**Different types of Disks in Disk Management**

## Disk technology

In the beginning days of Unix and later Linux, disks were physically large, but very small in terms of storage capacity. A 300 megabyte disk in the mid-90’s was the size of a shoebox. Today, you can get multi-terrabyte disks that are the size of a slice of toast.

Traditionally, files resided within file systems that resided in disk partitions that were themselves simply slices of disks. This organization still dominates today, though servers in large data centers often take on an entirely different structure.

/\

/ \

/ \

/ file \

/ \

/==========\

/ \

/ file system \

/ \

/==================\

/ disk partition \

/======================\

/ disk \

/==========================\

This simplistic view still works for many systems, but these days there are lot of complexities that make disk management harder in some ways and easier in others. A file system might be virtual – no longer residing on a single disk and more complex to manage, but far easier to resize as needed. In fact, the entire system could be virtual. And what we might manage as if it were a single disk could actually be some portion of a very large disk array.

Volume 0%

## Disk management

Sysadmins generally have to deal with many issues when it comes to managing disks. These include:

* Partitioning disks..
* Creating file systems
* Mounting file systems
* Sharing file systems
* Monitoring free space within file systems
* Backing up (and sometimes rest
* oring) file systems

The reasons to partition a disk include:

* protecting some file systems from running out of space (e.g., you may want the OS partition to be separated from home directories or applications to keep it from being affected if users’ files begin to take up far an excessive amount of disk space)
* improving performance
* allocating swap space
* facilitating maintenance and backups (e.g., you might be able to unmount /apps if it’s not part of / and you might want to back up /home more frequently than /usr)
* more efficient (and targeted) fsck
* maintaining (particularly on test systems) multiple operating systems
* reserving enough disk space for file system expansion
* sharing select file systems with other systems

## Partitioning commands

For most Linux servers, partitioning is done before the servers are deployed. On the other hand, you might add disks at some later time or hold back some significant amount of free disk space for future use.

To make changes or verify partitions, enter a command such as **fdisk /dev/sda** to start fdisk interactively and then type m to see a list of the things that you can do with the fdisk command.

$ sudo 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)

As you can see, the fdisk command provides a *lot* of functionality. The partitions that you set up may look something like this configuration in which four partitions have been set up on a single disk – /dev/sda.

sda

+------------+------------------------+--------------------+------+

| / 40G | /home 80G | /apps 70G | swap |

+------------+------------------------+--------------------+------+

sda1 sda2 sda3 sda4

## Examining disk space and disk partitions

There are a number of excellent commands for examining disk partitions. The **df** command is one of the most commonly used commands for reporting on disk space usage. With the -h option, the df command displays the measurements in the most "human-friendly" format and that is, in fact, what the “h” is meant to imply. As you can see in the example below, the measurements are displayed in kilobytes, megabytes or gigabytes depending on the sizes rather than all using the same scale.

$ df -h

Filesystem Size Used Avail Use% Mounted on

udev 969M 4.0K 969M 1% /dev

tmpfs 196M 1.1M 195M 1% /run

/dev/sda1 37G 4.5G 31G 13% /

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

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

none 980M 152K 979M 1% /run/shm

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

/dev/sda3 28G 44M 26G 1% /apps

The **pydf** command (think "python df" as it's really a python script) also provides a very useful disk usage display showing mount points and cute little illustrations for how full each partition is.

$ pydf

Filesystem Size Used Avail Use% Mounted on

/dev/sda1 37G 4534M 30G 12.1 [##...........] /

/dev/sda3 27G 44M 26G 0.2 [.............] /apps

The **parted** command displays partition information in a different format:

$ sudo parted -l

Model: ATA WDC WD800AAJS-60 (scsi)

Disk /dev/sda: 80.0GB

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

Partition Table: msdos

Number Start End Size Type File system Flags

1 1049kB 40.0GB 40.0GB primary ext4 boot

2 40.0GB 50.0GB 10.0GB primary linux-swap(v1)

3 50.0GB 80.0GB 30.0GB primary ext4

The **lsblk** (list block devices) command illustrates the relationship between disks and their partitions graphically and also supplies the major and minor device numbers and mount points.

$ lsblk

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

sda 8:0 0 74.5G 0 disk

├─sda1 8:1 0 37.3G 0 part /

├─sda2 8:2 0 9.3G 0 part [SWAP]

└─sda3 8:3 0 28G 0 part /apps

The **fdisk** command reports more details on disk partitions and uses very different numbers. You can also use fdisk to create or delete partitions, list unpartitioned space, change a partition type, or verify the partition table.

$ sudo fdisk -l

Disk /dev/sda: 80.0 GB, 80026361856 bytes

255 heads, 63 sectors/track, 9729 cylinders, total 156301488 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: 0x000f114b

Device Boot Start End Blocks Id System

/dev/sda1 \* 2048 78125055 39061504 83 Linux

/dev/sda2 78125056 97656831 9765888 82 Linux swap / Solaris

/dev/sda3 97656832 156301311 29322240 83 Linux

The **sfdisk** command is similar to fdisk, but makes some partition manipulation activities easier to perform.

$ sudo sfdisk -l -uM

Disk /dev/sda: 9729 cylinders, 255 heads, 63 sectors/track

Units = mebibytes of 1048576 bytes, blocks of 1024 bytes, counting from 0

Device Boot Start End MiB #blocks Id System

/dev/sda1 \* 1 38146 38146 39061504 83 Linux

/dev/sda2 38147 47683 9537 9765888 82 Linux swap / Solaris

/dev/sda3 47684 76318 28635 29322240 83 Linux

/dev/sda4 0 - 0 0 0 Empty

**NOTE:** A mebibyte (MiB) = 220bytes or 1,048,576 bytes.

The cfdisk command can also be used to display or manipulate disk partitions.

$ sudo cfdisk

cfdisk (util-linux 2.20.1)

Disk Drive: /dev/sda

Size: 80026361856 bytes, 80.0 GB

Heads: 255 Sectors per Track: 63 Cylinders: 9729

Name Flags Part Type FS Type [Label] Size (MB)

--------------------------------------------------------------------------

Pri/Log Free Space 1.05\*

sda1 Boot Primary ext4 39998.99\*

sda2 Primary swap 10000.27\*

sda3 Primary ext4 30025.98\*

Pri/Log Free Space 0.10\*

[ Help ] [ New ] [ Print ] [ Quit ] [ Units ]

[ Write ]

Create new partition from free space

## Monitoring disk performance

The **iostat** command can display statistics that illustrate how disks are performing, including how heavily they are being used. It also displays important measurements that show how busy the CPU is and how much of its resources are used for types of work. The system described below is idle more then 95% of the time. More importantly for our focus on disks, the %iowait (CPU waiting on disk IO) is very low. This would not be true if the disk were unusually busy and disk IO were a bottleneck.

$ iostat -x 60

Linux 3.13.0-129-generic (stinkbug) 08/31/2017 \_x86\_64\_ (2 CPU)

avg-cpu: %user %nice %system **%iowait** %steal %idle

0.93 1.15 0.35 **1.86** 0.00 95.73

Device: rrqm/s wrqm/s r/s w/s rkB/s wkB/s avgrq-sz avgqu-sz await r\_await w\_await svctm %util

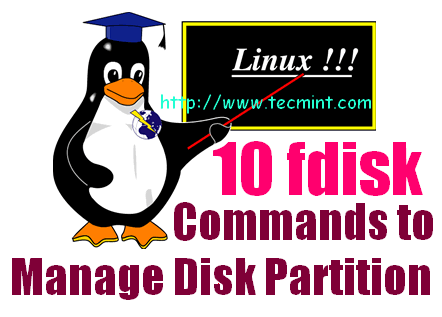
sda 8.37 3.26 13.41 2.79 341.14 191.82 65.79 0.61 37.60 30.40 72.14 2.52 4.08

Probably one of the most informative commands for looking at disk health is **smartctl** (part of smartmontools). While the command generates a lot of output, it provides valuable measurements that might help you pinpoint disk problems, particularly once you get used to working with its extensive output.

**10 fdisk Commands to Manage Linux Disk Partitions**

**fdisk** stands (for “**fixed disk** or **format disk**“) is an most commonly used command-line based disk manipulation utility for a **Linux/Unix** systems. With the help of fdisk command you can view, create, resize, delete, change, copy and move partitions on a hard drive using its own user friendly text based menu driven interface.

This tool is very useful in terms of creating space for new partitions, organising space for new drives, re-organising an old drives and copying or moving data to new disks. It allows you to create a maximum of four new **primary** partition and number of logical (**extended**) partitions, based on size of the hard disk you have in your system.

[](https://www.tecmint.com/10-fdisk-commands-to-manage-linux-disk-partitions/fdisk-command/)fdisk command to manage disk partition

This article explains 10 basic **fdisk commands** to manage a partition table in Linux based systems. You must be **root** user to run fdisk command, otherwise you will get a “**command not found**” error.

**Caution – Don’t Create, Delete or Modify Partitions. Unless you know what you are doing!**

### 1. View all Disk Partitions in Linux

The following basic command list all existing disk partition on your system. The ‘**-l**‘ argument stand for (listing all partitions) is used with fdisk command to view all available partitions on Linux. The partitions are displayed by their device’s names. For example: **/dev/sda**, **/dev/sdb** or **/dev/sdc**.

[root@tecmint.com ~]# fdisk -l

Disk /dev/sda: 637.8 GB, 637802643456 bytes

255 heads, 63 sectors/track, 77541 cylinders

Units = cylinders of 16065 \* 512 = 8225280 bytes

Device Boot Start End Blocks Id System

/dev/sda1 \* 1 13 104391 83 Linux

/dev/sda2 14 2624 20972857+ 83 Linux

/dev/sda3 2625 4582 15727635 83 Linux

/dev/sda4 4583 77541 586043167+ 5 Extended

/dev/sda5 4583 5887 10482381 83 Linux

/dev/sda6 5888 7192 10482381 83 Linux

/dev/sda7 7193 7845 5245191 83 Linux

/dev/sda8 7846 8367 4192933+ 82 Linux swap / Solaris

/dev/sda9 8368 77541 555640123+ 8e Linux LVM

### 2. View Specific Disk Partition in Linux

To view all partitions of specific hard disk use the option ‘**-l**‘ with device name. For example, the following command will display all disk partitions of device **/dev/sda**. If you’ve different device names, simple write device name as **/dev/sdb** or **/dev/sdc**.

[root@tecmint.com ~]# fdisk -l /dev/sda

Disk /dev/sda: 637.8 GB, 637802643456 bytes

255 heads, 63 sectors/track, 77541 cylinders

Units = cylinders of 16065 \* 512 = 8225280 bytes

Device Boot Start End Blocks Id System

/dev/sda1 \* 1 13 104391 83 Linux

/dev/sda2 14 2624 20972857+ 83 Linux

/dev/sda3 2625 4582 15727635 83 Linux

/dev/sda4 4583 77541 586043167+ 5 Extended

/dev/sda5 4583 5887 10482381 83 Linux

/dev/sda6 5888 7192 10482381 83 Linux

/dev/sda7 7193 7845 5245191 83 Linux

/dev/sda8 7846 8367 4192933+ 82 Linux swap / Solaris

/dev/sda9 8368 77541 555640123+ 8e Linux LVM

### 3. Check all Available fdisk Commands

If you would like to view all commands which are available for fdisk. Simply use the following command by mentioning the hard disk name such as **/dev/sda**as shown below. The following command will give you output similar to below.

[root@tecmint ~]# fdisk /dev/sda

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to

switch off the mode (command 'c') and change display units to

sectors (command 'u').

Command (m for help):

Type ‘**m**‘ to see the list of all available commands of fdisk which can be operated on **/dev/sda** hard disk. After, I enter ‘**m**‘ on the screen, you will see the all available options for fdisk that you can be used on the **/dev/sda** device.

[root@tecmint ~]# fdisk /dev/sda

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to

switch off the mode (command 'c') and change display units to

sectors (command 'u').

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):

### 4. Print all Partition Table in Linux

To print all partition table of hard disk, you must be on command mode of specific hard disk say **/dev/sda**.

[root@tecmint ~]# fdisk /dev/sda

From the command mode, enter ‘**p**‘ instead of ‘**m**‘ as we did earlier. As I enter ‘**p**‘, it will print the specific **/dev/sda** partition table.

Command (m for help): **p**

Disk /dev/sda: 637.8 GB, 637802643456 bytes

255 heads, 63 sectors/track, 77541 cylinders

Units = cylinders of 16065 \* 512 = 8225280 bytes

Device Boot Start End Blocks Id System

/dev/sda1 \* 1 13 104391 83 Linux

/dev/sda2 14 2624 20972857+ 83 Linux

/dev/sda3 2625 4582 15727635 83 Linux

/dev/sda4 4583 77541 586043167+ 5 Extended

/dev/sda5 4583 5887 10482381 83 Linux

/dev/sda6 5888 7192 10482381 83 Linux

/dev/sda7 7193 7845 5245191 83 Linux

/dev/sda8 7846 8367 4192933+ 82 Linux swap / Solaris

/dev/sda9 8368 77541 555640123+ 8e Linux LVM

Command (m for help):

### 5. How to Delete a Partition in Linux

If you would like to delete a specific partition (i.e **/dev/sda9**) from the specific hard disk such as **/dev/sda**. You must be in fdisk command mode to do this.

[root@tecmint ~]# fdisk /dev/sda

Next, enter ‘**d**‘ to delete any given partition name from the system. As I enter ‘**d**‘, it will prompt me to enter partition number that I want to delete from **/dev/sda** hard disk. Suppose I enter number ‘**4**‘ here, then it will delete partition number ‘**4**‘ (i.e. **/dev/sda4**) disk and shows free space in partition table. Enter ‘**w**‘ to write table to disk and exit after making new alterations to partition table. The new changes would only take place after next reboot of system. This can be easily understood from the below output.

[root@tecmint ~]# fdisk /dev/sda

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to

switch off the mode (command 'c') and change display units to

sectors (command 'u').

Command (m for help): **d**

Partition number (1-4): **4**

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.

You have new mail in /var/spool/mail/root

**Warning** : Be careful, while performing this step, because using option ‘**d**‘ will completely delete partition from system and may lost all data in partition.

### 6. How to Create a New Partition in Linux

If you’ve free space left on one of your device say **/dev/sda** and would like to create a new partition under it. Then you must be in fdisk command mode of **/dev/sda**. Type the following command to enter into command mode of specific hard disk.

[root@tecmint ~]# fdisk /dev/sda

After entering in command mode, now press “**n**” command to create a new partition under **/dev/sda** with specific size. This can be demonstrated with the help of following given output.

[root@tecmint ~]# fdisk /dev/sda

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to

switch off the mode (command 'c') and change display units to

sectors (command 'u').

Command (m for help): **n**

Command action

e extended

p primary partition (1-4)

**e**

While creating a new partition, it will ask you two options ‘**extended**‘ or ‘**primary**‘ partition creation. Press ‘**e**‘ for extended partition and ‘**p**‘ for primary partition. Then it will ask you to enter following two inputs.

1. First cylinder number of the partition to be create.
2. Last cylinder number of the partition to be created (Last cylinder, +cylinders or +size).

You can enter the size of cylinder by adding “**+5000M**” in last cylinder. Here, ‘**+**‘ means addition and **5000M** means size of new partition (i.e **5000MB**). Please keep in mind that after creating a new partition, you should run ‘**w**‘ command to alter and save new changes to partition table and finally reboot your system to verify newly created partition.

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.

### 7. How to Format a Partition in Linux

After the new partition is created, don’t skip to format the newly created partition using ‘**mkfs**‘ command. Type the following command in the terminal to format a partition. Here **/dev/sda4** is my newly created partition.

[root@tecmint ~]# mkfs.ext4 /dev/sda4

### 8. How to Check Size of a Partition in Linux

After formatting new partition, check the size of that partition using flag ‘**s**‘ (displays size in blocks) with fdisk command. This way you can check size of any specific device.

[root@tecmint ~]# fdisk -s /dev/sda2

5194304

### 9. How to Fix Partition Table Order

If you’ve deleted a logical partition and again recreated it, you might notice ‘**partition out of order**‘ problem or error message like ‘**Partition table entries are not in disk order**‘.

For example, when three logical partitions such as (**sda4**, **sda5** and **sda6**) are deleted, and new partition created, you might expect the new partition name would be **sda4**. But, the system would create it as **sda5**. This happens because of, after the partition are deleted, **sda7** partition had been moved as **sda4** and free space shift to the end.

To fix such partition order problems, and assign **sda4** to the newly created partition, issue the ‘**x**‘ to enter an extra functionality section and then enter ‘**f**‘ expert command to fix the order of partition table as shown below.

[root@tecmint ~]# fdisk /dev/sda

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to

switch off the mode (command 'c') and change display units to

sectors (command 'u').

Command (m for help): **x**

Expert command (m for help): **f**

Done.

Expert 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.

After, running ‘**f**‘ command, don’t forget to run ‘**w**‘ command to save and exit from fdisk command mode. Once it fixed partition table order, you will no longer get error messages.

### 10. How to Disable Boot Flag (\*) of a Partition

By default, fdisk command shows the boot flag (i.e. ‘**\***‘) symbol on each partition. If you want to enable or disable boot flag on a specific partition, do the following steps.

[root@tecmint ~]# fdisk /dev/sda

Press ‘**p**‘ command to view the current partition table, you see there is a boot flag (asterisk (**\***) symbol in orange color) on **/dev/sda1** disk as shown below.

[root@tecmint ~]# fdisk /dev/sda

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to

switch off the mode (command 'c') and change display units to

sectors (command 'u').

Command (m for help): p

Disk /dev/sda: 637.8 GB, 637802643456 bytes

255 heads, 63 sectors/track, 77541 cylinders

Units = cylinders of 16065 \* 512 = 8225280 bytes

Device Boot Start End Blocks Id System

/dev/sda1 **\*** 1 13 104391 83 Linux

/dev/sda2 14 2624 20972857+ 83 Linux

/dev/sda3 2625 4582 15727635 83 Linux

/dev/sda4 4583 77541 586043167+ 5 Extended

/dev/sda5 4583 5887 10482381 83 Linux

/dev/sda6 5888 7192 10482381 83 Linux

/dev/sda7 7193 7845 5245191 83 Linux

/dev/sda8 7846 8367 4192933+ 82 Linux swap / Solaris

/dev/sda9 8368 77541 555640123+ 8e Linux LVM

Next enter command ‘**a**‘ to disable boot flag, then enter partition number ‘**1**‘ as (i.e. **/dev/sda1**) in my case. This will disable boot flag on the partition **/dev/sda1**. This will remove the asterisk (**\***) flag.

Command (m for help): **a**

Partition number (1-9): **1**

Command (m for help): **p**

Disk /dev/sda: 637.8 GB, 637802643456 bytes

255 heads, 63 sectors/track, 77541 cylinders

Units = cylinders of 16065 \* 512 = 8225280 bytes

Device Boot Start End Blocks Id System

/dev/sda1 1 13 104391 83 Linux

/dev/sda2 14 2624 20972857+ 83 Linux

/dev/sda3 2625 4582 15727635 83 Linux

/dev/sda4 4583 77541 586043167+ 5 Extended

/dev/sda5 4583 5887 10482381 83 Linux

/dev/sda6 5888 7192 10482381 83 Linux

/dev/sda7 7193 7845 5245191 83 Linux

/dev/sda8 7846 8367 4192933+ 82 Linux swap / Solaris

/dev/sda9 8368 77541 555640123+ 8e Linux LVM

Command (m for help):

I’ve tried my best to include almost all basic commands of fdisk commands, but still fdisk contains a variety of other expert commands you can use them by entering ‘**x**‘. For more detailed information, check out ‘**man fdisk**‘ command from the terminal. If I’ve missed any important command, please do share with me via comment section.

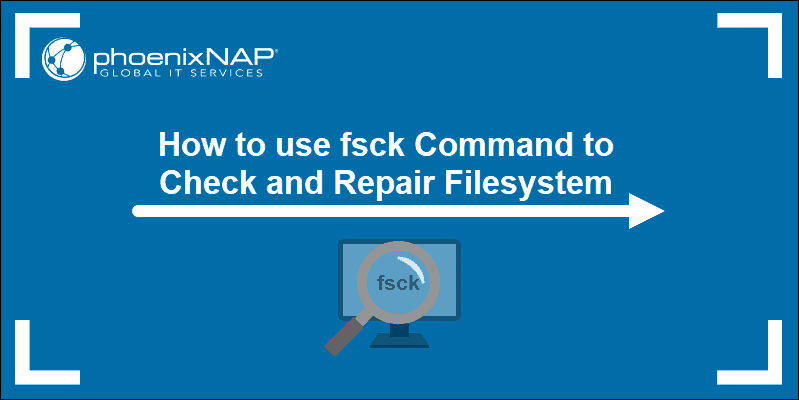
**Disk Management fsck command**

**Introduction**

The **fsck**(File System Consistency Check) Linux utility checks filesystems for errors or outstanding issues. The tool is used to fix potential errors and generate reports.

This utility comes by default with Linux distributions. No specific steps or an installation procedure is required to use fsck. Once you load the terminal, you are ready to exploit the functionalities of the tool.

Follow this guide to learn **how to use fsck to check and repair filesystem on a Linux machine**. The tutorial will list examples of how to use the tool and for which use cases.



**Prerequisites**

* Linux or UNIX-like system
* Access to a terminal or command line
* A user with root permissions to run the tool

## When to Use fsck in Linux

The fsck tool can be used in various situations:

* + - Use fsck to run a filesystem check as **preventive maintenance** or when there is an issue with your system.
    - One common problem fsck can diagnose is when the **system fails to boot**.
    - Another one is when you get an **input/output error** when the files on your system become corrupt.
    - You can also use the fsck utility to check the health of **external drives**, such as **SD cards**or**USB flash drives**.

## Basic fsck Syntax

The basic syntax for the fsck utility follows this pattern:

fsck <options> <filesystem>

In the above example, *filesystem* can be a device, a partition, a mount point, etc. You can also use filesystem-specific options at the end of the command.

## How to Check and Repair Filesystem

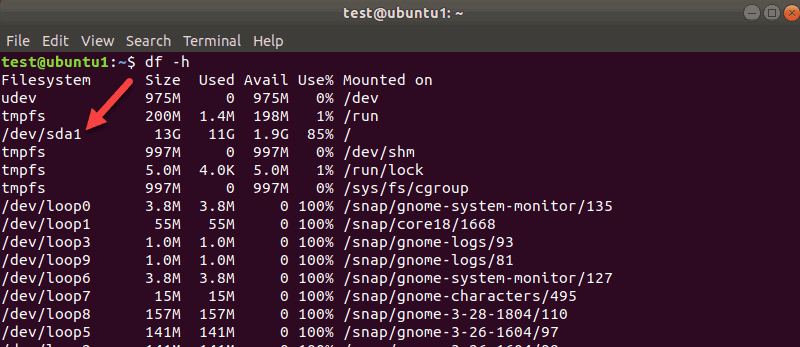
There are a few steps to do before you check and repair your file system. You need to locate a device and unmount.

### View Mounted Disks and Partitions

To view all mounted devices on your system and check disk location, use one of the available tools in Linux.

One method to locate the disk you want to scan is to list the filesystem disks with the **df**command:

df -h



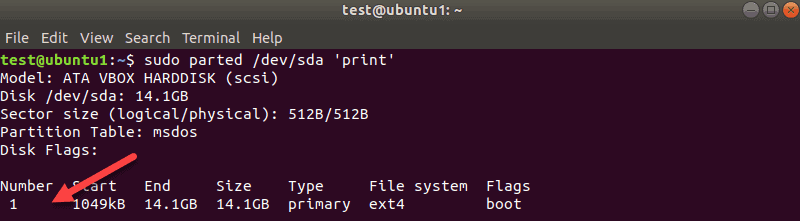
The tool prints the data usage on your system and filesystems. Take note of the disk you want to check with the **fsck** command.

**To view partitions**for your first disk, for example, use the following command:

sudo parted /dev/sda 'print'

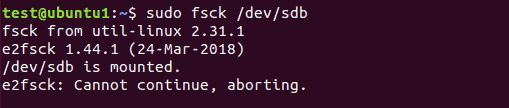
**sda**is how Linux refers to your first SCSI disk. If you have two, the second would be **sdb**, and so on.

In our example, we got one result since there was only one partition on this virtual machine. You will get more results if you have more partitions.



The disk name here is **/dev/sda**and then the number of partitions is shown in the Number column. In our case, it is one: **sda1.**

### Unmount the Disk

Before you can run a disk check with **fsck**, you need to unmount a disk or partition. If you try to run **fsck** on a mounted disk or partition, you will get a warning:  


Make sure to run the **unmount** command:

sudo umount /dev/sdb

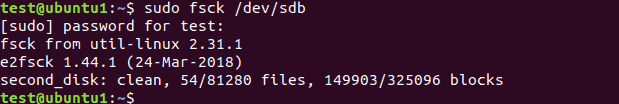
Replace */dev/sdb* with the device you want to unmount.

Note that you cannot unmount root filesystems. Hence, now **fsck** can’t be used on a running machine. More on that towards the end of the guide.

### Run fsck to Check for Errors

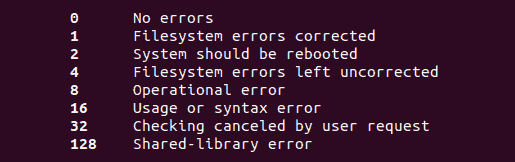
Now that you unmounted the disk, you can run **fsck**. To check the second disk, enter:

sudo fsck /dev/sdb



The above example shows the output for a clean disk. If there are multiple issues on your disk, a prompt appears for each one where you have to confirm the action.

The exit code the fsck utility returns is the sum of these states:



### Mount the Disk

When you finish checking and repairing a device, mount the disk so you can use it again.

In our case, we will remount the**sdb** disk:

mount /dev/sdb

### Do a Dry Run with fsck

Before you perform a live check, you can do a test run with fsck. Pass the **-N**option to the **fsck** command to perform a test:

sudo fsck -N /dev/sdb

The output prints what would happen but does not perform any actions.

### Fix Detected Errors Automatically with fsck

To try to fix potential problems without getting any prompts, pass the **-y** option to **fsck**.

sudo fsck -y /dev/sdb

This way, you say “yes, try to fix all detected errors” without being prompted every time.

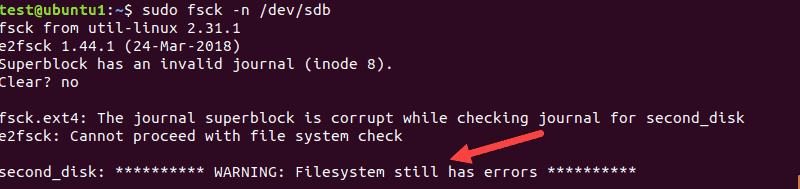
If no errors are found, the output looks the same as without the **-y**option.

### Skip Repair but Print fsck Errors in the Output

Use the **-n**option if you want to check potential error on a [file system](https://phoenixnap.com/kb/linux-file-system) without repairing them.

We have a second drive **sdb** with some journaling errors. The **-n** flag prints the error without fixing it:

sudo fsck -n /dev/sdb

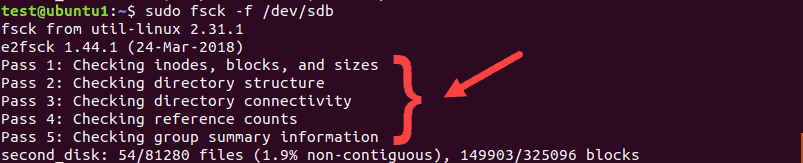


### Force fsck to Do a Filesystem Check

When you perform a fsck on a clean device, the tool skips the filesystem check. If you want to force the filesystem check, use the **-f**option.

For example:

sudo fsck -f /dev/sdb



The scan will perform all five checks to search for corruptions even when it thinks there are no issues.

### Run fsck on All Filesystems at Once

If you want to perform a check on all filesystems with fsck in one go, pass the **-A** flag. This option will go through the *etc/fstab* file in one run.

Since root filesystems can’t be unmounted on a running machine, add the **-R**option to skip them:

fsck -AR

To avoid the prompts, add the **-y** option we talked about.

### Skip fsck on a Specific Filesystem

If you want fsck to skip checking a filesystem, you need to add **-t**and “no” before a filesystem.

For example, to skip ext3 filesystem, run this command:

sudo fsck -AR -t noext3 -y

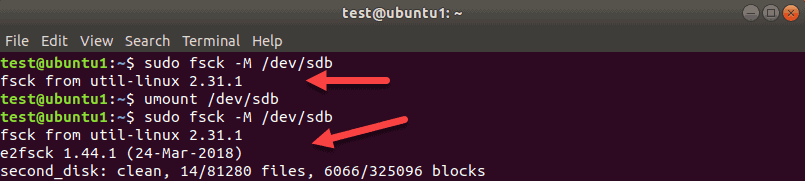
We added **-y** to skip the prompts.

### Skip Fsck on Mounted Filesystems

To make sure you do not try to run fsck on a mounted filesystem, add the **-M**option. This flag tells the fsck tool to skip any mounted filesystems.

To show you the difference, we will run fsck on **sdb** while it is mounted, and then when we unmount it.

sudo fsck -M /dev/sdb



While **sdb**is mounted, the tool exits without running a check. Then, we unmount **sdb** and run the same command again. This time, **fsck** checks the disk and reports it as clean, or with errors.

**Note:** To remove the first title line of the fsck tool “fsck from util-linux 2.31.1” use the **-T** option.

### Run fsck on Linux Root Partition

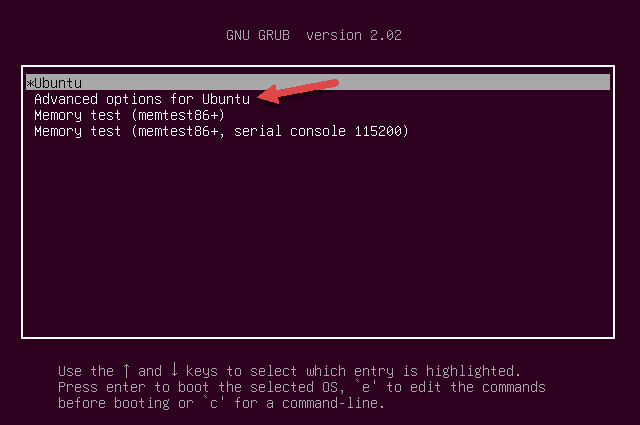
As we already mentioned, fsck cannot check root partitions on a running machine since they are mounted and in use. However, even Linux root partitions can be checked if you boot into recovery mode and run the **fsck check:**

1. To do so, power on or reboot your machine through the GUI or by using the terminal:

sudo reboot

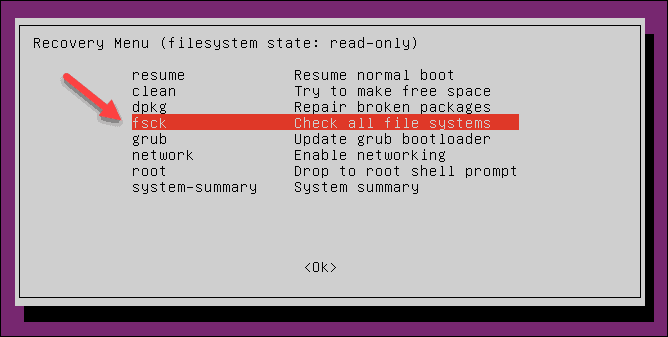
2. Press and hold the **shift**key during boot-up. The GNU GRUB menu appears.

3. Select **Advanced options for Ubuntu**.

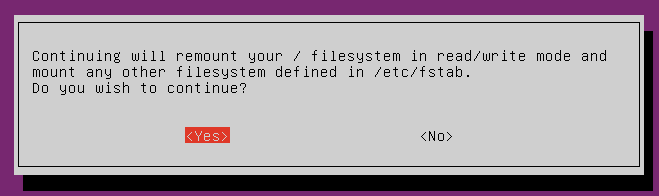


4. Then, select the entry with (recovery mode) at the end. Let the system load into the Recovery Menu.

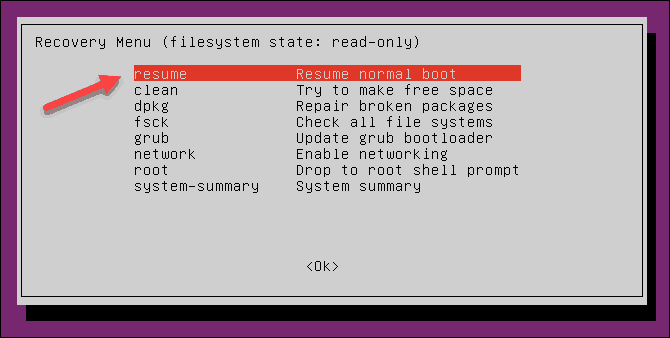
5. Select **fsck** from the menu.



6. Confirm by selecting **<Yes>**at the prompt.



7. Once finished, select **resume** at the recovery menu to boot up the machine.



## What if fsck is Interrupted?

You **should not interrupt** the fsck tool while it is in progress. However, if the process is interrupted, fsck will finish the ongoing check and then stop.

In case the utility found an error while the check was in process, it will not try to fix anything if interrupted. You can rerun the check next time and let it finish.

## fsck Linux Command Options Summary

To wrap up, below is the list of the options you can use with the **fsck Linux utility**.

| **Option** | **Description** |
| --- | --- |
| **-a** | **Try to repair filesystem errors automatically. There will be no prompts, so use it with caution.** |
| **-A** | **Check all filesystems listed in /etc/fstab.** |
| **-C** | **Show progress for ext2 and ext3 filesystems.** |
| **-f** | **Force fsck to check a filesystem. The tool checks even when the filesystem appears to be clean.** |
| **-l** | **Lock the device to prevent other programs from using the partition during the scan and repair.** |
| **-M** | **Do not check mounted filesystems. The tool returns an exit code 0 when a filesystem is mounted.** |
| **-N** | **Do a dry run. The output prints what the fsck would do without executing any actions. The warning or error messages are printed as well.** |
| **-P** | **Use to run a scan on multiple filesystems in parallel. It can cause issues, depending on your setup. Use with caution.** |
| **-R** | **Tell the fsck tool not to check the root filesystems when you use the -A option.** |
| **-r** | **Print device statistics.** |
| **-t** | **Specify which filesystems type(s) to check with fsck. Consult the man page for detailed information.** |
| **-T** | **Hide the title when the tool starts.** |
| **-y** | **Try to repair filesystem errors automatically during the check.** |
| **-V** | **Verbose output.** |

**Conclusion**

Now you know **how to use fsck Linux command to check and repair filesystems**.