**[fdisk to Manage Partitions on Linux](https://www.howtogeek.com/106873/how-to-use-fdisk-to-manage-partitions-on-linux/)**

<https://www.howtogeek.com/106873/how-to-use-fdisk-to-manage-partitions-on-linux/>

fdisk command is a text-based utility for viewing and managing hard disk partitions.

mkfs command to format new partitions.

Both are used as root user or prefixed with **sudo**

**For listing the present partitions:**

**fdisk –l**

**If we use disk name with this, it will list partitions of this disk only**

sudo fdisk -l /dev/sda

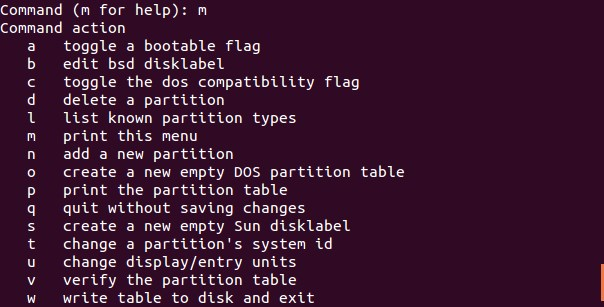
Command Mode:

To work on disks we use command mode. Use fdisk –l to get name and then

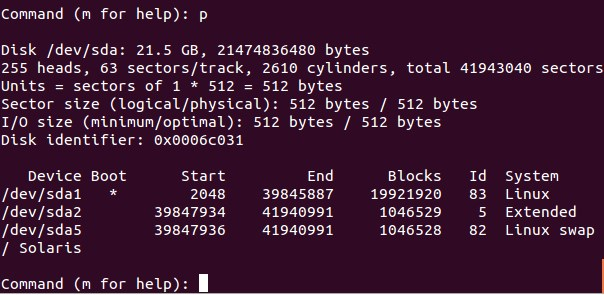
Example: fdisk /dev/sda to enter command mode for this disk

In command mode we use single alphabet command for actions we want to perform, provided in command mode

m :- use ‘m’ to get list of all alphabets with different powers



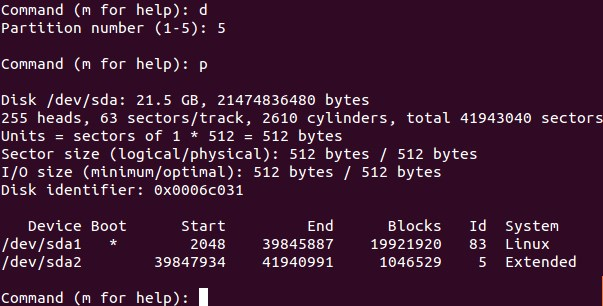
p :- is used to print partition table of disk we selected in command mode.

****

Here we hae 3 partitions in sda – 1,2 and 5

d :- is used to delete partition

When we use‘d’ it will ask partition number you want to delete. Suppose we gave 5 then it will delete partition number 5 and only 1 and 2 will remain.



n :- is used to create partitioncd

after it we get two options, use

l : to create logical partition and

p : to create primary partition

After giving response it we have to specify sector of the disk we want the partition to start at. By default system will give you available sector as it is aware of other partitions. Just press ENTER to go with default.

After it provide partition number and size to it. (Example: +10M for 10 MB and +1G for 1 GB).

**Exiting:**

Use **w** to write the changes you’ve made to disk.

Use **q** if you want to quit without saving changes.

**Formatting partition:**

mkfs is used to do so

Suppose we want to format 2nd partition of first disk with ext4 file system, then

mkfs.ext4 /dev/sda2

Extended partitions:

Difference between primary and extended:-

– The original partitioning scheme for PC hard disks allowed only four partitions, called primary partitions.  
– To create more than four partitions, one of these four partitions can be divided into many smaller partitions, called logical partitions. When a primary partition is subdivided in this way, it is known as an extended partition.

SDA/SDB/SDC:

<http://linuxbsdos.com/2014/11/08/a-beginners-guide-to-disks-and-disk-partitions-in-linux/>

Suppose in windows pc we have one external HD attached, so it will show 2 disk, Disk 0 (PC HD boot drive) and Disk 1 (External HD) so in terms of linux disk 0 is ‘sda’ disk 1 is ‘sdb’ and if we attach one more it will be ‘sdc’ and so on.,

Mounting partitions as folders:

<https://linuxexpresso.wordpress.com/2010/03/14/mount-partitions-in-terminal-fstab/>

Now, create a mount point for your drive. You can make it whatever you want, but a common place to mount things is in the /mnt directory. Create a folder there with

sudo mkdir mount\_name

mount the drive to the folder

sudo mount -t auto -v /dev/sdb1 /mnt/mount\_name

for us it will be sudo mount -t ext4 -v /dev/sdb1 /mnt/mount\_name (as we already formatted them with ext4)

To Unmount:

sudo umount /dev/sdb1

Make LVM partition:

<https://www.digitalocean.com/community/tutorials/how-to-use-lvm-to-manage-storage-devices-on-ubuntu-16-04>

### Creating Physical Volumes From Raw Storage Devices

First, use the lvmdiskscan command to find all block devices that LVM can see and use:

* sudo lvmdiskscan

To mark the storage devices as LVM physical volumes, use pvcreate. You can pass in multiple devices at once:

* sudo pvcreate /dev/sda /dev/sdb

### Creating a New Volume Group from Physical Volumes

To create a new volume group from LVM physical volumes, use the vgcreate command. You will have to provide a volume group name, followed by at least one LVM physical volume:

* sudo vgcreate volume\_group\_name /dev/sda

### Creating a Logical Volume by Specifying Size

To a logical volume from a volume group storage pool, use the lvcreate command. Specify the size of the logical volume with the -L option, specify a name with the -n option, and pass in the volume group to allocate the space from.

For instance, to create a 10G logical volume named test from the LVMVolGroup volume group, type:

* sudo lvcreate -L 10G -n test LVMVolGroup

### Growing the Size of a Logical Volume

One of the main advantages of LVM is the flexibility it provides in provisioning logical volumes. You can easily adjust the number or size of volumes on the fly without stopping the system.

To grow the size of an existing logical volume, use the lvresize command. Use the -L flag to specify a new size. You can also use relative sizes by adding a "+" size. In that case, LVM will increase the size of the logical volume by the amount specified. To automatically resize the filesystem being used on the logical volume as well, pass in the --resizefs flag.

To correctly provide the name of the logical volume to expand, you'll need to give the volume group, followed by a slash, followed by the logical volume:

* sudo lvresize -L +5G --resizefs LVMVolGroup/test

### Reducing the Size of a Logical Volume

To shrink a logical volume, you should first back up your data. Because this reduces the available capacity, mistakes can lead to data loss.

When you are ready, check on how much space is currently being used:

* df -h

Output

Filesystem Size Used Avail Use% Mounted on

. . .

/dev/mapper/LVMVolGroup-test 4.8G 521M 4.1G 12% /mnt/test

In this example, it looks like a little over 521M of the space is currently in use. Use this to help you estimate the size that you can reduce the volume to.

Next, unmount the filesystem. Unlike expansions, filesystem shrinking should be performed when unmounted:

* cd ~
* sudo umount /dev/LVMVolGroup/test

After unmounting, check the filesystem to ensure that everything is in working order. Pass in the filesystem type with the -t option. We'll use -f to check even when the filesystem appears okay:

* sudo fsck -t ext4 -f /dev/LVMVolGroup/test

After checking the filesystem, you can reduce the filesystem size using the filesystem's native tools. For Ext4 filesystems, this would be the resize2fs command. Pass in the final size for the filesystem:

Warning: The safest option here is to choose a final size that is a fair amount larger than your current usage. Give yourself some buffer room to avoid data loss and ensure that you have backups in place.

* sudo resize2fs -p /dev/LVMVolGroup/test 3G

Once the operation is complete, resize the logical volume by passing the same size to the lvresize command with the -L flag:

* sudo lvresize -L 3G LVMVolGroup/test

You will be warned about the possibility of data loss. If you are ready, type y to proceed.

After the logical volume has been reduced, check the filesystem again:

* sudo fsck -t ext4 -f /dev/LVMVolGroup/test

If everything is functioning correctly, you can remount the filesystem using your usual mount command:

* sudo mount /dev/LVMVolGroup/test /mnt/test

### Removing a Logical Volume

If you no longer need a logical volume, you can remove it with the lvremove command.

First, unmount the logical volume if it is currently mounted:

* cd ~
* sudo umount /dev/LVMVolGroup/test

Afterwards, remove the logical volume by typing:

* sudo lvremove LVMVolGroup/test

You will be asked to confirm the procedure. If you are certain you want to delete the logical volume, type y.

### Removing a Volume Group

To remove an entire volume group, including all of the logical volumes within it, use the vgremovecommand.

Before you remove a volume group, you should usually remove the logical volumes using the procedure above. At the very least, you must make sure that you unmount any logical volumes that the volume group contains:

* sudo umount /dev/LVMVolGroup/www
* sudo umount /dev/LVMVolGroup/projects
* sudo umount /dev/LVMVolGroup/db

Afterwards, you can delete the entire volume group by passing the volume group name to the vgremove command:

* sudo vgremove LVMVolGroup

You will be prompted to confirm that you wish to remove the volume group. If you have any logical volumes still present, you will be given individual confirmation prompts for those before removing.

### Removing a Physical Volume

If you wish to remove a physical volume from LVM management, the procedure you will need depends on whether the device is currently being used by LVM.

If the physical volume is in use, you will have to move the physical extents located on the device to a different location. This requires the volume group to have enough other physical volumes to handle the physical extents. If you are using more complex logical volume types, you might have to have additional physical volumes even when you have plenty of free space in order to accommodate the topology.

When you have enough physical volumes in the volume group to handle the physical extents, move them off of the physical volume you wish to remove by typing:

* sudo pvmove /dev/sda

This process can take awhile depending on the size of the volumes and the amount of data to transfer.

Once the extents have been relocated to peer volumes, you can remove the physical volume from the volume group by typing:

* sudo vgreduce LVMVolGroup /dev/sda

This will remove the vacated physical volume from the volume group. After this is complete, you can remove the physical volume marker from the storage device by typing:

* sudo pvremove /dev/sda