# 8.5.2 Mounting a File System

Click one of the buttons to take you to that part of the video.

Mount a File System 0:00-0:22

In this demonstration we're going to review how to mount a file system. Specifically, we're going to talk about how to mount a file system from the command line.

We're going to talk about how to unmount a file system from the command line. We'll talk about adding entries to the fstab file. Then we will end this demonstration by discussing how to mount and unmount LVM logical volumes.

Mount File Systems From the Command Line 0:23-5:07

To begin with, I need to switch my user account before I can do much of anything. Let's run the 'fdisk' command with the '-l' option on '/dev/sdb' to view a list of the partitions that have currently been defined on this second hard disk drive in my Linux system. sda contains my Linux operating system, my user home directories and such.

We're not going to play with that hard disk at all. Instead, we're going to focus on the partitions that are stored on this second hard disk drive in my system--sdb.

Notice that we have sdb1, which is a standard Linux data partition. We have an extended partition and within that extended partition we created two logical partitions. The first one is a standard Linux data partition. The second one is a swap partition.

We won't be playing with that partition. We will focus on this partition and this partition instead.

Previously, I already created file systems on both of these partitions. This partition has an ext4 file system created and this one has an ext3 file system created. The partitions have been defined.

The file systems have been created on the partitions, but we still can't save data on them because they're not mounted currently in the Linux file system. Let's fix that.

Understand on Linux, file systems are mounted within directories in the file system hierarchy. They are not assigned drive letters as you would see in Windows. In Windows, if you added a hard disk with two partitions on it to the system, by default you would see two drive letters added to Windows Explorer, maybe drive D and drive E.

You would access those partitions by accessing drive letter D or drive letter E. It does not work that way in Linux. We have a single file system. All of our partitions are mounted somewhere in that file system.

What we want to do here is go into the '/mnt' directory; mnt is short for mount. Most distributions provide a mount directory by default. Its purpose is to do as its name implies. It's a place to mount file systems.

We could mount one of these partitions directly into the /mnt directory and that would work fine, but then we would have no place to mount the second one. What we want to do is create two subdirectories within /mnt and mount this partition in one and this partition in the other.

This partition is a bigger one. It's designed to be used for shared data among many users, so let's create a directory called 'shared' within /mnt. We also need a directory for this partition which is designed to hold proprietary data that will have restricted access.

Let's mount it in a directory named 'private'. We do an 'ls' command. We can now see that we have private and shared defined within the mount directory.

There are two different ways we can mount these partitions, these files systems, in these directories. One is to do it from the command line with the mount command. For example, I can run 'mount' and then I enter '-t', followed by the type of file system on the partition we are going to mount. In this case, we are going to mount sdb1, which has an 'ext4' file system.

By the way, if you forget to use the -t option, it will usually still work, because mnt--if you don't specify a file system type--will take a look at the partition and say, "Oh, that's a ext4 file system" and mount it appropriately.

Then we have to specify the name of the device file for the partition that we want to mount '/dev /sdb1' in this case. Then we have to specify where we want it mounted. In this case, we want it mounted in '/mnt/shared'. Hit Enter, and now this partition is mounted in this directory.

If I were to switch into this directory at the command prompt, such as like this. I have just switched over to a completely different hard disk drive. I was on sda, on a partition on sda. By changing directories, I've switched over to this partition on sdb.

Go back up again. Therefore, any data that I store in the shared directory will actually be stored on this hard disk drive instead of the sda hard disk drive.

Let's do the same thing. Do 'mount -t ext3' this time because we created an ext3 file system on this partition, '/dev/sdb5'. This time we want it mounted in '/mnt/private'. Now you can check and see what the status is of your mounted file systems using the mount command with no options, just enter 'mount'.

When we do, we see down here, /dev/sdb1 is mounted in the directory we specified with an ext4 file system and /dev/sdb5 is mounted in /mnt/private with an ext3 file system.

---

## Unmount File Systems From the Command Line 5:08-7:24

If you have a file system mounted in a directory somewhere and you decide, "I don't need that anymore; I don't want it mounted," you can unmount it. This is commonly done when you're using removable external storage devices like a thumb drive or an external USB drive, and so on.

Let's first unmount this file system right here, /dev/sdb1. This will unmount it from this directory, making any data that's on that partition inaccessible. Be aware of that before you unmount a file system.

I enter umount; this is the point of confusion with new Linux system administrators, notice that it is not unmount, it is umount. Some Linux distributions do define aliases by default, because this is such a common mistake. for unmount, that actually points to the umount command--some don't, though.

Let's enter 'umount' and then we enter the name of the device file for the partition, '/dev/sdb1', press Enter. If I type 'mount', we see that sdb1 is no longer mounted in the /shared directory.

Let's do the same thing with sdb5, but we're going to do it slightly differently. With the umount command, you can either specify the device file or you can specify the name of the directory where that partition is currently mounted. In this case, we're going to unmount sdb5 by entering 'umount /mnt/private'.

If I do a 'mount' command, we see that they are both currently now mounted. Again, if I do this, any data that was saved on those partitions will be gone. I can still switch into the /shared directory, for example, and I can do an 'ls' command. Now because we've not mounted that partition in that directory, I'm actually looking at data on sda.

This is a confusing thing for a lot of new Linux users. Once that umount command has been executed, the /shared directory just becomes a normal, regular directory on the default partition where it's located. If I put files in this directory and then mount a partition into that same directory, the files that were stored originally will not be accessible anymore.

They're still saved on sda; they're just not accessible because we've mounted a different partition into that directory. As soon as you umount that partition, then the original files you have saved in that directory would become available.

---

## Add an Entry Using the fstab File 7:25-9:53

Be aware if we use the mount command to mount a file system, that mounting will only be persistent as long as the system stays running. As soon as you reboot, the mounting will be gone and you'll have to redo it manually.

Therefore, if you want the file system to be mounted persistently across system reboots, you're going to have to add an entry to the /etc/fstab file. To do this we enter 'vi', open the vi editor, and then specify, '/etc/fstab'. There's already some default file mounting specified here in the file.

We want to press the Insert key. Add a new line and add an additional entry that will mount sdb1 and sdb5 every time the system boots.

Let's do that. We enter, first of all, the device filename for the partition. We'll start with '/dev/sdb1', then a tab, and then we enter the name of the directory where we want it mounted '/mnt/shared', then another tab.

Then we specify the file system type we want to use, 'ext4', then another tab. Then we specify 'defaults'. defaults specifies that the basic default mount options will be used, which essentially means that they'll be mounted as read write so that you can both read data from it and save data to it at the same time.

Then we enter a '1' and then a space and then a '2'. This 1 specifies that the file system can be dumped. If you have a zero, such as right here, it means that the file system is not allowed to be dumped. The 2 specifies the order in which the file systems should be checked whenever the system boots up.

The root partition, as you can see right here, should have a value of 1 in this field. All other file systems should have a 2, because the root file system is the one where the operating system is installed. You can see that I'm going to use a 2 here and this file system also has a 2 because it is not the root file system.

Let's add an additional entry for our other partition, '/dev/sdb5'. We want it mounted in '/mnt/private'. It uses the 'ext3' file system. We'll use our 'defaults' mount options, and we'll use the exact same numbers for dumping and for file system checking.

We'll hit Escape, write changes to the file. Because of the entries we've added to the fstab file, those two partitions will automatically be mounted in those directories whenever the system boots.

---

## Mount LVM Logical Volumes 9:54-12:06

At this point we need to shift gears and talk about how to accomplish these same tasks with LVM volumes instead of disk partitions. This Linux system, unlike the one we were just working on, has two additional hard disks installed in the system--sdb and sdc.

These two disks were used to create logical LVM volumes. Switch to our root user account here. We'll run the 'lvscan' command. We see that we have a volume group named data and within data, we have two volumes defined--shared and private.

These other two LVM volumes were created by default when the system was initially installed. We are not worried about those.

We do want to work with these two because they have been created. They both have file systems defined on them--ext4 file systems--but they're not mounted, so we can't access them. We can mount these into the file system just in the same way we did a standard disk partition.

First of all, let's use 'mkdir' to make two directories in '/mnt'. First one named 'shared'. Let's do the same thing again. Make it named 'private'.

One point of confusion that sometimes comes up is whether or not the directory name specified here has to match the volume name, the LVM volume name, and it does not. I'm just doing it for convenience purposes. You can name these directories whatever you want.

Now you do the 'mount' command again. We use the '-t' option just like we did before, then 'ext4', and then we specify the device file for the LVM volume, '/dev'.

Now things are a little different here when we're dealing with LVM. Instead of specifying a physical partition, such as /dev/sdb1, we instead specify /dev and then the name of the volume group where the volume resides--in this case 'data', and then the name of the volume itself--in this case 'shared'.

We have to specify where we want to mount these in the file system. Let's do it in '/mnt/shared'. Let's do the same thing for the private volume. Change the mount point.

Now if we run the mount command. we see that the two logical volumes have been mounted in my Linux file system. I switch to those directories and I'm able to store data within those volumes.

---

## Summary 12:07-12:23

That's it for this demonstration. In this demo we talked about how to mount a Linux file system. We talked about mounting file systems from the command line. We talked about unmounting file systems from the command line. We then talked about how to add an entry to the fstab file to make the mounting persistent across reboots. Then we ended this discussion by looking at how to mount LVM logical volumes.

---