Are you someone who is frequently running out of storage space on your linux machine? Well, adding a hard disk to your linux machine was never so easy! Just follow these simple steps to add a new hard disk.

**Step 1 − Physically connect the hard disk**

* Power off your linux system
* Connect new hard disk to appropriate data and power cord
* Turn on the system again

It’s time to play with linux machine now. Login to your system and get access to superuser. Run system terminal. The first command to shoot is −

**$ fdisk -l**

This command should list all the disks available with the system. The output of the command should look like this −

Disk /dev/sda: 8589 MB, 8589934592 bytes

255 heads, 63 sectors/track, 1044 cylinders, total 16777216 sectors

Units = sectors of 1 \* 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk identifier: 0x000eeb83

Device Boot Start End Blocks Id System

/dev/sda1 \* 2048 12582911 6290432 83 Linux

/dev/sda2 12584958 16775167 2095105 5 Extended

/dev/sda5 12584960 16775167 2095104 82 Linux swap / Solaris

Disk /dev/sdb: 8589 MB, 8589934592 bytes

255 heads, 63 sectors/track, 1044 cylinders, total 16777216 sectors

Units = sectors of 1 \* 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk identifier: 0x00000000

**Disk /dev/sdb doesn't contain a valid partition table**

And we observe that currently it is currently showing one hard disk (**sda**) with three partitions. The newly added hard disk (**sdb**) is shown as unpartitioned. The last line of output says it all.

**Step 2 − Create Extended Partition**

We shall first format the drive using **fdisk** utility −

**$ fdisk /dev/sdb**

The command shows us some notification to be ignored and put us under a sub-command system, prompting for another command. We press ‘m’ here to see its full menu available.

Command action

a toggle a bootable flag

b edit bsd disklabel

c toggle the dos compatibility flag

d delete a partition

l list known partition types

m print this menu

**n add a new partition**

o create a new empty DOS partition table

p print the partition table

q quit without saving changes

s create a new empty Sun disklabel

t change a partition's system id

u change display/entry units

v verify the partition table

w write table to disk and exit

x extra functionality (experts only)

Command (m for help):

Okay! We proceed with option **n** which should allow us to create a new partition on the disk. Remember that we came this far by providing the disk name so all action are applicable to that disk only.

Partition type:

p primary (0 primary, 0 extended, 4 free)

**e extended**

Select (default p):

fdisk is asking which type of partition do we want to create on the disk. Primary partitions holds bootable images but we are not going to make it bootable. We choose here extended. Extended file format is itself a container of logical partitions, that we shall see here soon.

Select (default p): e

Partition number (1-4, default 1):

We shall choose default 1, for partition number.

First sector (2048-16777215, default 2048):

OMG! Its complicated! Just hit enter key to select default value of first sector i.e. 2048.

Using default value 2048

Last sector, +sectors or +size{K,M,G} (2048-16777215, default 16777215):

Again! Just press enter, let linux choose its default values and move on

Using default value 16777215

Command (m for help):

This will use some big default value and put us again in the fdisk-command prompt. All these changes which we (or system) has done are not applied on the physical hard disk yet. To write them on the hard disk we shall use the command ‘w’

Command (m for help): w

The partition table has been altered!

Calling ioctl() to re-read partition table.

Syncing disks.

The system is trying to convince us that partition information has been applied on the disc and synced with the system. To verify it we shall use the fdisk command again.

Disk /dev/sdb: 8589 MB, 8589934592 bytes

86 heads, 1 sectors/track, 195083 cylinders, total 16777216 sectors

Units = sectors of 1 \* 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk identifier: 0x4811713e

Device Boot Start End Blocks Id System

/dev/sdb1 2048 16777215 8387584 5 Extended

Now, fdisk has more information for us about the new hard disk. It is now showing the new hard disk formatted as Extended system. We cannot use extended file system for storage rather Extended partitions contain logical partitions and are accessible for storage purpose.

**Step 3 − Create Logical Partition**

Now we shall create a logical partition inside the Extended partition.

**$ fdisk /dev/sdb**

As history repeats itself, this command will put us back in the fdisk command prompt.

Command (m for help): m

Command action

a toggle a bootable flag

b edit bsd disklabel

c toggle the dos compatibility flag

d delete a partition

l list known partition types

m print this menu

n add a new partition

o create a new empty DOS partition table

p print the partition table

q quit without saving changes

s create a new empty Sun disklabel

t change a partition's system id

u change display/entry units

v verify the partition table

w write table to disk and exit

x extra functionality (experts only)

Command (m for help):

We shall choose here ‘n’ to create a new partition.

Partition type:

p primary (0 primary, 1 extended, 3 free)

l logical (numbered from 5)

Select (default p):

This time we are greeted with an option to choose logical partition. So hit on it and choose ‘l’.

Adding logical partition 5

First sector (4096-16777215, default 4096):

As being human, press ‘return’ key to choose default first sector.

Using default value 4096

Last sector, +sectors or +size{K,M,G} (4096-16777215, default 16777215):

Press ‘return’ again to default last sector value. The system will inform you that it is using default values and then prompt you to enter new command.

Using default value 16777215

Command (m for help):

Now we need to ‘w’rite the changes to be applied on hard disk in physical manner. Type ‘w’ and press return key.

The partition table has been altered!

Calling ioctl() to re-read partition table.

Syncing disks.

This time ‘fdisk -l’ should give a picture close to this one −

Device Boot Start End Blocks Id System

/dev/sdb1 2048 16777215 8387584 5 Extended

/dev/sdb5 4096 16777215 8386560 83 Linux

Now we have got our new hard disk partitioned.

**Step 4 − Format the partition**

Lets now format the disk partition make it accessible. We use **mkfs** command to format our new partition with ext3 filesystem.

**$ mkfs -t ext3 /dev/sdb5**

mke2fs 1.42.9 (4-Feb-2014)

Filesystem label=

OS type: Linux

Block size=4096 (log=2)

Fragment size=4096 (log=2)

Stride=0 blocks, Stripe width=0 blocks

524288 inodes, 2096640 blocks

104832 blocks (5.00%) reserved for the super user

First data block=0

Maximum filesystem blocks=2147483648

64 block groups

32768 blocks per group, 32768 fragments per group

8192 inodes per group

Superblock backups stored on blocks:

32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Allocating group tables: done

Writing inode tables: done

Creating journal (32768 blocks): done

Writing superblocks and filesystem accounting information: done

At this stage, our new disk is physically connected, partitioned and the partition has a file system.

**Step 5 − Mount it up!**

Our accessible new hard disk is at /dev/sdb5. To use it we need to mount it. Make sure that there exists a folder with ‘temp’ name (or whatever is you preferred) before you execute this command. Let’s mount and use it.

**$ mount -t ext3 /dev/sdb5 /usr/home/temp**

‘mount’ command shall help us mounting /dev/sdb5 to temp folder. Now we can access our newly installed & configured hard disk at **temp**.

**Step 6 − Cross Check!**

Its always good to cross-verify things. Though by now, you may see a folder mounted on your desktop or wherever you have pointed it to be. You can access it in GUI. But to verify in command prompt we execute the following command −

**$ df -h**

Filesystem Size Used Avail Use% Mounted on

/dev/sda1 5.8G 3.7G 1.9G 66% /

none 4.0K 0 4.0K 0% /sys/fs/cgroup

udev 991M 4.0K 991M 1% /dev

tmpfs 201M 948K 200M 1% /run

none 5.0M 0 5.0M 0% /run/lock

none 1001M 152K 1001M 1% /run/shm

none 100M 36K 100M 1% /run/user

/dev/sdb5 7.8G 19M 7.4G 1% /home/user/temp

Look at the last line!!! I told you its easy to install a new hard disk in linux system. We just did it, hence, proved!