

Partitions and Filesystems

Partitions:

The concept of breaking an object into multiple parts for better advantages is called Partitioning. Each part is called as partition.

In operating system level, we use same concept to break the hard disk into multiple parts for the below advantages.

Advantages:

1. Data Redundancy
2. Performance
3. Multiple Filesystems
4. Dual Booting
5. Better Data Management

To store the partitions information we have two partition tables. They are

1. MBR(Master Boot Record)
2. GPT(GUUID Partition Table)
GUUID----->Globally Unique User Identifier.

MBR:

The size of the MBR is 512 bytes which has been divided into 3 parts.

1. First part consists of boot loader information and size is 0-446 bytes.
2. Second part consists of partitions information and size is 64 bytes.
3. Third part is for MBR validation check and size is 2 bytes.

So, if we use MBR as a partition table we can create only four primary partitions because in MBR 64 bytes are reserved for storing partitions information(to store one partition information it needs 16bytes so $4 \times 16 = 64$ bytes).

If you want to create more than 4 partitions by using MBR as a partition table then in place of one primary partition we can create extended partition which allows use to create 11 more logical partitions.

We cannot use MBR for the hard disks which are more than 2TB.

GPT:

If we use GPT as partition table then we can create unlimited partitions but the limit depends on the operating system we are using.

If we are using windows or linux, we can create 128 partitions.

Naming Conventions For Hard Disks:

1. If we are using SCSI or SATA hard disks, then those hard disks are named as /dev/sda-----> scsi or sata disk A.

If we have more than one SCSI or SATA types of hard disks then the naming Conventions are

/dev/sda , /dev/sdb , /dev/sdc e.t.c.

And the partitions inside the hard disks will be named as

/dev/sda1 , /dev/sda2 , /dev/sd3 e.t.c.

2. If we are using IDE type of hard disks then the naming conventions are /dev/hda , /dev/hdb e.t.c

And the partitions will be named as /dev/hda1 , /dev/hda2 e.t.c

Creating partitions in MBR partition table labelled hard disks using FDISK :

#fdisk -l -----> to list all the hard disks information along with partitions information.

```

[root@server ~]# fdisk -l

Disk /dev/sda: 12.9 GB, 12884901888 bytes
255 heads, 63 sectors/track, 1566 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000bf3bf

   Device Boot      Start         End      Blocks    Id System
/dev/sda1  *           1           26       204800    83  Linux
Partition 1 does not end on cylinder boundary.
/dev/sda2                26        1046       8192000    83  Linux
/dev/sda3            1046        1177       1048576    82  Linux swap
/dev/sda4            1177        1567       3136512     5  Extended
/dev/sda5            1177        1241        512000    83  Linux
[root@server ~]# _

```

Or

#lsblk -----> to list all the hard disks information along with partitions

To create partitions in a particular hard disk, use the below syntax:

fdisk <hard-disk-name>

Ex: fdisk /dev/sda

#fdisk /dev/sda

```
susel:~ # fdisk /dev/sda
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):
```

As we are new to fdisk, we can take help from fdisk by typing m as shown in the above figure.

To print all the partitions information use p and the output will be like below

```
root@slackware1:/# fdisk /dev/sda
The number of cylinders for this disk is set to 1385.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
 1) software that runs at boot time (e.g., old versions of LILO)
 2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): p

Disk /dev/sda: 18.7 GB, 18737418240 bytes
255 heads, 63 sectors/track, 1385 cylinders
Units = cylinders of 16065 = 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1             1          125     1864831    83  Linux
/dev/sda2          126          612     3911827+    83  Linux
/dev/sda3           613          878     2838085    83  Linux
/dev/sda4           879         1385     2626627+    83  Linux

Command (m for help): t
Partition number (1-4): 1
Hex code (type L to list codes): L_
```

To create a partition, use the n option and it prompts for multiple values like below diagram

```
Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-130, default 1):
Using default value 1
Last cylinder, +cylinders or +size(K,M,G) (1-42, default 42): +100M
Command (m for help): p
```

Once you create the partition don't forget to save the partition. To save the partition use 'w'.

After that you will be out of fdisk prompt and next step is to update the changes to the kernel. By executing the below command we can update changes to kernel.

#partprobe ---->to update changes to kernel.

And repeat the same process to create partitions in MBR partition table.

Process for creation of Extended Partition:

```
[root@ip-172-31-21-159 ~]# fdisk /dev/xvdf
Welcome to fdisk (util-linux 2.23.2).

Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table
Building a new DOS disklabel with disk identifier 0x317da265.

Command (m for help): n
Partition type:
   p   primary (0 primary, 0 extended, 4 free)
   e   extended
Select (default p): e
Partition number (1-4, default 1):
First sector (2048-16777215, default 2048):
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-16777215, default 16777215):
Using default value 16777215
Partition 1 of type Extended and of size 8 GiB is set
Command (m for help):
```

Then save the changes by using 'w' and partprobe to update changes to kernel.

Process for Logical partition creation:

To create the logical partition please the below process

```
[root@ip-172-31-21-159 ~]# fdisk /dev/xvdf
Welcome to fdisk (util-linux 2.23.2).

Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Command (m for help): n
Partition type:
   p   primary (0 primary, 1 extended, 3 free)
   l   logical (numbered from 5)
Select (default p): l
Adding logical partition 5
First sector (4096-16777215, default 4096):
Using default value 4096
Last sector, +sectors or +size{K,M,G} (4096-16777215, default 16777215): +2G
Partition 5 of type Linux and of size 2 GiB is set

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
[root@ip-172-31-21-159 ~]#
```

Then save the changes by using 'w' and partprobe to update changes to kernel.

Deleting a partition:

Process to delete a partition.

```
[root@ip-172-31-21-159 ~]# fdisk /dev/xvdf
Welcome to fdisk (util-linux 2.23.2).

Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Command (m for help): d
Partition number (1,5, default 5): 5
Partition 5 is deleted

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
[root@ip-172-31-21-159 ~]#
```

Then save the changes by using 'w' and partprobe to update changes to kernel.

Creating partitions in GPT Partition table using parted command:

To list all hard disks information using parted command use the below syntax

parted -l and the output will be like below

```
[root@ip-172-31-21-159 ~]# parted -l
Model: Xen Virtual Block Device (xvd)
Disk /dev/xvda: 10.7GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags: pmbr_boot
```

Number	Start	End	Size	File system	Name	Flags
1	1049kB	2097kB	1049kB			bios_grub
2	2097kB	10.7GB	10.7GB	xfs		

To create a partition follow the below procedure.

```
[root@ip-172-31-21-159 ~]# parted /dev/xvdf
GNU Parted 3.1
Using /dev/xvdf
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) mkpart
Partition name? []? arjun
File system type? [ext2]?
Start? 0m
End? 2000m
Warning: The resulting partition is not properly aligned for best performance.
Ignore/Cancel? i
(parted) q
Information: You may need to update /etc/fstab.

[root@ip-172-31-21-159 ~]#
```

To save the changes use 'q' and to update changes to kernel execute partprobe command.

Repeat the same procedure to create partitions in GPT partition table.

Deleting a partition:

Follow the below procedure to delete a partition.

```
[root@ip-172-31-21-159 ~]# parted /dev/xvdf
GNU Parted 3.1
Using /dev/xvdf
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) p
Model: Xen Virtual Block Device (xvd)
Disk /dev/xvdf: 8590MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number   Start    End      Size      File system  Name   Flags
  1       17.4kB   2000MB   2000MB                        arjun

(parted) rm 1
(parted) q
Information: You may need to update /etc/fstab.

[root@ip-172-31-21-159 ~]#
```

To save the changes use 'q' and to update changes to kernel execute partprobe command.

```
[root@ip-172-31-21-159 ~]# partprobe
[root@ip-172-31-21-159 ~]#
```

Filesystem:

File system is used to store the data in an organized manner. There are number of filesystems available like ext, ext2, ext3, ext4 and xfs e.t.c

If you want to store the data once after creating partitions, we should format the partitions.

Syntax:

mkfs.<type of filesystem> <partition name>

or


```
mkfs -t <filesystem name> <partition name>
```

Mounting:

Mounting is nothing but attaching a directory to the filesystem to access the data exists in the filesystem or to store the data in the filesystem.

That directory which we use to mount the filesystem is called as mount point.

There are two ways of mounting the filesystems:

1. Temporary Mounting
2. Permanent Mounting

Temporary Mounting:

To mount the filesystem temporarily, use the below syntax.

Syntax:

```
mount <filesystem-name(partition name)> <mount point>.
```

use the below command to list all the mounted filesystems information.

```
#df -h
```

Use the below command to unmount the filesystem.

```
#umount <mount point> or <filesystem name>
```

the drawback with the temporary mounting is if you reboot the machine filesystem will get un-mounted.

Permanent Mounting:

To mount the filesystem permanently, we have to update the filesystem information in /etc/fstab file

Syntax:

```
# vi /etc/fstab
```

Go to the last line of the file and create a new line by pressing 'o' and provide filesystem details like below

<filesystem-name> <mount-point> <type of filesystem> <mount options>

<filesystem dump> <fsck check>

Ex: /dev/sda1 /mnt/Arjun ext4 defaults 0 0

:wq!

And run the below command to mount the filesystem automatically.

```
# mount -a
```

Note: To troubleshoot the ext version filesystems use e2fsck or fsck command along with filesystem name but the filesystem should be un-mounted before you run the command. If it is a xfs filesystem use xfs_repair command along with filesystem.

