**Disk Partition [ fdisk, cfdisk, parted, gparted, lvm ]**

**Introduction**

Disk partitioning is an essential step in managing storage efficiently, allowing me to organize the hard drive into separate sections for different purposes. Tools like fdisk and cfdisk offer command-line simplicity, while parted provides advanced options for resizing and creating partitions. For a more visual approach, gparted makes things even easier with its graphical interface.

**Fdisk**

Fdisk is a dialog-driven program for creation and manipulation of partition tables, easy to create and delete partitions and has various options in it.

*fdisk <device name> (or) fdisk /dev/sda*

**Working**

1. When you type fdisk, the shell looks for the binary in your $PATH directories.
2. fdisk opens the disk device file (e.g., /dev/sda) using system calls like open() to access the disk.
3. It reads the partition table by directly accessing the disk's first few sectors.
4. When you create or delete a partition, it modifies the in-memory partition table and writes changes back to the disk using write() and related system calls.
5. The changes are finalized when the user writes the partition table, notifying the kernel using ioctl() to re-read the partition table.

**Options :**

m – To list help menu.

p – To print the partition.

n – To create new partition.

d – To delete the partition.

t – To select type of the partition.

w – To write changes to disk.

q – To exit the fdisk.

g – To create new empty GPT partition table.

o – To create new empty MBR (Dos) partition table.

F – To list free unpartitioned space .

v – To verify the partition table.

l – To list available partition types.

**Note :**

**Cfdisk**

Cfdisk is a curses-based program for manipulating any block device, easy to create, delete and resize partitions not the filesystem.

*cfdisk <device name> or cfdisk /dev/sda*

**Working**

1. Like fdisk, the binary is located and executed by the shell.
2. cfdisk uses the ncurses library to render its text-based UI.
3. It interacts with the disk device file and partition table in the same way as fdisk (using open(), read(), and write()).
4. Changes to the partition table are made in memory and written back only when the user confirms.

**Options :**

b – To add bootable flag on selected partition.

n – To create new partition.

h – To print the help menu.

d – To delete selected partition.

W – To write changes to disk.

r – To resize the disk partition (only the disk not the filesystem).

q – To exit cfdisk.

Up Arrow – To move the cursor to previous partition.

Down Arrow – To move the cursor to next partition.

Left Arrow – To move the cursor to previous menu item.

Right Arrow – To move the cursor to next menu item .

**Note :**

**Parted**

Parted is a program to manipulate disk partitions, easy to create partitions, deleted partition, resize partitions and copy data to new hard drives and more.

*parted <device name> (or) parted /dev/sda*

**Working**

1. The shell finds the parted binary and starts the process.
2. parted opens the disk device file (e.g., /dev/sda) using low-level system calls.
3. It reads the GPT or MBR structures from the disk, parses them, and loads them into memory.
4. Commands like creating or resizing partitions modify the in-memory partition table, which is written back when changes are confirmed.

**Options :**

help – To display help menu / display help about below commands.

*help <command> (or) help mktable*

mkpart – To create a new disk partition.

mktable – To create new partition table (GPT, MBR, etc).

*mktable <type> (or) mktable gpt*

rm – To delete a partition.

*rm <partition number> (or) rm 1*

print – To print partition details

*print device (or) print free (or) print list (or) print all*

quit – To exit parted program

resizepart – To resize given partition

*resizepart <partition number> <size> (or) resizepart 1 10G*

**Note :**

**Gparted**

Gparted application is a GNOME partition editor for creating, reorganizing and deleting disk partitions as well as reducing the risk of loss of data.

**Working**

1. When launched, gparted initializes a graphical interface using the GTK library.
2. It scans all connected storage devices by executing low-level commands (lsblk, parted) or reading files like /proc/partitions and /sys/block/.
3. For any operation (e.g., resizing a partition), it spawns backend tools (parted, mkfs) to execute the requested changes.
4. It updates the GUI in real time to reflect disk modifications.

**Note :**

**LVM**

The Logical Volume Manager provides tools to create logical block devices from physical devices. **Volume Group** is a collection of one or more physical devices. Each of these physical devices are called **Physical Group**. A **Logical Group** is a virtual block device that can be used by the system or applications.

**Working**

1. The lvm command interacts with the device-mapper kernel module.
2. It reads metadata stored on physical volumes (PVs) and volume groups (VGs).
3. LVM maintains a mapping of logical to physical volumes. When you create or resize a logical volume, it updates this mapping in the metadata.
4. It uses ioctl() system calls to notify the kernel of changes to the logical volumes.

**Steps to create LVM**

**1. Create Physical Volume**

*pvcreate <device name> (or) pvcreate /dev/sda*

**Physical Volume Commands**

pvdisplay –

*pvchange –*

*pvremove –*

*pvresize –*

*pvck –*

*pvs –*

*pvscan –*

*pvmove –*

**2. Create Volume Group**

*vgcreate <vg name> <physical volumes > (or)*

*vgcreate volgrp0 /dev/sda3 /dev/sdb*

**Volume Group Commands**

vgdisplay –

vgck –

vgextend –

vgrename –

vgremove –

vgreduce –

vgs –

vgscan –

vgmerge –

vgchange –

vgsplit –

**3. Create Logical Volumes**

*lvcreate --size <size> --name <lv name> <vg name> (or)*

*lvcreate --size 5G --name lv\_root volgrp0*

**Logical Volume Commands**

lvdisplay –

lvs –

lvscan –

lvextend –

lvreduce –

lvrename –

lvremove –

lvresize –

lvchange –

**Note :**

* logical volumes are represented in the device mapper directory (/dev/mapper)

**4. Create Filesystem**

*mkfs.<filesystem type> <device name> (or)*

*mkfs.ext4 /dev/mapper/lv\_root*

**5. Mounting Logical Volume**

*mount <device name> <target directory> (or)*

*mount /dev/mapper/lv\_root /logical\_vol\_folder*