Waiting room:

Linux file system basics

Source **Red Hat Linux 8.0: The Official Red Hat Linux System Administration Primer**

Partitions

Partitions

Partitions are a way of dividing a disk drive's storage into distinctly separate regions. Using partitions gives the system administrator much more flexibility in terms of allocating storage.

Because they are separate from each other, partitions can have different amounts of space utilized, and that space will in no way impact the space utilized by other partitions. For example, the partition holding the files comprising the operating system will not be affected even if the partition holding the users' files becomes full. The operating system will still have free space for its own use.

Partition Attributes

Partitions are defined by the following attributes:

- Partition geometry
- Partition type
- Partition type field

Geometry

A partition's geometry refers to its physical placement on a disk drive. In order to understand geometry, we must first understand **how data is stored on a disk drive**.

As the name implies, a disk drive contain one or more disks coated with a magnetic material. It is this material that actually stores the data. The surface of each disk is read and written by a *head*, similar in function to the head in a cassette tape recorder.

The head for each disk surface is attached to an *access arm*, which allows the heads to sweep across the surfaces of the disks. As the disks rotate under the heads, the section of the disks under the heads at any given position of the access arm make up a *cylinder* (when only one disk surface is involved, this circular slice of magnetic media is known as a *track*). Each track making up each cylinder is further divided into *sectors*; these fixed-sized pieces of storage represent the smallest directly-addressable items on a disk drive. There are normally hundreds of sectors per track. Present-day disk drives may have tens of thousands of cylinders, representing tens of thousands of unique positions of the access arm.

Partitions are normally specified in terms of cylinders, with the partition size is defined as the amount of storage between the starting and ending cylinders.

Partition Type

The partition type refers to the partition's relationship with the other partitions on the disk drive. There are three different partition types:

- Primary partitions
- Extended partitions
- Logical partitions

Primary Partitions

Primary partitions are partitions that take up one of the four primary partition slots in the disk drive's partition table.

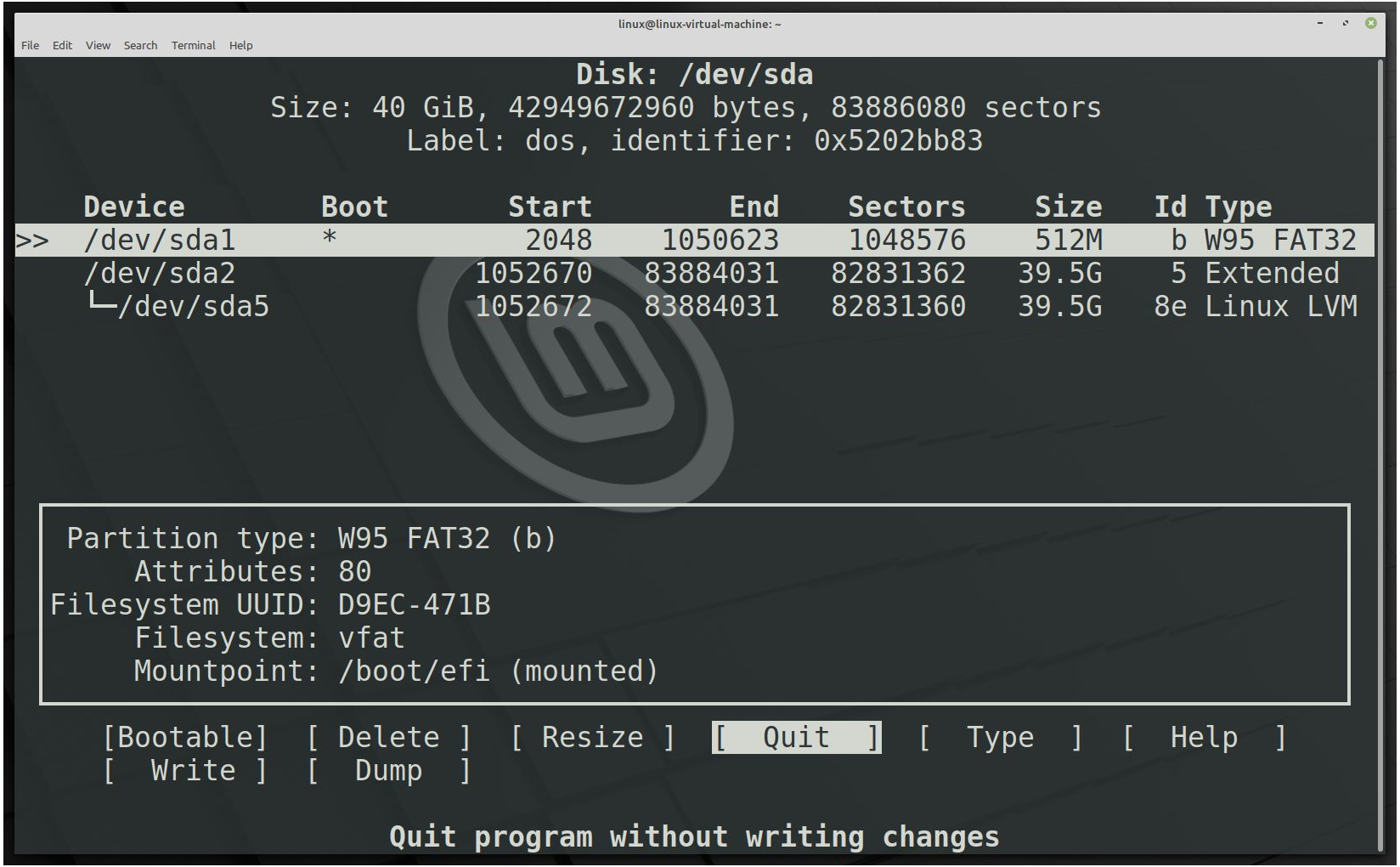
Extended Partitions

Extended partitions were developed in response to the need for more than four partitions per disk drive. An extended partition can itself contain multiple partitions, greatly extending the number of partitions possible.

Logical Partitions

Logical partitions are those partitions contained within an extended partition.

cfdisk utility


linux@linux-virtual-machine: ~
File Edit View Search Terminal Help
Disk: /dev/sda
Size: 40 GiB, 42949672960 bytes, 83886080 sectors
Label: dos, identifier: 0x5202bb83
Device Boot Start End Sectors Size Id Type
>> /dev/sda1 * 2048 1050623 1048576 512M b W95 FAT32
/dev/sda2 1052670 83884031 82831362 39.5G 5 Extended
└─/dev/sda5 1052672 83884031 82831360 39.5G 8e Linux LVM
Partition type: W95 FAT32 (b)
Attributes: 80
Filesystem UUID: D9EC-471B
Filesystem: vfat
Mountpoint: /boot/efi (mounted)
[Bootable] [Delete] [Resize] [Quit] [Type] [Help]
[Write] [Dump]
Quit program without writing changes

What is File System ?

A disk drive by itself provides a place to store data, and nothing more. In fact, by itself, the only way to access data on a hard drive is by either specifying the data physical location (in terms of cylinder, head, and sector), or by its logical location (the 65,321st block) on the disk.

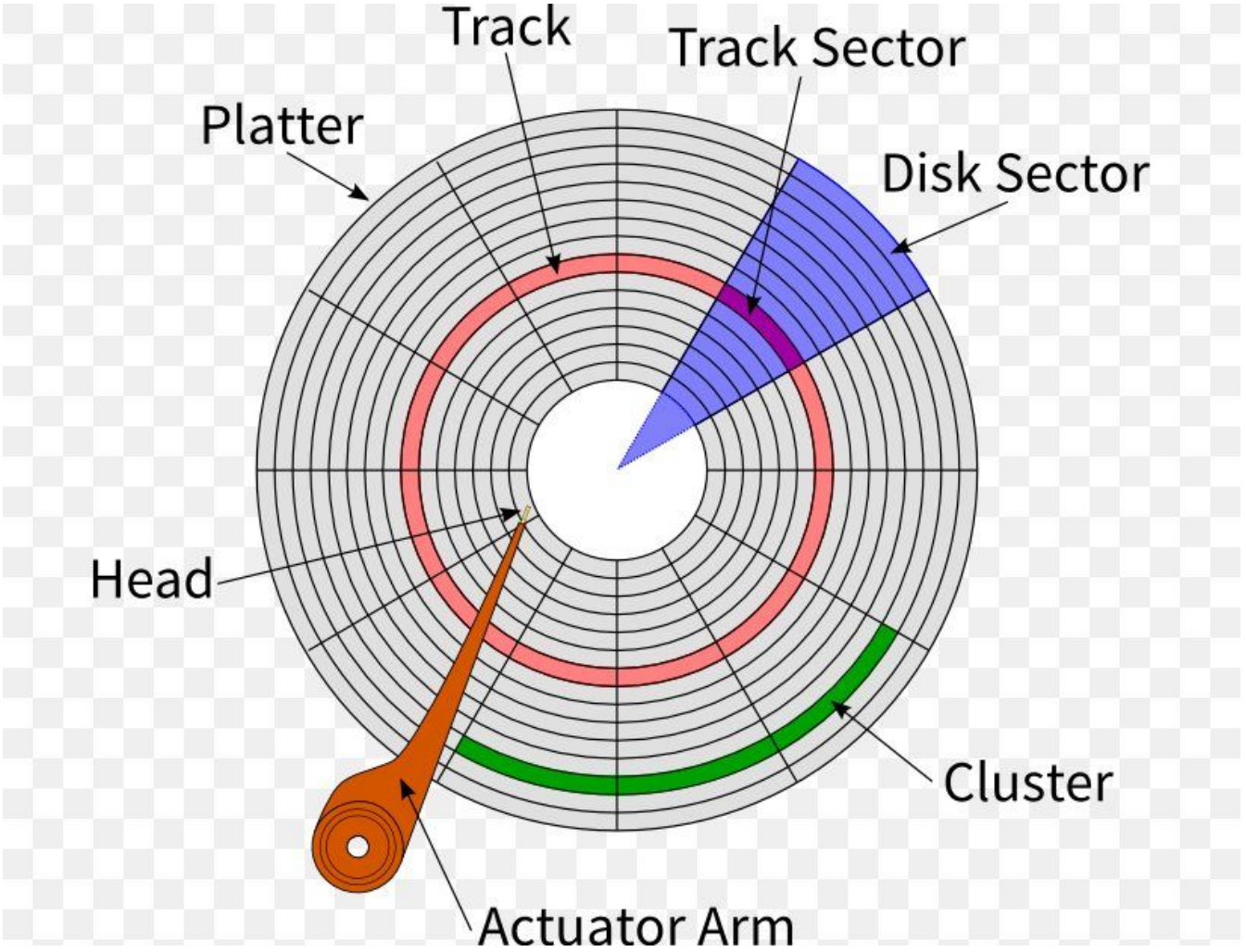
What is needed is a way to more easily keep track of things stored on hard drives; a way of filing information in an easily-accessible way.

HDD
3.5”

Shock resistant up to 350g/2ms

SSD
2.5”

Shock resistant up to 1500g/0.5ms


Track
Track Sector
Platter
Disk Sector
Head
Cluster
Actuator Arm

An Overview of File Systems

File systems, as the name implies, treat different sets of information as files. Each file is separate from every other. Over and above the information stored within it, each file includes additional information:

- The file's name
- The file's access permissions
- The time and date of the file's creation, access, and modification.

..

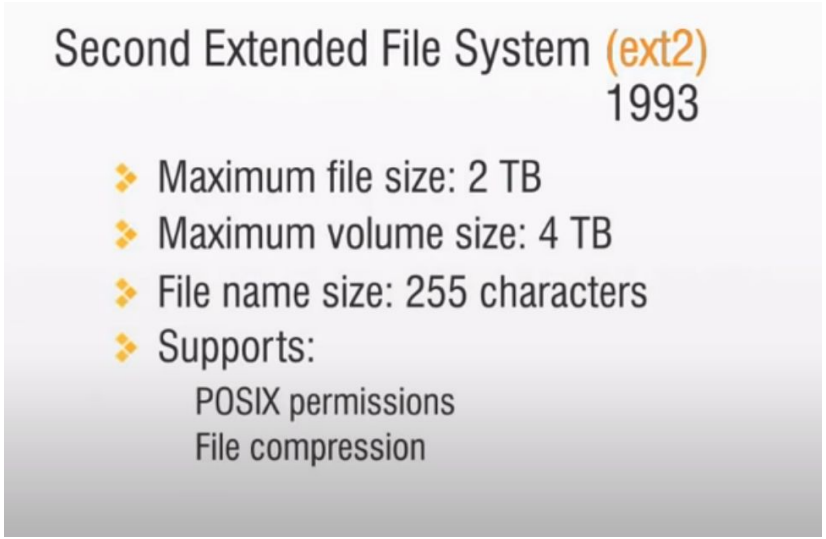
While file systems in the past have included no more complexity than that already mentioned, present-day file systems include mechanisms to make it easier to group related files together. The most commonly-used mechanism is the directory. Often implemented as a special type of file, directories make it possible to **create hierarchical structures of files and directories**.

File Systems frequently used by different Linux

EXT2

Until recently, the ext2 file system has been the standard Linux file system for Red Hat Linux. As such, it has received extensive testing, and is considered one of the more robust file systems in use today.

However, there is no perfect file system, and ext2 is no exception. One problem that is very commonly reported is that an ext2 file system must undergo a lengthy file system integrity check if the system was not cleanly shut down. While this requirement is not unique to ext2, the popularity of ext2, combined with the advent of larger disk drives, meant that file system integrity checks were taking longer and longer. Something had to be done.



EXT3

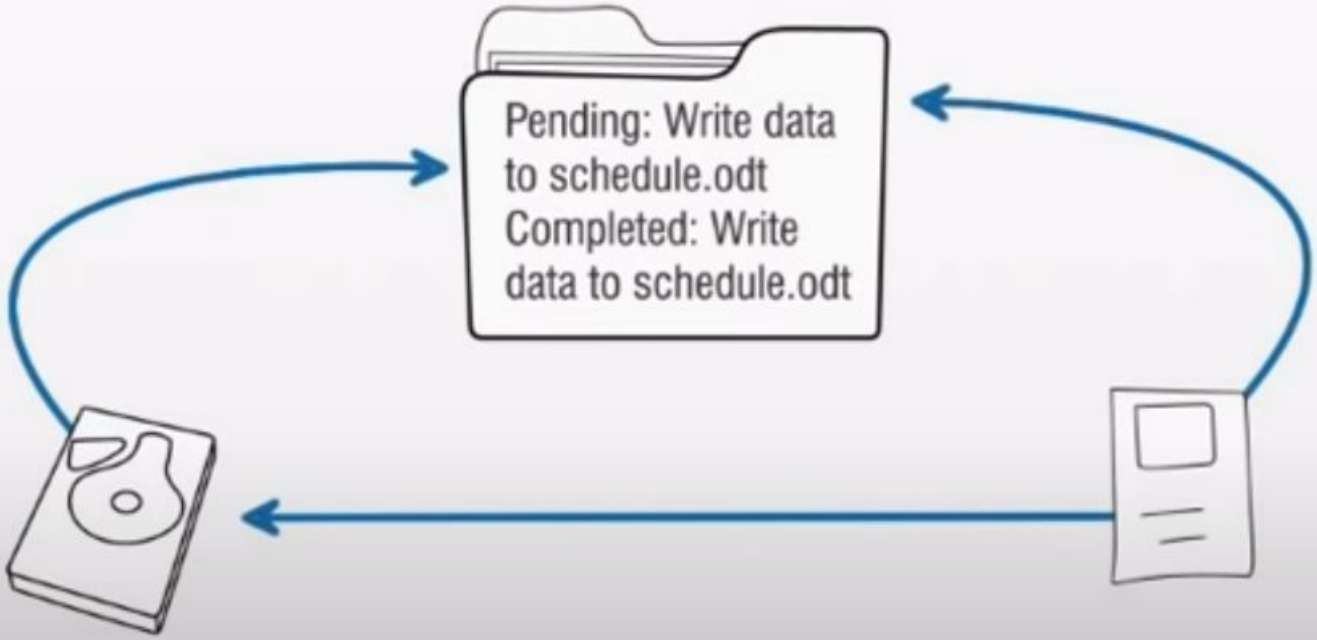
The ext3 file system builds upon ext2 by adding journaling capabilities to the already-proven ext2 codebase. As a journaling file system, ext3 always keeps the file system in a consistent state, eliminating the need for file system integrity checks.

This is accomplished by writing all file system changes to an on-disk journal, which is then flushed on a regular basis. After an unexpected system event (such as a power outage or system crash), the only operation that needs to take place prior to making the file system available is to process the contents of the journal; in most cases this takes approximately one second.

Because ext3's on-disk data format is based on ext2, it is possible to access an ext3 file system on any system capable of reading and writing an ext2 file system (without the benefit of journaling, however). This can be a sizable benefit in organizations where some systems are using ext3 and some are still using ext2.

Even the disk performance might degrade a little !! Due to journaling
, Still system administrator prefers EXT3 over EXT2

Journaling


Pending: Write data
to schedule.odt
Completed: Write
data to schedule.odt

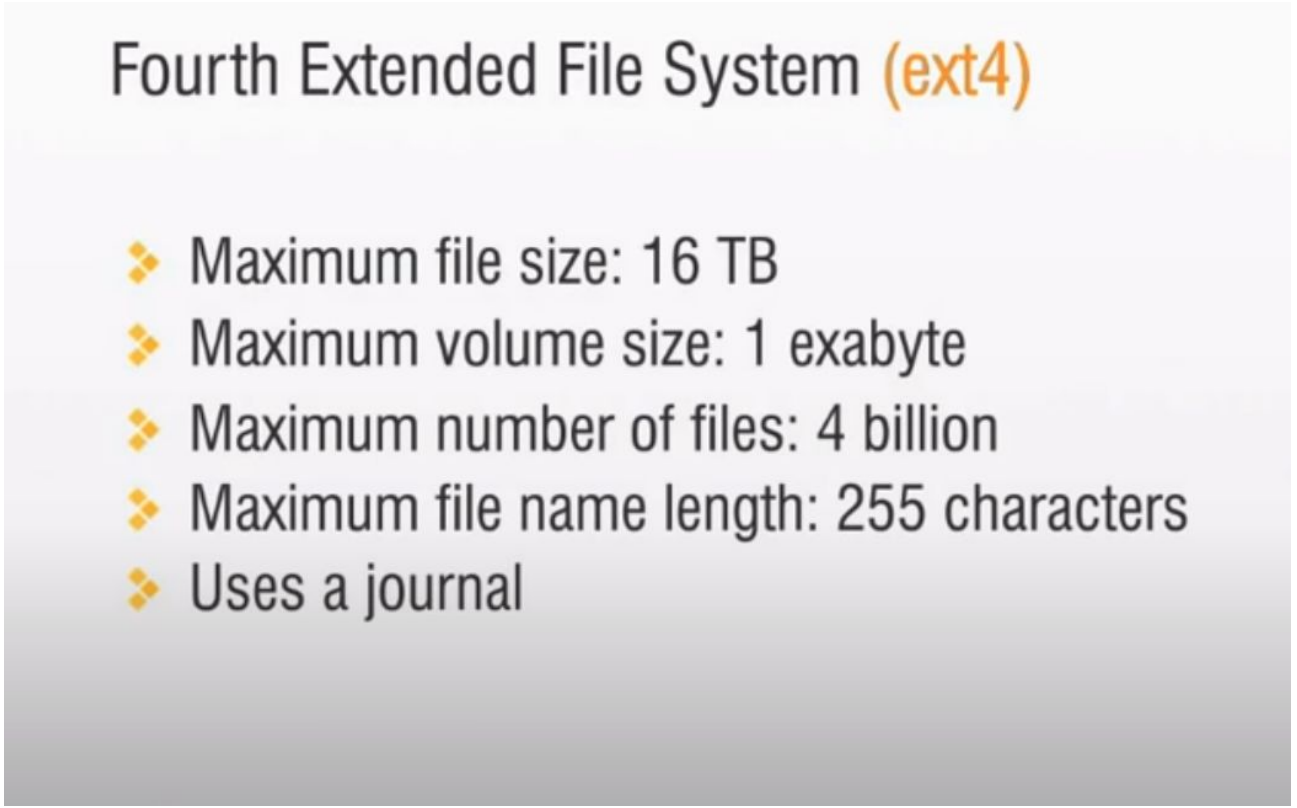
NFS

As the name implies, the Network File System (more commonly known as NFS) is a file system that may be accessed via a network connection. With other file systems, the storage device must be directly attached to the local system. However, with NFS this is not a requirement, making possible a variety of different configurations, from centralized file system servers, to entirely diskless computer systems.

However, unlike the other file systems discussed here, NFS does not dictate a specific on-disk format. Instead, it relies on the server operating system's native file system support to control the actual I/O to local disk drive(s). NFS then makes the file system available to any operating system running a compatible NFS client.

While primarily a Linux and UNIX technology, it is worth noting that NFS client implementations exist for other operating systems, making NFS a viable technique to share files with a variety of different platforms.

ext4


Fourth Extended File System (ext4)
Maximum file size: 16 TB
Maximum volume size: 1 exabyte
Maximum number of files: 4 billion
Maximum file name length: 255 characters
Uses a journal

VFAT

The vfat file system was first used by Microsoft's Windows® series of operating systems. An improvement over the msdos file system, file names on a vfat file system may be longer than msdos's 8.3. However, permissions and ownership still cannot be changed.

Mounting File Systems

In order to **access any file system**, it is first necessary to *mount* it. By mounting a file system, you direct Red Hat Linux to make a specific device (and partition) available to the system. Likewise, when access to a particular file system is no longer desired, it is necessary to *unmount* it.

In order to mount any file system, two pieces of information must be specified:

- A device file representing the desired disk drive and partition
- A directory under which the mounted file system will be made available (otherwise known as a *mount point)*

Demonstration

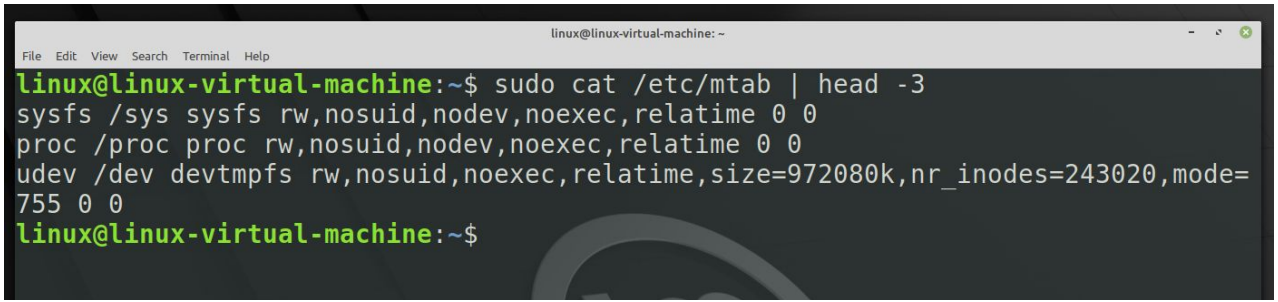
Seeing What is Mounted

In addition to mounting and unmounting disk space, it is possible to see what is mounted. There are several different ways of doing this:

- Viewing /etc/mtab
- Viewing /proc/mounts
- Issuing the df command

Viewing /etc/mtab

The file /etc/mtab is a normal file that is updated by the mount program whenever file systems are mounted or unmounted. Here is a sample /etc/mtab:

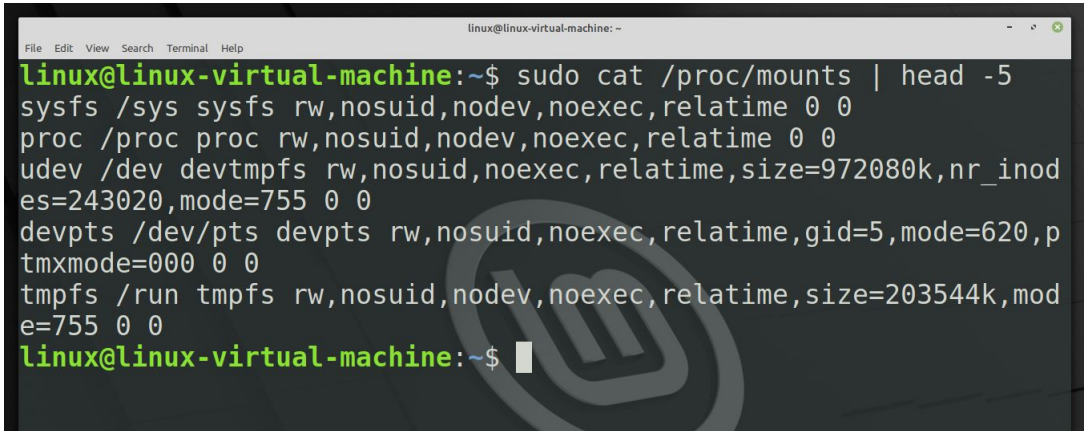


Each line represents a file system that is currently mounted and contains the following fields (from left to right):

- The device specification
- The mount point
- The file system type
- Whether the file system is mounted read-only (ro) or read-write (rw), along with any other mount options
- Two unused fields with zeros in them (for compatibility with /etc/fstab)

Viewing /proc/mounts

The /proc/mounts file is part of the proc virtual file system. As with the other files under /proc/, mounts does not exist on any disk drive in your Linux system. Instead, these files are representations of system status made available in file form. Using the command cat /proc/mounts, we can view /proc/mounts:

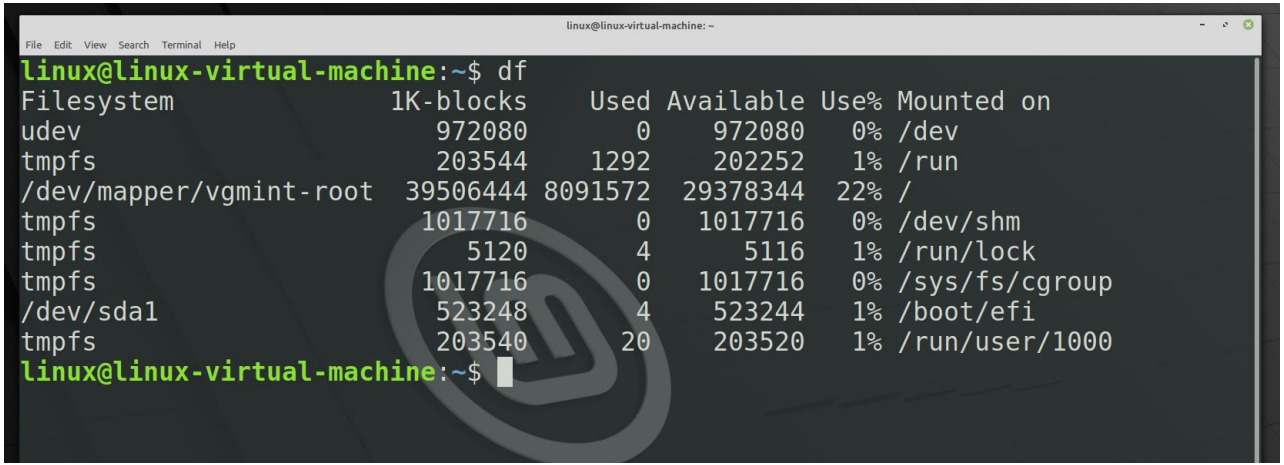


As we can see from the above example, the format of /proc/mounts is very similar to that of /etc/mtab. There are a number of file systems mounted that have nothing to do with disk drives. Among these are the /proc/ file system itself (along with two other file systems mounted under /proc/), pseudo-ttys, and shared memory.

The df Command

While using /proc/mounts will let you know what file systems are currently mounted, it does little beyond that. Most of the time you will be more interested in one particular aspect of the file systems that are currently mounted:

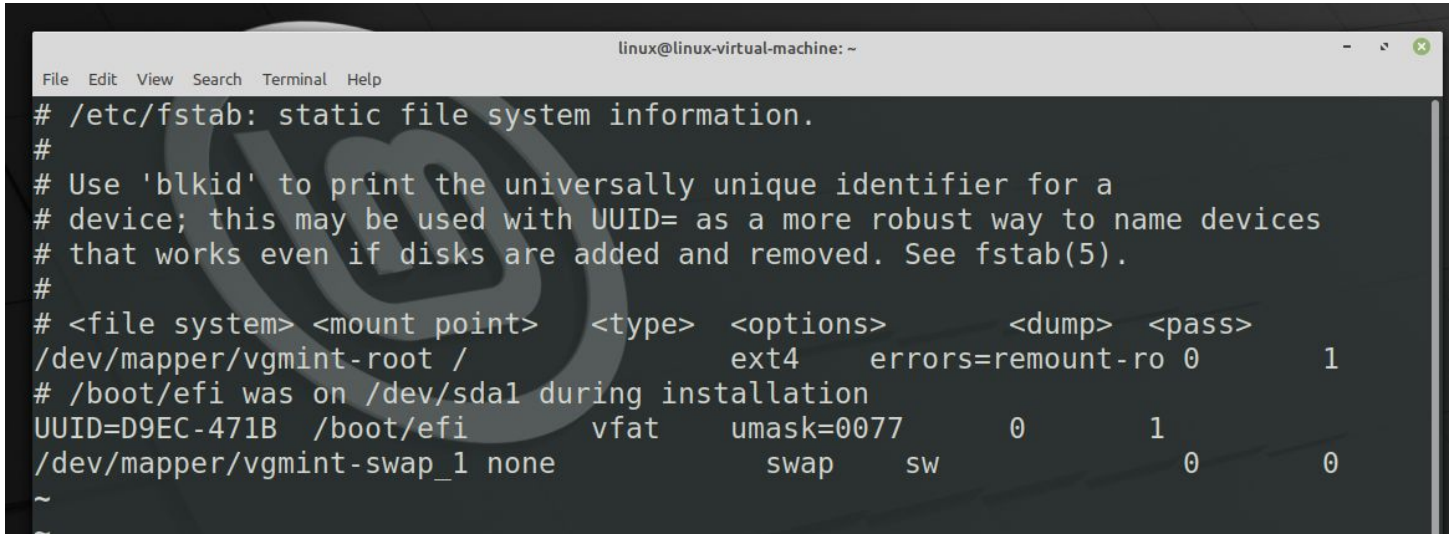
The amount of free space on them.



Mounting File Systems Automatically with /etc/fstab

When a Red Hat Linux system is newly-installed, all the disk partitions defined and/or created during the installation are configured to be automatically mounted whenever the system boots. However, what happens when additional disk drives are added to a system after the installation is done? The answer is "nothing" because the system was not configured to mount them automatically. However, this is easily changed.

The answer lies in the /etc/fstab file. This file is used to control what systems are mounted when the system boots, as well as to supply default values for other file systems that may be mounted manually from time to time. Here is a sample /etc/fstab file:



Each line represents one file system, and contains the following fields:

- File system specifier — For disk-based file systems, either a device file, or a device label specification
- Mount point — Except swap partitions, this field specifies the mount point to be used when the file system is mounted
- File system type — The type of file system present on the specified device (note that auto may be specified to select automatic detection of the file system to be mounted, which is handy for CD-ROMs and diskette drives)
- Mount options — A comma-separated list of options that can be used to control mount's behavior
- Dump frequency — If the dump backup utility is used, the number in this field will control dump's handling of the specified file system
- File system check order — Controls the order in which the file system checker fsck checks the integrity of the file systems.

An Overview of Linux file system Layout

In Linux, every file and device on the system resides under the “root” directory, which is denoted by a starting “/”.

Note: This is different from the default administrative user, which is also called “root”. It is also different from the default administrative user’s home directory, which is located at “/root”.

/bin

This directory contains basic commands and programs that are needed to achieve a minimal working environment upon booting. These are kept separate from some of the other programs on the system to allow you to boot the system for maintenance even if other parts of the filesystem may be damaged or unavailable.

If you search this directory, you will find that both `ls` and `pwd` reside here. The `cd` command is actually built into the shell we are using (bash), which is in this directory too.

/boot

This directory contains the actual files, images, and kernels necessary to boot the system. While `/bin` contains basic, essential utilities, `/boot` contains the core components that actually allow the system to boot.

If you need to modify the bootloader on your system, or if you would like to see the actual kernel files and initial ramdisk (initrd), you can find them here. This directory must be accessible to the system very early on.

/dev

This directory houses the files that represent devices on your system. Every hard drive, terminal device, input or output device available to the system is represented by a file here. Depending on the device, you can operate on the devices in different ways.

For instance, for a device that represents a hard drive, like `/dev/sda`, you can mount it to the filesystem to access it. On the other hand, if you have a file that represents a line printer like `/dev/lpr`, you can write directly to it to send the information to the printer.

/etc

This is one area of the filesystem where you will spend a lot of time if you are working as a system administrator. This directory is basically a configuration directory for various system-wide services.

By default, this directory contains many files and subdirectories. It contains the configuration files for most of the activities on the system, regardless of their function. In cases where multiple configuration files are needed, many times a application-specific subdirectory is created to hold these files. If you are attempting to configure a service or program for the entire system, this is a great place to look.

/home

This location contains the home directories of all of the users on the system (except for the administrative user, root). If you have created other users, a directory matching their username will typically be created under this directory.

Inside each home directory, the associated user has write access. Typically, regular users only have write access to their own home directory. This helps keep the filesystem clean and ensures that not just anyone can change important configuration files.

Within the home directory, that are often hidden files and directories (represented by a starting dot) that allow for user-specific configuration of tools. You can often set system defaults in the `/etc` directory, and then each user can override them as necessary in their own home directory.

/lib

This directory is used for all of the shared system libraries that are required by the `/bin` and `/sbin` directories. These files basically provide functionality to the other programs on the system. This is one of the directories that you will not have to access often.

/lost+found

This is a special directory that contains files recovered by `/fsck`, the Linux filesystem repair program. If the filesystem is damaged and recovery is undertaken, sometimes files are found but the reference to their location is lost. In this case, the system will place them in this directory.

In most cases, this directory will remain empty. If you experience corruption or any similar problems and are forced to perform recovery operations, it’s always a good idea to check this location when you are finished.

/media

This directory is typically empty at boot. Its real purpose is simply to provide a location to mount removable media (like cds). In a server environment, this won’t be used in most circumstances. But if your Linux operating system ever mounts a media disk and you are unsure of where it placed it,

/mnt

This directory is similar to the `/media` directory in that it exists only to serve as a organization mount point for devices. In this case, this location is usually used to mount filesystems like external hard drives, etc.

This directory is often used in a VPS environment for mounting network accessible drives. If you have a filesystem on a remote system that you would like to mount on your server, this is a good place to do that.

/opt

This directory's usage is rather ambiguous. It is used by some distributions, but ignored by others. Typically, it is used to store optional packages. In the Linux distribution world, this usually means packages and applications that were not installed from the repositories.

For instance, if your distribution typically provides the packages through a package manager, but you installed program X from source, then this directory would be a good location for that software. Another popular option for software of this nature is in the `/usr/local` directory.

/opt

This directory's usage is rather ambiguous. It is used by some distributions, but ignored by others. Typically, it is used to store optional packages. In the Linux distribution world, this usually means packages and applications that were not installed from the repositories.

For instance, if your distribution typically provides the packages through a package manager, but you installed program X from source, then this directory would be a good location for that software. Another popular option for software of this nature is in the `/usr/local` directory.

/root

This is the home directory of the administrative user (called “root”). It functions exactly like the normal home directories, but is housed here instead.

/run

This directory is for the operating system to write temporary runtime information during the early stages of the boot process. In general, you should not have to worry about much of the information in this directory.

/sbin

This directory is much like the `/bin` directory in that it contains programs deemed essential for using the operating system. The distinction is usually that `/sbin` contains commands that are available to the system administrator, while the other directory contains programs for all of the users of the system.

/selinux

This directory contains information involving security enhanced Linux. This is a kernel module that is used to provide access control to the operating system. For the most part, you can ignore this.

/tmp

This is a directory that is used to store temporary files on the system. It is writable by anyone on the computer and does not persist upon reboot. This means that any files that you need just for a little bit can be put here. They will be automatically deleted once the system shuts down.

/usr

This directory is one of the largest directories on the system. It basically includes a set of folders that look similar to those in the root `/` directory, such as `/usr/bin` and `/usr/lib`. This location is basically used to store all non-essential programs, their documentation, libraries, and other data that is not required for the most minimal usage of the system.

/var

This directory is supposed to contain variable data. In practice, this means it is used to contain information or directories that you expect to grow as the system is used.

For example, system logs and backups are housed here. Another popular use of this directory is to store web content if you are operating a web server.