# Unit – IV System administration

**Changing Permissions and ownerships :**

The **chmod** command is used for changing the access permissions of files and directories. Suppose we have a regular file called testfile, and the file has full access permissions for all the groups (owner/user, group, others), then the long directory listing would show -rwxrwxrwx as the file's permissions.

Examples: ( **Setting file permissions in symbolic mode )**

1. To wipe out all the permissions and add read permission for everybody:
   **$ chmod a=r testfile** ## After the command, the file's permissions would be -r--r--r--
2. To add execute permissions for the group:
   **$ chmod g+x testfile** ## Now, the file's permissions would be -r--r-xr--
3. To add both write and execute permissions for the file's owner :
   **$ chmod u+wx testfile** ## After this, the file permissions will be -rwxr-xr--
4. To remove the execute permission from both the file's owner and group
   **$ chmod ug-x testfile** ## Now, the permissions are -rw-r--r--

Examples: ( **Setting file permissions in numeric mode )**

1. **$ chmod 755 testfile** ## This would change the testfile's permissions to -rwxr-xr-x.
2. **$ chmod 640 testfile** ## In this case, testfile's permissions would be -rw-r-----.

**Changing file owner ( chown command) :**

"chown" command allows to change the owner of a file. You can give ownership of your files to somebody else, using the command "chown". The "chown" command also allows to change the group ownership.

For example, if George decides to give ownership of myfile to Robert, use the command:

**chown robert myfile**

**Creating and managing groups**

Users are members of a default group. New users are added to a group of the same group name as the user name. The default group for a user is specified in the file/etc/passwd

A new user may be created and assigned a group with the **useradd** command:

Examples:

1. To add a new user as a member to the group "accounting":
   useradd -m -g accounting user2
2. To add a new user as member of the group "accounting" and supplementary group "tally":
   useradd -m -g accounting -G tally user1
3. To View group membership for a user with the command "groups".
   groups user2

The group has a group system number (gid) associated with it and this is defined in /etc/group

Other group Commands**:**

1) **gpasswd**: administer the /etc/group file
2) **groupadd**: Create a new group
   Example: groupadd accounting
3) **groupdel**: Delete a group
   Example: groupdel accounting
4) **groupmod**: Modify group attributes ( e.g., the group ID, group name, etc)
   Format: groupmod [-g gid [-o ]] [-n new_group_name] group

The options of **groupmod** command are:

**-g** The group ID of the given *GROUP* will be changed to *GID*. The value of *GID* must be a non-negative decimal integer. This value must be unique, unless the **-o** option is used. Values between 0 and 999 are typically reserved for system groups.

**-n** The name of the group will be changed to *NEW_GROUP* name.

**-o** (n**on-unique)** When used with the **-g** option, allow to change the group *GID* to a non-unique value.

**-p** gives the encrypted password

Examples:

1. Change name of a group
   groupmod -n accounting guys
2. To change groupid of group:
   groupmod -g 777 oldgroup
3. To use same gid for multiple groups, use -o option
   groupmod -g 777 –o newgroup

**Changing file group : ( chgrp command )**

You can change the group of a user, by using the command "chgrp". The "chown" command also allows to change the group ownership.

Example: If Robert later on decides to make the file only available to members of the group “SeniorAdmin" group rather than the old group:

chgrp senioradmin myfile

The following chown command can also do the same:

chown robert:senioradmin myfile

# Disabling a user temporarily

It is sometimes necessary to temporarily disable an account, without removing it.

**1) Editing /etc/shadow**

The easiest way to disable the user account is to modify a /etc/shadow file. It is the file holding encrypted passwords for users listed /etc/passwd.

Here is an exaple of user entry found in the /etc/shadow file:

Tom:$6dKR$Yku3LWgJmomsynpcle9BCA:15711:0:99999:7:::

Here, the second field is the encrypted password.

You can replace the password with "*" or "!".

Tom:*$6dKR$Yku3LWgJmomsynpcle9BCA:15711:0:99999:7:::

**2) Using passwd command**

Passwd command can be used to disable the user account.

passwd Tom –l

Output : "Password changed."

Above command changes the shadow file and adds "!" in front of the user password:

To enable the account just unlock it using –u option as follows:

passwd Tom -u

You can also enable account by removing manually the "!" character from the user's password line in /etc/shadow.

**3) Using usermod command**

usermod -L Tom

Any login method, which uses the /etc/shadow file , will not decrypt the user's password and thus will not allow login.

To enable the user account simply remove "!" from the /etc/shadow file or use the usermod command: usermod -U Tom

**4) Using nologin Shell**

**On Debian/Ubuntu : Use chsh (change shell) command to change the users shell in /etc/passwd file from something like /bin/bash or /bin/sh to /bin/false meaning do nothing**

sudo chsh -s /bin/false Tom

On RHL/Fedora: Use **chsh** (**change shell**) command to change the users shell in **/etc/passwd** file from something like /bin/bash or /bin/sh to /bin/nologin meaning refuse a login.

# chsh -s /bin/nologin Tom

**5)** The best way to disable an account is to change its shell into a special program that just prints a message

**6) Permanently remove user account**

You can permanently remove the user; just run userdel command.

userdel Tom

Or

userdel -r Tom

**(Make sure to check home of the user before running this command.)**

## Creating File System:

### File System Types

Creating a file system writes information to the device and creates order of the empty space. This file system–related data consumes a small percentage of the space. The remaining space on the disk drive is split into small sized segments called blocks. Linux supports a number of file system types:

| Filesystem | Description |
|---|---|
| ext2 | (second extended file system) High performance for fixed disk and removable media |
| ext3 | Journaling file system. It is a filesystem that maintains a special file called a journal that is used to repair any inconsistencies that occur as the result of an improper shutdown of a computer. |
| ext4 | Supports larger files and file system sizes |
| vfat | MS-DOS file system useful when sharing files between Windows and Linux |
| XFS | High-performance journaling file system |

### mkfs - build a Linux filesystem

The command to build a Linux file system on a device, or hard disk partition, is mkfs. The syntax for the command is:

```
# mkfs [options] device
```

The device argument is either the device name (e.g. /dev/hda1), or a regular file that contain the filesystem.

**-t** Specify the type of filesystem to be built. If not specified, the default filesystem type ( ext2) is used.

**-V, --verbose** Produce verbose output, including all filesystem-specific commands that are executed. Specifying this option more than once inhibits execution of any filesystem-specific commands. This is really only useful for testing

**-V, --version** Display version information and exit. (Option -V will display version information only when it is the only parameter, otherwise it will work as --verbose.)

**Eg:**

To create an ext2 file system, mkfs /dev/xvdd1

To create an ext3 file system, mkfs –t ext3 /dev/xvdd1

To create an ext4 file system, mkfs –t ext4 /dev/xvdd1

# Mounting File Systems :

**Mounting** a filesystem **means** making a filesystem accessible at a point in the **Linux** directory tree. File systems on different removable devices, such as CDs, DVDs, or USB flash drives and partitions, must be attached to the directory hierarchy to be accessed. To attach a partition or device, a mount point must be created. A mount point is simply a directory created with the mkdir command. Then attach the partition by using the mount command.

1) mount -- Mount/Attach a File System to a File Tree.

Syntax for the mount command is:

mount [options] device_file mount_point

Examples:

1. The following example creates a mount point (/test) and attaches the partition:

mkdir /test

mount /dev/xvdf1 /test

2. mount -- List all mounted FileSystems
3. mount -l -- List all mounted FSs with Labels.
4. mount -a -- Mount all entries in the /etc/fstab.
5. mount /mnt/cdrom -- Mount known device to cdrom dir. Mount will find the cdrom device (/dev/hdX) from /etc/fstab.
6. mount /dev/hdd -- Mount device at the known place. Mount will find the mount point (Ex. /mnt/cdrom) from /etc/fstab.
7. mount -t ext2 /dev/hda2 /tmp/dir1 -- Mount a Linux partition at /tmp/dir1.

**Unmounting File Systems**

2) **umount :** To unmount a file system, use the **umount** command. The partition name, the device name, or the mount point is used as an argument.

Examples:

umount /mnt/test1

umount /test

## Checking and monitoring System Performance.

**Checking Hardware** : When the System boots, the kernel detects the hardware and loads drivers that allow linux to work with that hardware. Messages about hardware detection scroll quickly off the screen when you boot. The commands used to check hardware are:

1. **dmesg command**: To view kernel boot messages after linux comes up use the dmesg command. It displays what hardware was detected and which drivers were loaded by the kernel at boot time and also the messages generated by the kernel.

2**. journalct1 command**: Journalct1 command is also used to show the messages associated with a particular boot instance. The output contains linux kernel version, information about the computer (eg : Dell workstation), kernel command line options, type of processors (eg : Intel Xeon), number of CPUs, serial ports, mouse port, CD drive, network interface card (eg : e1000) and parellel port.

3. **lspci command:** The lspci command lists PCI buses on your computer and devices connected to them. (A Peripheral Component Interconnect Bus (PCI bus) connects the CPU and expansion

boards such as modem cards, network cards and sound cards.) The output contains the host bridge which connects the local bus to the other components on the PCI bridge, sound (audio device), flash drives, other USB devices, video display and wired network cards (Ethernet controller). If you have trouble in getting any of these devices to work, note the model names and numbers given. To get detailed output from lspci use -v, -vv, -vvv.

4. **lsusb command:** The lsusb command lists information about the computer's USB hubs along with any USB devices connected to the computers USB ports.

5. **lscpu command**: To see details about your processor, run the lscpu command. This command gives basic information about your computers processors. The output contains architecture (64 bit system), CPU operate modes (32 bit 64 bit), number of CPUs etc.

**Checking processes:** Linux provide some commands that allow users to monitor processes and system resource usage on Linux

1. **sysstat**: to find Linux CPU utilization.

**2. mpstat:** Display the utilization of each CPU individually if you are using SMP (Multiple CPU) system.

**Output:**

```
Linux 2.6.15.4 (debian)        Thursday 06 April 2006
 05:13:05  IST  CPU   %user   %nice    %sys %iowait    %irq   %soft  %steal   %idle    intr/s
05:13:05  IST   all    16.52    0.00    2.87    1.09    0.07    0.02    0.00   79.42   830.06
```

3. **sar**: Report CPU utilization by displaying todays CPU activity.

Output:

```
Linux 2.6.9-42.0.3.ELsmp (dellbox.xyz.co.in)        01/13/2007
12:00:02 AM       CPU  %user  %nice  %system  %iowait    %idle
12:10:01 AM       all     1.05   0.00     0.28         0.04         98.64
12:20:01 AM       all     0.74   0.00     0.34         0.38         98.54
...
Average:          --- --       --       ----           ----          -----
```

**4. iostat :** This command reports Central Processing Unit (CPU) statistics and input/output statistics for devices and partitions. It can be use to find out your system's average CPU utilization since the last reboot.

Output:

```
Linux 2.6.15.4 (debian)        Thursday 06 April 2006
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
   16.36    0.00    2.99    1.06    0.00   79.59
Device:            tps   Blk_read/s   Blk_wrtn/s   Blk_read   Blk_wrtn
hda               0.00         0.00         0.00         16          0
hdb               6.43        85.57       166.74     875340    1705664
hdc               0.03         0.16         0.00       1644          0
sda               0.00         0.00         0.00         24          0
```

5. **top command** : The top command allows users to monitor processes and system resource usage on Linux. It is one of the most useful tools in a sysadmin's toolbox. It starts up an interactive command line application. The top command's output is divided into two different sections.The upper half of the output contains statistics on processes and resource usage. The lower half contains a list of the currently running processes. You can use the arrow keys and Page Up/Down keys to browse through the list. If you want to quit, press "q".

The upper half contains :

System time

Uptime

Number of active users

Memory usage - total, free and used - RAM and swap space

Tasks - total number of processes, number of running processes, number of sleeping processes, number of stopped processes and the number of zombies ( terminated child processes whose data structures are still around ).

CPU usage - The CPU usage section shows the percentage of CPU time spent on various tasks. (**us -** userspace, **sy** - kernelspace, **nice** - priority of a process, **id -**remains idle, **wa** - waiting time, **hi** - time spent on handling hardware interrupt, **si** - time spent on handling software interrupt, **st** - steal time (CPU is busy on some other VM)

Load average - the average load over one, five and fifteen minutes. "Load" is a measure of the amount of computational work a system performs.

```
top - 15:39:37 up 90 days, 15:26,  2 users,  load average: 0.00, 0.00, 0.00
Tasks:  27 total,   1 running,  26 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem :   524288 total,    22792 free,   119380 used,   382116 buff/cache
KiB Swap:   131072 total,    43716 free,    87356 used.   322002 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
    1 root      20   0   37168   1192    680 S   0.0  0.2   2:21.51 systemd
    2 root      20   0       0      0      0 S   0.0  0.0   0:00.00 kthreadd/646
    3 root      20   0       0      0      0 S   0.0  0.0   0:00.01 khelper/646
   62 root      20   0   38896   1316   1188 S   0.0  0.3   1:08.85 systemd-journal
  219 root      20   0   26012    328    200 S   0.0  0.1   0:11.09 cron
  226 root      20   0   65464    924    220 S   0.0  0.2   0:13.11 sshd
  229 syslog    20   0  184632   1524    464 S   0.0  0.3   0:28.85 rsyslogd
  231 root      20   0   47572    504     40 S   0.0  0.1   0:07.80 rpcbind
  274 root      20   0   12788      8      4 S   0.0  0.0   0:00.00 agetty
  275 root      20   0   12788      8      4 S   0.0  0.0   0:00.00 agetty
  293 root      20   0  308984  12800   2248 S   0.0  2.4  27:15.96 fail2ban-server
 4452 root      20   0   92996   3124   3120 S   0.0  0.6   0:00.03 sshd
 4461 supriyo   20   0   92996   1000    996 S   0.0  0.2   0:00.00 sshd
 4462 supriyo   20   0   19472   1604   1600 S   0.0  0.3   0:00.05 bash
 4696 root      20   0   92996   4036   3132 S   0.0  0.8   0:00.02 sshd
 4705 supriyo   20   0   92996   1952   1008 S   0.0  0.4   0:00.02 sshd
 4706 supriyo   20   0   19472   3364   1600 S   0.0  0.6   0:00.05 bash
 4718 supriyo   20   0   36608   1784   1324 R   0.0  0.3   0:00.31 top
 5830 root      20   0   41532    728    320 S   0.0  0.1   0:01.25 systemd-udevd
13879 www-data  20   0  290032   2632   2612 S   0.0  0.5   0:01.18 php-fpm7.0
14031 cloud-t+  20   0   19788   9736   3276 S   0.0  1.9  10:11.27 cloud-torrent
14089 root      20   0  286060    560    452 S   0.0  0.1   1:11.67 php-fpm7.0
14091 www-data  20   0  289508   2168   2064 S   0.0  0.4   0:02.32 php-fpm7.0
```

**6. htop :** htop has some added features that make easier sorting by different factors, killing processes quicker and some display options look better. You can access its options using F1-10 as indicated on the bottom of the terminal.

# File Security & Permissions

**File security:** Linux file security is simple in design, but effective in controlling access to files and directories. The tasks that affect system security can only be managed by the root user. System Administrator manage the following features:

1. File Systems: The Directory structure is set up when you first install the Linux. If users later want to change the file system outside their home directory, they need administrative privilege. A root user has permission to access file owned by any user. A root user can copy, move, or change any user's file. Thus system administrator can make bake up copies of the file system for safe keeping.
2. Software installations: A user need root privilege to install software because malicious software can harm the system.
3. User Accounts: Only the root user can add or remove user accounts and group accounts.
4.Servers: Configuring web servers, file servers, domain name servers, mail servers, etc requires root privilege. If someone wants to crack a service, they can not get root privilege, to such services and system resources.
5. Security: Security features, such as firewalls nad user access lists are set up with root privilege. The root user should should monitor the usage of services, and server resources against abusal and exhausting.

**Permissions:**

The permission types for file/directory access are read, write, and execute. The command used for viewing the Permissions is " *ls –l* ".

The output contains : **-rwxrwxrwx 1 owner: group**

The first character underscore denoted whether it is a file or directory(f/d). The following set of three characters (rwx) is for the owner permissions. The second set of three characters (rwx) is for the Group permissions. The third set of three characters (rwx) is for the All Users permissions. The following integer number gives the number of hard links to the file. The chmode command is used to modify the permissions.

**Securing the password files:**

The /etc directory contains most of the basic Linux system configuration files.
The **/etc/passwd** file is the file in the linux system used to check user account information. It has the following permission settings: -rw-r--r-- . Users are not able to modify the /etc/passwd directly.

The **/etc/shadow** file has the appropriate settings: ----------.
For the /etc/shadow file, there is no access permission for the user. User running passwd command temporarily becomes root while the command is executing in memory and can then write to the /etc/shadow file. But, only to change the users on passwod related information.

The **/etc/group** file contains all the groups under the linux system. Its file permissions is rw-r--r--.

The group password file **/etc/gshadow** is properly secured with permissions ---------
The root user is all powerful and has complete access to all files, whether the permissions are listed or not.

**/etc/security** directory contains files that set a variety of default security conditions for your computer, basically defining how authentication is done. These files are part of the pam (pluggable authentication modules) package.

## Managing dangerous file system permissions, for securing the file system

**SUID** is a special file permission for executable files which enables other users to run the file with effective permissions of the file owner. Files with the SUID permission in the owner category and execute permission in the other category allow anyone to termporarly become the files owner. The riskiest case is that if the files owner is root.

**SGID** is a special file permission for executable files and enables other users to inherit the effective **GID** of file group owner. Files with the SGID permissions in the owner category and executive permission in the other category allow anyone to temporarily become a group member of the files group. SGID can also be set on directories. This sets the group ID of any files created in the directory to the group ID of the directory. Executable files with SUID and SGID are favorites of malicious users. Thus it is best to use them restrictedly.

Commands such as passwd and sudo are designed to be used as SUID programmes. Even though these commands run as root user, a regular user can change his own password with passwd.

A more dangerous situation would be if a hacker created SUID command, anyone running that command can effectively change everything on the system that has root access.

Using the **find** command, it can be searched to see whether there are any hidden or inappropriate SUID and SGID commands on the system. Instead of the normal `x` which represents execute permissions, a `s` (to indicate **SUID**) special permission is displayed.

Eg : find -ls.

### Setting permissions :

For managing users, access control lists (ACL) feature is used. Regular users can share their files and directories selectively with other users and groups.

1. **setfacl command** : setfacl command is used to set ACLs. A user can allow others to read, write and execute files and directories without requiring the root user to change the user or group.

Syntax: setfacl -m u:username:rwx file name.

To modify permissions -m option is used. To remove permissions -x option is used.

Eg: To add permission for other users and groups on a file.

setfacl -m u:bill:rw /tmp/memo.txt

setfacl -m g:sales:rw /tmp/memo.txt

2. **getfacl command** : The getfacl command is used to display the permissions set by setfacl command.

Eg: getfacl /tmp/memo.txt

The permissions in output of ls -l /tmp/memo.txt is - rw-rw-r--+

In the output the + sign indicates that ACLs are set on the file.

Setting default ACLs on a directory enables the ACLs to be inherited.

# Becoming Superuser

The person who manage all the Linux system resources is called system administrator. The username of the system administrator is *root*. The root user has complete control of the operation of the linux system. That user can open any file or run any programme. The root user also installs software packages and adds accounts for other people who use the system. Many administrative tasks and their associated commands require superuser status.

When you first install a linux systems, you add a password for the root user. You must remember and protect this password. You need it to login as root or to obtain root permission while you are logged in as some other user.

There are two ways to become the superuser. The first is to log in as *root* directly. The second way is to execute the command `su` while logged in to another user account. The `su` command may be used to change one's current account to that of a different user after entering the proper password.

After you have logged in as root, the home directory for the root user is /root. Information associated with the root user account are located in the /etc/passwd file.

Example : root:x:0:0:root:/root:/bin/bash

This shows that user name is root, the user ID is 0, the group ID is 0, home directory is /root, shell for the user is /bin/bash. Linux uses /etc/shadow file to store encrypted password data. So the password filed is shown as x.

At this point, any command you run from your shell is run with root previlege. Type exit or Ctrl + D when you are finished. Two methods to become a super user are:

1. log in as superuser on the system console.

```
hostname console: root
Password: root-password
#
```

*The pound sign (#) is the Bourne shell prompt for the superuser account. This method provides complete access to all system commands and tools.*

2. Log in as a user, and then change to the superuser account by using the `su` command at the command line.

```
$ su
Password: root-password
#
```

This method provides complete access to all system commands and tools.

It takes the username corresponding to the desired account as its argument; *root* is the default when no argument is provided.

After you enter the `su` command (without arguments), the system prompts you for the *root* password. If you type the password correctly, you'll get the normal root account prompt ( default #), indicating that you have successfully become superuser and that the rules normally restricting file access and command execution do not apply. Options are : -c, -p, -l, -s

Examples:

```
1. $ su
Password:
```

You may exit from the superuser account with `exit` or Ctrl-D.

2. Login into another user account

```
 su guest
Password:
```

## 2. Run specific command with another user privilege

```
      Used to switch from one account to another. User will be prompted for the
password of the user switching to. User can also use it to switch to root
account.
$ su guest -c date
```

```
Password:
Sat Jan  8 11:18:12 IST 2011
```

To preserve the current environment use the following command:

```
$ su -p guest
```

OPTIONS:

| | |
|---|---|
| **-c C***OMMAND* | Specify a command that will be invoked by the shell using its **-c**. |
| **-**, **-l** | Provide an environment similar to the user logged in directly. |
| **-s** *SHELL* | The shell that will be invoked. |

**3. Gaining Administrative Access with sudo**:

Users can be given administrative permissions for particular tasks by typing sudo followed by the command they want to run, without giving the root password.

**sudo (** super user do ) allows a user with proper permissions to execute a command as another user such as super user. ( according to the specifications in the /etc/sudoer file)

eg: sudo ls /usr/local/protected to get file listing of an unreadable directory.

*Many administrative tasks and their associated commands require superuser status.*

*Examples:*

on a Debian or Ubuntu Linux using apt-get command:

# sudo apt-get install sysstat

on RHEL/CentOS Linux v4.x or older:

# sudo up2date install sysstat

on RHEL/CentOS Linux v5.x+ or newer:

# sudo yum install sysstat

# Disk partitions and size

To install Linux operating system on a hard disk, it must be subdivided into distinct storage units. Those storage units are called partitions. Under the MBR partitioning scheme (default), Partitions can be of type **primary** (maximum four), **extended** (maximum one) or **logical** (contained within the extended partition).

Partitions can be of different sizes, and different partitions may have different filesystems on them. So a single disk can be used for many purposes. A single disk can be used for multiple operating systems. For example, different Linux distributions and Windows operating system can sharing one hard disk.

Partition information is stored in a partition table on the disk. The table contains information about the start and end of each partition, information about its type, and whether it is bootable or not. The **partition table** is written in the **master boot record** or **mbr**.

The standard partitions scheme for most Linux installations is as follows:

- A 12-20 GB partition for the OS, which gets mounted as / (called "root")
- A smaller partition used to augment your RAM, mounted and referred to as swap (equal to RAM size or double the RAM size (Swap partition ID is 82)
- A larger partition for personal use, mounted as /home ( Linux native partion ID is 83)

A hard drive in Linux system is represented by /dev/sda, /dev/sdb, /dev/sdc, … etc. The "dev" is short for device, and in this case, a block storage device. The "sd" is short for SCSI mass-storage driver. (SCSI stands for Small Computer System Interface.) The commands used to create and delete partitions are fdisk, gdisk, and parted.

**fdisk command**: (fixed disk or format disk) It is a text-based **command-line** utility for viewing and managing hard disk partitions on Linux. fdisk command can be used to view, create, resize, delete, change, copy and move partitions. Single-letter command set of fdisk command are used to work with the partitions. Some of them are:

d delete a partition
l list known partition types
m print the menu of options
n add a new partition
p print the partition table
q quit without saving changes
t change a partition's system id
v verify the partition table
w write table to disk and exit

**Creating a New Disk Partition with Specific Size :**

To add a new storage medium (hard disk, usb flash drive etc.) to the computer, we should:

1. connect the device to the computer
2. Partition the new disk using fdisk command
3. Create the file systems on the new disk
4. Mount the file systems

To create a new partition, the steps are:

1. Enter the following command to enter/view the disk
   sudo fdisk /dev/sdb

The fdisk command mode is displayed as follows:
  Command (m for help):

2. Use p to view all partitions
3. Use d to delete existing partitions
4. Use n to create new partition. Now type of partition should be chosen from extended (e) or primary partition (p).
5. Type the letter p to choose primary.
6. Type the partition number. For creating the first partition, type the number 1.
7. Now we have to give first sector to start the partition. Select the first available sector number.
8. Enter the size of the partition.
9. The last sector value can be selected.
10. To make other partitions, give partition number , 2,3 etc with starting sector number.
11. Type w to make changes to the partition table permanent

After the creation of partition, format it with a file system.
   sudo mkfs -t ext4 /dev/sdb

Create a mount point and mount it to the partition.
  mkdir /mnt/test
  mount /dev/sda /mnt/test

To view the Size of the new or existing Partition use the following command:
  sudo fdisk -s /dev/sdb

If you have multiple partitions, you need to watch the percentage of space used on each mounted partition. If, for example, space runs out on a separate /var file system, programs that need to spool data (such as mail and printing utilities), write to log files in /var/log, or use temporary file space in /var/tmp may fail. Even if plenty of space is available in the root partition or another partition, if the assigned partition runs out of space, it won't draw from other partitions.

# Getting System Information

**1. uname command**: It is one of the most useful commands to display basic information about the Linux system. It is used to find out the hostname of the system, the hardware architectures supported by the currently used kernel and the exact release of the system.

Syntax: uname [OPTION]...

**Options:**

-a Prints all information
-s Print the kernel name.
-n Print the network node hostname.
-r Print the kernel release.
-v Print the kernel version.
-m Print the machine hardware name.
-p Print the processor type, or "unknown".

**examples:**

1) uname –a
2) uname : It is same as **uname –s**

***2. hostname* command:** It is used to show or set the system's host name. Host names are used by many networking programs to identify the machine in the network. DNS name means Domain Name System name, NIS domain name means Network Information System domain name.

**Syntax :** hostname [OPTION]...

**Options:**

-a Display the alias name of the host.
-b set a hostname
-d Display DNS domain name.
-h Print help message and exit.
-i Displays network ip address of the host
-s Displays short host name. This displays the hostname until the first dot.
-V Print version information on standard output and exit successfully.
-v Be verbose
-y Displays NIS domain name. You can also set a new NIS domain using this option.

**Examples:**

1) hostname : to print host name output : testserver.example.com
2) hostname –i : *output:* 192.168.134.128
3) hostname –d : To print the domain name *output:* example.com
4) hostname -s : To print short hostname *output:* testserver

**3. Users command**: The users command displays login names of users currently logged in on system. It displays a blank-separated list of user names of users currently logged in to the current host, on a single line.

**Examples:**

1) users
2) users -- help
3) users –version

4. The kernel : The heart of the Linux system is called the kernel. The kernel provides the interface between you (and the programs you run) and the hardware (hard disks, RAM, network cards, etc.). Using the /proc file system, we can find out a lot of information about the kernel, by simply displaying the contents of /proc files.

For each process currently running in Linux, there is a directory in /proc consisting of the process number for the running process. (Type ps aux | more to see the running processes and their associated PID numbers.) The /proc directory contains other files that are connected to certain features (such as networking, SCSI devices,and other components).

To display the contents of /proc files, you can use the cat command. For example, change to the /proc directory (cd /proc), then type the following command:

cat version

The output of this command contains the Linux version number and other information (such as the compiler version and the system install date). There are other files under the /proc directory structure that you can also list information about your running Linux system. Some files that we can "cat" to get information are:

cpuinfo — Tells you the type of CPU in your computer, the speed (CPU MHz), the CPU family, and other information related to your computer's processor.

devices — Displays the character and block devices currently being used on your computer, along with their major device numbers.

ioports — Shows the I/O port addresses for the devices on your computer.

meminfo — Contains information about memory usage and swap space usage. You can see the total amounts of memory and the how much is currently being used.

modules — Shows a list of modules that are currently installed in the system.

mounts — Displays the file systems that are currently mounted in the system.

partitions — Contains the names of your hard disk partitions, the number of blocks in each partition, and each partition's major and minor device number.

pci — Lists the PCI devices installed in your computer. You can see the bus device numbers, names, and other information. For cards that are installed (such Ethernet or modem cards), you can see their IRQs, addresses, and other information.

swaps — Shows the swap partitions that are currently mounted on your system, along with their sizes and the amount of space being used.

net/dev — Displays the contents of the net/dev file to see your active network interfaces.

sys/* — Looks at the contents of these directories for information related to debugging (debug),devices (dev), file systems (fs), the kernel (kernel), networks (net), and processes (proc).