Linux/kernel information

```
$ cat /proc/version
```

SCSI/Sata devices

```
$ cat /proc/scsi/scsi
```

Partitions

```
$ cat /proc/partitions
```

Hardware Information with Dmidecode Command on Linux

If we want to upgrade a system we need to gather information like Memory, BIOS and CPU etc.
With help of Dmidecode command we will come to know the details without opening system chasis.
Dmidecode command works for RHEL/CentOS/Fedora/Ubuntu Linux.

Basic Output of Demidecode	**# dmidecode 2.11** ========== # dmidecode 2.11 SMBIOS version fixup (2.31 -> 2.3). SMBIOS 2.3 present. 45 structures occupying 1642 bytes. Table at 0x000E0010. Handle 0x0000, DMI type 0, 20 bytes BIOS Information Vendor: Phoenix Technologies LTD Version: 6.00 Release Date: 12/06/2006 Address: 0xE78A0 Runtime Size: 100192 bytes ROM Size: 64 kB Characteristics: ISA is supported PCI is supported PC Card (PCMCIA) is supported PNP is supported APM is supported BIOS is upgradeable BIOS shadowing is allowed ESCD support is available USB legacy is supported Smart battery is supported BIOS boot specification is supported
How to Get DMI Types (DMI Id give us particular hardware information of system.Id 6 will give us Memory Module information.)	**# dmidecode -t 6**
Get BIOS information	**dmidecode -t bios**

Get the Manufacturer, Model and Serial Number dmidecode -t system	**dmidecode -t system** **dmidecode -t processor**

Information about SATA Devices

how to find the block device /dev/sda1 which the harddisk on my system.

```
hdparm  /dev/sda1
```

To find information about device geometry interms of cylinders, heads, sectors, size and the starting offset of the device, use the -g option.

```
hdparm -g /dev/sda1
```