**CentOS 7 Partition Management with fdisk Utility**

[February 11, 2019](https://systemzone.net/centos-7-partition-management-with-fdisk-utility/)  [Abu Sayeed](https://systemzone.net/author/sayeed/)  [Linux System Administration](https://systemzone.net/category/server-os/red-hat-linux/system-administration/), [Red Hat/CentOS Linux](https://systemzone.net/category/server-os/red-hat-linux/)

**CentOS or Red Hat Linux** is a popular server operating system to the system administrators. Almost all the system admins or IT guys who play with mail server, file server, proxy server and so on are so known with CentOS or Red Hat Linux and cannot go a single day without this operating system. The guys who are also wish to be a system admin should be familiar with CentOS or Red Hat operating system. In my previous article, I discussed [how to install CentOS 7 step by step with GUI mode](https://systemzone.net/centos-7-5-installation-step-by-step-with-gui-gnome-desktop/) (GNOME Desktop) or [CLI mode (minimal installation)](https://systemzone.net/red-hat-7-centos-7-minimal-installation/). I also discussed [how to configure network with nmtui tool](https://systemzone.net/centos-7-network-configuration-using-nmtui-tool/) as well as [editing network files](https://systemzone.net/red-hat-7centos-7-network-configuration/). In this article, I will discuss how to manage CentOS 7 or Red Hat 7 Linux partition with the fdisk tool. But before going to start, we should have some basic idea on the following terms.

**File System**

In computer system, a **File System** defines how data or information is stored and retrieved from a storage disk. In Windows Operating System, the popular file systems are FAT32 and NTFS. On the other hand in Linux Operating System, the popular file systems are ext2, ext3, ext4, xfs (current), vfat, swap, ZFS and GlusterFS.

**Partition Table**

Partition can be considered as a piece of disk space. A **partition table** is a partition of a disk that contains information about sizes and locations of partitions on hard disk and it is located at the first sector of the disk. The two most popular partition tables are MBR and GPT.

**Master Boot Record (MBR)**

The MBR holds the information about how the logical partitions that contains file systems are organized on the disk. It also contains executable code (referred to as a boot loader) to function as a loader for the installed operating system. In Linux Operating System the MBR can have maximum 15 partitions and among them 4 partitions are primary and the rest are logical partitions. On the other hand in the Windows Operating System, the MBR can have maximum 24 partitions where 4 are primary and the rest are logical. The MBR uses 32 bit for storing block address and for hard disks with 512-byte sectors, the MBR can handle maximum 2TB (232 × 512 bytes) hard disk.

**GUID Partition Table (GPT)**

As we can see MBR has limitation on both partitions (4 primary partitions) and storage capacity (2TB), a new partition table named GPT (Globally Unique Identifiers Partition Table) has been introduced (a part of the Unified Extensible Firmware Interface (UEFI) ) that can have 128 partitions. The GPT uses 64 bit for logical block address and for disks with 512-byte sectors, maximum size is 9.4 ZB (9.4 × 1021 bytes) or 8 ZiB.

**How Disks Are Found in CentOS 7**

In CentOS 7, the device files are located in /dev directory. SATA, SCSI and USB disks are shown as sda, sdb or  sdc (according to the number of disk) where sd represents SCSI DISK. For example, if you have a SATA or SCSI disk and install CentOS there, your disk drive will be shown as sda. As minimum automatic partition for CentOS 7 is 3, after installing CentOS 7 your sda disk partition will be labeld as sda1, sda2 and sda2. You can see your disk partition with the following commands.

[root@localhost ~]# ls /dev/sd\*  
/dev/sda  /dev/sda1  /dev/sda2  /dev/sda3

Now if you add another raw SATA or SCSI HDD, the disk will be labeled as sdb. If you run the above command, you will find the new disk label as below.

[root@localhost ~]# ls /dev/sd\*  
/dev/sda  /dev/sda1  /dev/sda2  /dev/sda3  **/dev/sdb**

Now if you add a USB flash drive, the disk will be labeled as the next available character but a partition will also be labeled like the below command output.

[root@localhost ~]# ls /dev/sd\*  
/dev/sda  /dev/sda1  /dev/sda2  /dev/sda3  /dev/sdb  /**dev/sdc  /dev/sdc1**

On the other hand your attached CD or DVD ROM will be found as cdrom or dvdrom or sr0 in dev directory.

**The fdisk Tool to Manage CentOS 7 Partition**

The fdisk is a command line utility that is used to manage partition in CentOS 7. You can find details about the fdisk tool with the man utility. In this article, we will know how to create a new partition and how to delete a partition with fdisk utility.

**Note:** The fdisk tool does not understand GUID partition table (GPT) and it is not designed for large partition (more than 2TB). In this case, the parted tool that supports multiple partition table formats (including MS-DOS and GPT) as well as large file size (more than 2TB) has to use.

To show available disk devices and partitions in your CentOS Linux, issue the fdisk command with –l option.

[root@localhost dev]# fdisk -l

**Disk /dev/sda: 21.5 GB, 21474836480 bytes, 41943040 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 label type: dos

Disk identifier: 0x00096c2a

Device Boot      Start         End      Blocks   Id  System

**/dev/sda1**   \*        2048     1026047      512000   83  Linux

/**dev/sda2**         1026048     5220351     2097152   82  Linux swap / Solaris

**/dev/sda3**         5220352    41943039    18361344   83  Linux

**Disk /dev/sdb: 1073 MB, 1073741824 bytes, 2097152 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 /dev/sdc: 4026 MB, 4026531840 bytes, 7864320 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 label type: dos

Disk identifier: 0xfdc01076

Device Boot      Start         End      Blocks   Id  System

**/dev/sdc1**   \*          63     7864319     3932128+   7  HPFS/NTFS/exFAT

From the above output we can see there are three disk drives (sda, sdb and sdc) and their present status is like the below description.

* **/dev/sda**: It is the first SATA/SCSI disk where CentOS 7 is installed and has three partitions. /dev/sda1 is using as /boot partion, /dev/sda2 is swap partition and /dev/sda3 is the root partition.
* **/dev/sdb:**It is a raw SATA/SCSI disk drive.
* **/dev/sdc:** It is a USB flash drive and has a partition /dev/sdc1 and its partition is informing that it is a NTFS formatted disk. Linux normally cannot handle NTFS drive unlike FAT32. To play with NTFS drive in CentOS, you have to use NTFS3G utility.

**Linux Partition ID**

The partition ID in a partition’s entry in the partition table inside a master boot record (MBR) is a byte value intended to specify the file system the partition contains and/or to flag special access methods used to access these partitions. The following table shows the common Linux file system and their partition ID.

|  |  |
| --- | --- |
| **File System** | **Partition ID** |
| NTFS | 7 |
| ext3/ext4/xfs | 83 |
| swap | 82 |
| LVM | 8e |
| vfat | f |
| RAID | fd |

**How to Create a New Partition with fdisk Tool in CentOS 7**

We will now create a new partition with fdisk tool. To create a new partition, there must be free sectors/free space in a disk otherwise we cannot create a new partition there. At the time of CentOS 7 installation we used the entire available sectors/spaces in sda disk for three partitions. So, we cannot do any partition in sda drive. On the other hand, we can see sdb is a raw disk added later. So we will now be able to create a new partition here. Issue the following command to start creating a new disk partition in sdb drive.

[root@localhost dev]# fdisk /dev/sdb

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 0x74bdb0ad.

Command (m for help):

Now fdisk command is waiting for further instruction. To get available instructions you can type m and hit enter, then it will show available instructions with description.

Command (m for help): m  
Command actiona   toggle a bootable flagb   edit bsd disklabelc   toggle the dos compatibility flag

d   delete a partition

g   create a new empty GPT partition table

G   create an IRIX (SGI) partition table

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):

So to create a new partition, we have to type n and hit Enter key.

Command (m for help): n

Partition type:

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

e   extended

Select (default p):

Now it will ask to choose partition type. As we are using MBR partition table, we can do only 4 primary partitions and others are extended partitions on a primary partition. We will create two partitions (one partition for data and other for backup) in sdb drive. As we are creating first partition, type p and hit Enter key.

Select (default p): p

Partition number (1-4, default 1):1

Now it asks for partition number. Type 1 and hit Enter key.

First sector (2048-2097151, default 2048): {Press Enter}

It will now ask for entering first sector. Press Enter key to choose default

Using default value 2048Last sector, +sectors or +size{K,M,G} (2048-2097151, default 2097151): +500M

It will now ask for choosing last sectors or size. The size should start with PLUS SIGN and the unit will be (K=KiloByte, M=MegaByte and G=GigaByte). Put your desired value (I am giving 500MB) and hit Knter key.

Partition 1 of type Linux and of size 500 MiB is set

So, 500MB partition has been created. To see created partition, type p and hit Enter.

Command (m for help): p

Disk /dev/sdb: 1073 MB, 1073741824 bytes, 2097152 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 label type: dos

Disk identifier: 0x4b6e12a4

Device Boot      Start         End      Blocks   Id  System

**/dev/sdb1            2048     1026047      512000   83  Linux**

To save this created partition type, w and hit Enter key.

Command (m for help): w

The partition table has been altered!

Calling ioctl() to re-read partition table.

Syncing disks.

Now if you see disk list, you will find sdb1 is now created.

[root@localhost dev]# fdisk -l

Disk /dev/sda: 21.5 GB, 21474836480 bytes, 41943040 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 label type: dos

Disk identifier: 0x00096c2a

Device Boot      Start         End      Blocks   Id  System

/dev/sda1   \*        2048     1026047      512000   83  Linux

/dev/sda2         1026048     5220351     2097152   82  Linux swap / Solaris

/dev/sda3         5220352    41943039    18361344   83  Linux

Disk /dev/sdb: 1073 MB, 1073741824 bytes, 2097152 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 label type: dos

Disk identifier: 0x4b6e12a4

Device Boot      Start         End      Blocks   Id  System

**/dev/sdb1            2048     1026047      512000   83  Linux**

Disk /dev/sdc: 4026 MB, 4026531840 bytes, 7864320 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 label type: dos

Disk identifier: 0xfdc01076

Device Boot      Start         End      Blocks   Id  System

/dev/sdc1   \*          63     7864319     3932128+   7  HPFS/NTFS/exFAT

Similarly, you can create another partition. I am creating another partition named sdb2 following the below command.

[root@localhost ~]# fdisk /dev/sdb

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 (1 primary, 0 extended, 3 free)

e   extended

Select (default p): p

Partition number (2-4, default 2): 2

First sector (1026048-2097151, default 1026048):

Using default value 1026048

Last sector, +sectors or +size{K,M,G} (1026048-2097151, default 2097151): +250M

Partition 2 of type Linux and of size 250 MiB is set

Command (m for help): p

Disk /dev/sdb: 1073 MB, 1073741824 bytes, 2097152 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 label type: dos

Disk identifier: 0x4b6e12a4

Device Boot      Start         End      Blocks   Id  System

**/dev/sdb1            2048     1026047      512000   83  Linux**

**/dev/sdb2         1026048     1538047      256000   83  Linux**

Command (m for help): w

The partition table has been altered!

Calling ioctl() to re-read partition table.

Syncing disks.

Now we have two newly created partitions. After creating partition, it is now required to inform the operating system to update partition table. Issue the following command to update partition table permanently.

[root@localhost ~]# partprobe /dev/sdb

Our partition table now has been updated. Now it is required to format our partition to use. The recent Linux supported file format system is xfs. So, issue the following commands to format sdb1 partition with xfs.

[root@localhost ~]# mkfs.xfs /dev/sdb1

meta-data=/dev/sdb1              isize=256    agcount=4, agsize=32000 blks

=                       sectsz=512   attr=2, projid32bit=1

=                       crc=0

data     =                       bsize=4096   blocks=128000, imaxpct=25

=                       sunit=0      swidth=0 blks

naming   =version 2              bsize=4096   ascii-ci=0 ftype=0

log      =internal log           bsize=4096   blocks=853, version=2

=                       sectsz=512   sunit=0 blks, lazy-count=1

realtime =none                   extsz=4096   blocks=0, rtextents=0

Similarly, format the sdb2 partition to xfs with the following command.

[root@localhost ~]# mkfs.xfs /dev/sdb

2meta-data=/dev/sdb2              isize=256    agcount=4, agsize=16000 blks

=                       sectsz=512   attr=2, projid32bit=1

=                       crc=0

data     =                       bsize=4096   blocks=64000, imaxpct=25

=                       sunit=0      swidth=0 blks

naming   =version 2              bsize=4096   ascii-ci=0 ftype=0

log      =internal log           bsize=4096   blocks=853, version=2

=                       sectsz=512   sunit=0 blks, lazy-count=1

realtime =none                   extsz=4096   blocks=0, rtextents=0

We have successfully formatted our newly created partition with xfs file system. Now these partitions are ready to keep data. But before keeping data, we have to mount partition to folder or directory. For practice, we will mound sdb1 to data directory and sdb2 to backup directory. So, first create these folders with the following command.

[root@localhost ~]# mkdir /data /backup

Now mount data to sdb1 and backup to sdb2 with the following command.

[root@localhost ~]# mount /dev/sdb1 /data

[root@localhost ~]# mount /dev/sdb2 /backup

Browsing these folders, you can keep any data now. You can see usage of disk with the following command.

[root@localhost ~]# df –HT

Filesystem     Type      Size  Used Avail Use% Mounted on

/dev/sda3      xfs        19G  5.1G   14G  28% /

devtmpfs       devtmpfs  506M     0  506M   0% /dev

tmpfs          tmpfs     514M  132k  514M   1% /dev/shm

tmpfs          tmpfs     514M  7.4M  507M   2% /run

tmpfs          tmpfs     514M     0  514M   0% /sys/fs/cgroup

/dev/sda1      xfs       521M  122M  400M  24% /boot

**/dev/sdb1      xfs       521M   27M  495M   6% /data**

**/dev/sdb2      xfs       259M   14M  246M   6% /backup**

At this stage, our mounted partitions are temporary. If our Operating System gets restarted, these mounted directory will be lost. So, we need to do permanent mount. To do permanent mount, we have to put entry in fstab file. You can see current fstab entry with the following command.

[root@localhost ~]# cat /etc/fstab

#

# /etc/fstab

# Created by anaconda on Thu Feb  7 10:22:12 2019

#

# Accessible filesystems, by reference, are maintained under ‘/dev/disk’

# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info

#

UUID=bee34176-dd55-479e-95c5-9545912d14b9 /                       xfs     defaults        1 1

UUID=dede0f3e-6c9a-4185-afb4-f129fb873246 /boot                   xfs     defaults        1 2

UUID=40464ff6-29e5-4492-9698-1b8229ca58f6 swap                    swap    defaults        0 0

You can see our new partitions are not present here. So, we have to put our new partition entry here. You can also see, to put entry we have to know UUID and type of any partitions. We can know UUID and file system with the following command.

[root@localhost ~]# blkid /dev/sdb1

/dev/sdb1: UUID=”7791407e-2c8f-4743-a3fa-93ac8c43b658″ TYPE=”xfs”

[root@localhost ~]# blkid /dev/sdb2

/dev/sdb2: UUID=”48031258-fe46-41d3-ade2-5e70971a5fa9″ TYPE=”xfs”

Knowing UUID and file system, we are now ready to put entry in fstab file. The fstab file has the following format.

[Device] [Mount Point] [File System Type] [Options] [Dump] [Pass]

|  |  |  |
| --- | --- | --- |
| **fields** | **description** | **Example** |
| <device> | The device/partition (by /dev location or UUID) that contain a file system. | UUID=7791407e-2c8f-4743-a3fa-93ac8c43b658  or /dev/sdb1 |
| <mount point> | The directory on your root file from which it will be possible to access the content of the device/partition (note: swap has no mount point). Mount points should not have spaces in the names. | /data  for /dev/sdb1 and /backup for /dev/sdb2 |
| <file system type> | Type of file system. | ext2, ext4, xfs and so on |
| <options> | Mount options of access to the device/partition (quota, acl) | defaults |
| <dump> | Enable or disable backing up of the device/partition (the command *dump*). This field is usually set to 0, which disables it. | 0 |
| <pass num> | Controls the order in which *fsck* checks the device/partition for errors at boot time. The root device should be 1. Other partitions should be 2, or 0 to disable checking. | 0 |

So, open fstab file and put two entries for sdb1 and sdb2 like the following commands and save fstab file.

[root@localhost ~]# vim /etc/fstab

UUID=bee34176-dd55-479e-95c5-9545912d14b9 /                       xfs     defaults        1 1

UUID=dede0f3e-6c9a-4185-afb4-f129fb873246 /boot                   xfs     defaults        1 2

UUID=40464ff6-29e5-4492-9698-1b8229ca58f6 swap                    swap    defaults        0 0

**/dev/sdb1                                /data                    xfs     defaults        0 0**

**/dev/sdb2                                /backup                  xfs     defaults        0 0**

To mount all file systems in fstab file, issue the following command. It also show you if any error is occurred in fstab file.

[root@localhost ~]# mount -a

New partition creation with permanent mounting using fdisk utility has been completed. Now we will show how to delete any partition in CentOS Linux if required.

**How to Delete Linux Partition with fdisk Tool**

Sometimes we may require deleting any partition. Using fdisk tool, we can easily delete any partition at any time. But before deleting any partition, you have to unmount the partition and remove fastab entry. Say, we want to remove sdb2 partition. So, first remove fstab entry by opening with vim editor.

[root@localhost ~]# vim /etc/fstab

UUID=bee34176-dd55-479e-95c5-9545912d14b9 /                       xfs     defaults        1 1

UUID=dede0f3e-6c9a-4185-afb4-f129fb873246 /boot                   xfs     defaults        1 2

UUID=40464ff6-29e5-4492-9698-1b8229ca58f6 swap                    swap    defaults        0 0

/dev/sdb1                                /data                    xfs     defaults        0 0

Now unmounts the backup directory mounted to sdb2 partition with the following command.

[root@localhost ~]# umount /backup

We are now ready delete partition sdb2 because we have removed the fstab entry and also unmounted the backup directory from sdb2 partition. Now issue the following command to delete the sdb2 partition.

[root@localhost ~]# fdisk /dev/sdb

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):

Type ‘d’ for deleting any partition.  It will also ask for partition number. Type ‘2’ because we want to delete sdb2 partiton.

Command (m for help): d

Partition number (1,2, default 2): 2

Partition 2 is deleted

You will find sdb2 partition has been deleted. To confirm deletion, type ‘p’ and then type ‘w’.

Command (m for help): p

Disk /dev/sdb: 1073 MB, 1073741824 bytes, 2097152 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 label type: dos

Disk identifier: 0x4b6e12a4

Device Boot      Start         End      Blocks   Id  System

/dev/sdb1            2048     1026047      512000   83  Linux

Command (m for help): w

The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16: Device or resource busy.

The kernel still uses the old table. The new table will be used at

the next reboot or after you run partprobe(8) or kpartx(8)

Syncing disks.

Now run partprobe command to update partition table.

[root@localhost ~]# partprobe /dev/sdb

Now if you see disk list with fdisk command, you will find the sdb2 partition is no more there.

[root@localhost ~]# fdisk -l

Disk /dev/sda: 21.5 GB, 21474836480 bytes, 41943040 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 label type: dos

Disk identifier: 0x00096c2a

Device Boot      Start         End      Blocks   Id  System

/dev/sda1   \*        2048     1026047      512000   83  Linux

/dev/sda2         1026048     5220351     2097152   82  Linux swap / Solaris

/dev/sda3         5220352    41943039    18361344   83  Linux

Disk /dev/sdb: 1073 MB, 1073741824 bytes, 2097152 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 label type: dos

Disk identifier: 0x4b6e12a4

Device Boot      Start         End      Blocks   Id  System

/dev/sdb1            2048     1026047      512000   83  Linux

With the above steps we have learnt how to delete a Linux partition with fdisk utility. We will now know how to mount a USB flash drive.

**How to Mount USB Flash Drive (Pen drive) in CentOS 7**

As I said before, the attached USB Flash Drive will show as the next available SD drive with an automatic created partition number. With the fdisk tool we can see the USB Flash Drive’s partition.

[root@localhost ~]# fdisk -l

Disk /dev/sda: 21.5 GB, 21474836480 bytes, 41943040 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 label type: dos

Disk identifier: 0x00096c2a

Device Boot      Start         End      Blocks   Id  System

/dev/sda1   \*        2048     1026047      512000   83  Linux

/dev/sda2         1026048     5220351     2097152   82  Linux swap / Solaris

/dev/sda3         5220352    41943039    18361344   83  Linux

Disk /dev/sdb: 1073 MB, 1073741824 bytes, 2097152 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 label type: dos

Disk identifier: 0x4b6e12a4

Device Boot      Start         End      Blocks   Id  System

/dev/sdb1            2048     1026047      512000   83  Linux

**Disk /dev/sdc: 4026 MB, 4026531840 bytes, 7864320 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 label type: dos

Disk identifier: 0xfdc01076

Device Boot      Start         End      Blocks   Id  System

**/dev/sdc1   \*          63     7864319     3932128+   b  W95 FAT32**

From the above output, we can see that a USB Flash Drive (sdc with partition /dev/sdc1) is attached. If the USB drive is formatted with the Linux File System described above, we can attached the partition with a directory and can list files, copy and paste any file or delete any file from there. But if the USB Flash Drive is formatted with NTFS, we have to use NTFS3G (a third party application) that will be discussed in my future article.

As we can see the attached USB Flash Drive is a FAT32 formatted drive, we will be able to mount the partition and can do copy, paste, delete or any other file operation. The following command will show how to mount a Linux file system supported USB Flash Drive in your CentOS 7 Operating System.

[root@localhost ~]# mount /dev/sdc1 /mnt

We have mounted the USB Flash Drive in automated created /mnt directory. Now we can browse the /mnt directory and can list or do any file operation there.

[root@localhost ~]# cd /mnt  
[root@localhost mnt]# ls  
Crack  dotnetfx45\_full\_x86\_x64.exe  Firefox Setup 36.0.4.exe  FoxitReaderSetup.exe  Microsoft Office 2010 (working crack)  winrar-x64-521.exe

After using USB Flash Drive or Pendrive, we have to unmount the mounted directory and can safely remove the removable device.

[root@localhost mnt]# cd ..  
[root@localhost /]# umount /mnt/  
[root@localhost /]# cd /mnt  
[root@localhost mnt]# ls

You will find no file is there. That means, the mounted disk is now uncounted and we can safely remove the USB device.

**Note:** If you use a new USB Flash Drive or a file system that is not supported by the Linux, you have to format the USB device partition with the supported file system (ext2, ext3, ext4, xfs and so on) by the mkfs tool and mount the device in a directory or folder and then use the USB device browsing that directory or folder.

**How to Mount CD/DVD in CentOS 7**

As I said before, CD or DVD device is found in /dev/sr0 location. So to access CD or DVD media, we have to mount this location in a directory or folder. In the following example, we will mount CD or DVD location in automatic created directory media and browse media directory to get files in CD or DVD ROM.