



phoenixNAP[®]
GLOBAL IT SERVICES

How to Format Disk Partitions in Linux



ext4
FAT32
NTFS



Introduction

A disk partition must be formatted and mounted before use. The formatting process can also be done for several other reasons, such as changing the file system, fixing errors, or deleting all data.

In this tutorial, you will learn how to format and mount disk partitions in Linux using ext4, FAT32, or NTFS file system.

Prerequisites

- A system running Linux
- A user account with **sudo** or **root** privileges
- Access to a terminal window / command line (**Activities** > **Search** > **Terminal**)

Checking the Partitions

Before formatting, locate a partition you wish to format. To do so, run the **lsblk** command that displays block devices. Block devices are files that represent devices such as hard drives, RAM disks, USB drives, and CD-ROM drives.

```
lsblk
```

The terminal prints out a list of all block devices as well as information about them:

- **NAME** – Device names
- **MAJ:MIN** – Major or minor device numbers
- **RM** – Whether the device is removable (1 if yes, 0 if no)
- **SIZE** – The size of the device
- **RO** – Whether the device is read-only
- **TYPE** – The type of the device
- **MOUNTPOINT** – Device's mount point

We will use the `/dev/sdb1` partition as an example.

```
n@n-VirtualBox:~$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
loop0       7:0      0   55M  1 loop /snap/core18/1880
loop1       7:1      0  55,4M  1 loop /snap/core18/1932
loop2       7:2      0 255,6M  1 loop /snap/gnome-3-34-1804/36
loop3       7:3      0 217,9M  1 loop /snap/gnome-3-34-1804/60
loop4       7:4      0  62,1M  1 loop /snap/gtk-common-themes/1506
loop5       7:5      0  29,9M  1 loop /snap/snapd/8542
loop6       7:6      0   51M  1 loop /snap/snap-store/498
loop7       7:7      0  49,8M  1 loop /snap/snap-store/467
loop8       7:8      0  64,8M  1 loop /snap/gtk-common-themes/1514
loop9       7:9      0  31,1M  1 loop /snap/snapd/10238
sda         8:0      0  10G   0 disk
├─sda1      8:1      0  512M   0 part /boot/efi
├─sda2      8:2      0    1K   0 part
└─sda5      8:5      0   9,5G   0 part /
sdb         8:16     0   2,5G   0 disk
└─sdb1      8:17     0  762M   0 part
sr0        11:0     1 1024M   0 rom
n@n-VirtualBox:~$
```

The `lsblk` command without additional options does not display information about the devices' file systems.

To display a list containing file system information, add the `-f` option:

```
lsblk -f
```

The terminal prints out the list of all block devices. The partitions that do not contain information on the file system in use are non-formatted partitions.

```
n@n-VirtualBox:~$ lsblk -f
```

NAME	FSTYPE	LABEL	UUID	FS-AVAIL	FS-USE%	MOUNTPOINT
loop0	squashf			0	100%	/snap/core18/1
loop1	squashf			0	100%	/snap/core18/1
loop2	squashf			0	100%	/snap/gnome-3-
loop3	squashf			0	100%	/snap/gnome-3-
loop4	squashf			0	100%	/snap/gtk-comm
loop5	squashf			0	100%	/snap/snapd/85
loop6	squashf			0	100%	/snap/snap-sto
loop7	squashf			0	100%	/snap/snap-sto
loop8	squashf			0	100%	/snap/gtk-comm
loop9	squashf			0	100%	/snap/snapd/10
sda						
├ sda1	vfat		5096-4E6F	511M	0%	/boot/efi
├ sda2						
└ sda5	ext4		efe91a2f-b255-40cd-88ee-9217871e9733	1,6G	78%	/
sdb						
└ sdb1						
sr0						

Formatting Disk Partition in Linux

There are three ways to format disk partitions using the **mkfs** command, depending on the file system type:

- ext4
- FAT32
- NTFS

The general syntax for formatting disk partitions in Linux is:

```
mkfs [options] [-t type fs-options] device [size]
```

Formatting Disk Partition with ext4 File System

1. Format a disk partition with the ext4 file system using the following command:

```
sudo mkfs -t ext4 /dev/sdb1
```

2. Next, verify the file system change using the command:

```
lsblk -f
```

The terminal prints out a list of block devices.

3. Locate the preferred partition and confirm that it uses the ext4 file system.

```
n@n-VirtualBox:~$ sudo mkfs -t ext4 /dev/sdb1
mkfs.ext4 1.45.5 (07-Jan-2020)
Creating filesystem with 195072 4k blocks and 48768 inodes
Filesystem UUID: bb941567-c1f4-447d-a99e-1d6777e67979
Superblock backups stored on blocks:
    32768, 98304, 163840

Allocating group tables: done
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done

n@n-VirtualBox:~$ lsblk -f
NAME        FSTYPE LABEL UUID                                 FSAVAIL FSUSE% MOUNTPOINT
sda
├─sda1 vfat          5096-4E6F                                511M    0% /boot/efi
├─sda2
├─sda5 ext4        efe91a2f-b255-40cd-88ee-9217871e9733    1,5G    79% /
sdb
└─sdb1 ext4        bb941567-c1f4-447d-a99e-1d6777e67979
```

Formatting Disk Partition with FAT32 File System

1. To format a disk with a FAT32 file system, use:

```
sudo mkfs -t vfat /dev/sdb1
```

2. Again, run the `lsblk` command to verify the file system change and locate the preferred partition from the list.

```
lsblk -f
```

The expected output is:

```
n@n-VirtualBox:~$ sudo mkfs -t vfat /dev/sdb1
mkfs.fat 4.1 (2017-01-24)
n@n-VirtualBox:~$ lsblk -f
NAME        FSTYPE LABEL UUID                                 FSAVAIL FSUSE% MOUNTPOINT
sda
├─sda1 vfat          5096-4E6F                                511M    0% /boot/efi
├─sda2
├─sda5 ext4        efe91a2f-b255-40cd-88ee-9217871e9733    1,5G    79% /
sdb
└─sdb1 vfat        CA24-BFB7
sr0
```

Formatting Disk Partition with NTFS File System

1. Run the `mkfs` command and specify the NTFS file system to format a disk:

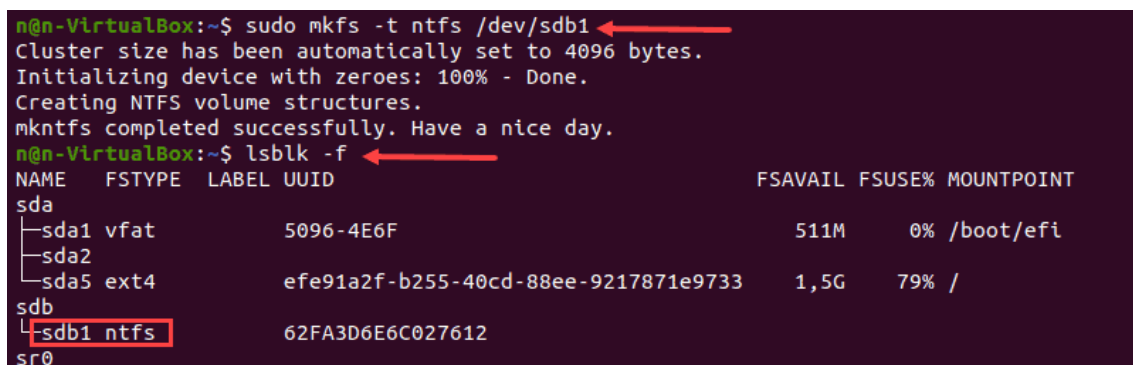
```
sudo mkfs -t ntfs /dev/sdb1
```

The terminal prints a confirmation message when the formatting process completes.

2. Next, verify the file system change using:

```
lsblk -f
```

3. Locate the preferred partition and confirm that it uses the NTFS file system.



The terminal output shows the successful execution of the `mkfs` command and the subsequent `lsblk -f` command. The `lsblk -f` output is as follows:

NAME	FSTYPE	LABEL	UUID	FSAVAIL	FSUSE%	MOUNTPOINT
sda						
├─sda1	vfat		5096-4E6F	511M	0%	/boot/efi
├─sda2						
├─sda5	ext4		efe91a2f-b255-40cd-88ee-9217871e9733	1,5G	79%	/
sdb						
└─sdb1	ntfs		62FA3D6E6C027612			
sr0						

Mounting the Disk Partition in Linux

Before using the disk, create a mount point and mount the partition to it. A mount point is a directory used to access data stored in disks.

1. Create a mount point by entering:

```
sudo mkdir -p [mountpoint]
```

2. After that, mount the partition by using the following command:

```
sudo mount -t auto /dev/sdb1 [mountpoint]
```

Note: Replace [mountpoint] with the preferred mount point (example: /usr/media).

There is no output if the process completes successfully.

```
n@n-VirtualBox:~$ sudo mkdir -p /mnt/sdb1
n@n-VirtualBox:~$ sudo mount -t auto /dev/sdb1 /mnt/sdb1
```

3. Verify if the partition is mounted using the following command:

```
lsblk -f
```

The expected output is:

```
n@n-VirtualBox:~$ lsblk -f
NAME        FSTYPE LABEL UUID                                 FSAVAIL FSUSE% MOUNTPOINT
loop0       squashf                                0      100% /snap/core18/1
loop1       squashf                                0      100% /snap/core18/1
loop2       squashf                                0      100% /snap/gnome-3-
loop3       squashf                                0      100% /snap/gnome-3-
loop4       squashf                                0      100% /snap/gtk-comm
loop5       squashf                                0      100% /snap/snapd/85
loop6       squashf                                0      100% /snap/snap-sto
loop7       squashf                                0      100% /snap/snap-sto
loop8       squashf                                0      100% /snap/gtk-comm
loop9       squashf                                0      100% /snap/snapd/10
sda
├─sda1      vfat          5096-4E6F              511M     0% /boot/efi
├─sda2
├─sda5      ext4          efe91a2f-b255-40cd-88ee-9217871e9733 1,6G    78% /
sdb
├─sdb1      ext4          f4b8ab07-ea8f-4e48-8130-0cd8724fadbd 679,2M  0% /mnt/sdb1
sr0
```

Understanding the Linux File System

Choosing the right file system before formatting a storage disk is crucial. Each type of file system has different file size limitations or different operating system compatibility.

The most commonly used file systems are:

- FAT32
- NTFS
- ext4

Their main features and differences are:

File System	Supported File Size	Compatibility	
FAT32	up to 4 GB	Windows, Mac, Linux	For maximum compatibility
NTFS	16 EiB – 1 KB	Windows, Mac (read-only), most Linux distributions	For internal drives and Windows system file
Ext4	16 GiB – 16 TiB	Windows, Mac, Linux (requires extra drivers to access)	For files larger than 4 GB

Note: Refer to our [Introduction to Linux File System](#) article to learn more about the evolution and features of different Linux file systems.

Conclusion

After following this tutorial, you should be able to format and mount a partition in Linux in various file systems. Partition manipulation is essential for efficient data management, and next, we recommend learning [how to delete a partition in Linux](#).