

## 8.4.4 Creating a File System

---

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

Create a File System 0:00-0:18

In this demonstration, we're going to practice creating file systems. First we're going to look at creating file systems on standard disk partitions, then we're going to look at creating swap file systems, and then we'll end this demonstration by talking about how to create file systems on LVM logical volumes.

---

Create File Systems on Standard MBR Disk Partitions 0:19-2:56

Let's begin by talking about how to create a file system on standard disk partitions. On this particular Linux system, I have two hard disks installed. sda contains my operating system, but sdb is basically blank. Although it does have some partitions already defined, but these partitions do not have file systems; hence, there is no data currently being stored.

Let's go ahead and look at the partition setup on sdb. First, I do have to switch to my root user account and let's run 'fdisk -l' to list and then the name of the device we want to look at: '/dev/sdb'. You can see that I have a standard MBR partition defined for sdb1. 10 GB in size. Notice that my second disk partition is an extended partition, and within that partition I would define two 5 GB standard Linux volumes--sdb5 and sdb6.

Let's begin by creating an ext4 file system on this first partition, on sdb1. We do this using the mkfs command. Before we run mkfs, let's go ahead and look at its man page so we can see some of the different options that we can use.

One of the key ones that you need to be familiar with is the type option right here. As noted here, the -t option allows us to specify what type of file system we want to create. Do we want to create an ext2, ext3, ext4, Reiser, or so on.

We'll get out of the man page by pressing 'q' and let's run 'mkfs' and use the '-t' option to specify the type of file system we want to create on sdb1. Let's create an ext4 file system on sdb1, and the file system is now created. With the file system created on 'sdb1', we can mount it in the directory within the Linux file system hierarchy, and we can begin saving data on it.

Let's do the same thing with sdb5, only this time let's create an ext3 file system on this partition. We are done. At this point, we've created an ext4 file system on sdb1 and we've created an ext3 file system on sdb5. Again, with the file system created on sdb5, we can now mount it somewhere in a directory in the Linux file system and use it to store data.

---

Create Swap File Systems 2:57-5:20

Let's shift gears a little bit and talk about creating swap file systems. Remember, the swap file system is used for virtual memory by the Linux system. This system already has a swap file system defined on sda; however, we can add additional swap file systems. Doing so, especially on a hard disk where the operating system is not installed, can actually help improve system performance.

We're going to do that using sdb6. However, before we can do that, we're going to have to go in and modify the partition type in fdisk, because notice here that sdb6 is currently set to a standard Linux data partition--type 83. That won't work for swap.

Let's do 'fdisk /dev/sdb'. Let's print the partition table. Let's use the 'l' option to review the list of partition types. Currently, sdb6 is set to 83. We want to switch it to type 82, or a Linux swap file system. Let's enter 'p' again. Print our partition table.

Enter 't' to toggle the file system type. We want to toggle the file system type of number 6, sdb6. Let's change it to 82, for Linux swap. If we print the partition table, we should see that the type has changed, and it has.

Let's write our changes to disk. Let's run 'partprobe' just in case the Linux kernel did not pick up the change in the partition type. I don't think we needed to do that, but it's not a bad idea to do it any way just in case.

To create the swap file system, instead of using mkfs, we're going to use a different command--mkswap. We enter 'mkswap' followed by the name of the partition device file '/dev/sdb6' of the partition that we defined as a swap partition with fdisk. Press Enter.

The swap file system has now been created on this partition and now we need to activate that swap partition. This is done using the 'swapon' command followed by the partition that we want to activate: '/dev/sdb6'.

At this point, we've taken a Linux system that uses standard MBR partitions and we've created ext4 file systems. We've created ext3 file systems and we've created swap file systems.

**Create File Systems on LVM Volumes 5:21-7:08**

We need to shift gears and look at how to accomplish the same task on a Linux system that does not use MBR partitions, but uses LVM logical volumes. This Linux system has several LVM logical volumes currently defined, but don't have file systems on them.

If I use the 'su' command to switch to my root user account first and then use the 'lvscan' command, we can see that I have two logical volumes--one named private and one named shared--that have been defined on the system, but currently they don't have the file system and, hence, we cannot store data on them.

Let's go ahead and fix that. Let's create ext4 file systems on both of these logical volumes. We do it, basically, in the same way that we do it with a system that uses MBR or GUID partitions for that matter.

We type 'mkfs -t' then the file system type, 'ext4', and then the name of the device. It's a little different here because we're dealing with LVM. When we're dealing with MBR partitions, we specify the device file for the partitions, such as /dev/sdb1.

Here, we have to specify, first of all, '/dev' and then we have to specify the name of the volume group that the logical volume resides in. In this case, the name of the volume group is 'data' and then we specify the name of the logical volume itself within that volume group. In this case, 'shared'. Enter.

Now we've created an ext4 file system on the shared logical volume. We'll do the same thing with private. We are done. We can now mount these two logical volumes in the file system and begin using them to store data.

---

**Summary 7:09-7:21**

That's it for this demonstration. In this demo, we talked about creating Linux file systems. We first talked about creating file systems on standard MBR disk partitions. We then talked about creating swap file systems and then we ended this demonstration by talking about how to create file systems on LVM volumes.

---

**Copyright © 2022 TestOut Corporation All rights reserved.**