2.9.2 File Management

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

File Management 0:00-0:35

A key skill that you need to have as a Linux administrator is knowing how to manage the files in the Linux file system from the command line. Let's begin by talking about how you create new files from the command line. From time to time, you're going to need to create new files in the file system. There are many different ways to do this.

One way would be to open up an application that creates new files, such as a text editor or an Office application, and do File > New > Add Contents to File, and then save it. That would create a new file in the file system.

The touch Command 0:36-1:59

However, there may be times when you just need to create a new file that doesn't have to have any contents in it. We'll add those later on. You can do this using the touch command at the shell prompt. To use touch, you simply enter 'touch' at the command prompt, followed by the name of the file that you want to create. When you do, touch will create a new, blank 0-byte file in the file system.

In this example, I'm creating a new file in the current directory named mylog. My current directory, notice, over here, is my home directory. It's going to create the mylog file in my user's home directory. As you can see here, I ran touch, specified a filename of mylog, and then I ran the ls -l command. And in my user's home directory, there is a new file. The new file's name is mylog. Notice that it is a 0-byte file. It's basically an empty shell. The filename is created, but there is absolutely nothing in it.

Just as with any other Linux command that manages files or directories, you can either use a relative path or an absolute path. In this example, I used a relative path because I didn't specify any path information for the file that I wanted to create. Therefore, the touch command just assumed that I wanted to create it in the current directory, which is my home directory represented by the tilde (~). If I wanted to create this file somewhere else in the file system, I would have had to provide the full path of that file.

The file Command 2:00-3:20

Next, let's discuss how you determine what type of file a particular file in the file system is. You can use the file command at the shell prompt to determine a file's type. This is an issue on Linux because on other operating systems, such as Windows, we use file extensions to identify what type of file we're actually dealing with. For example, on a Windows system, an executable file has a .exe extension, a text file has a .txt extension, a word file has a DOCX extension, and so on. This is not the case on Linux systems. On a Linux system, file extensions such as TXT are not commonly used. Therefore, it may not be immediately obvious from a file's name what kind of file it is we're looking at.

The file command can be used at the shell prompt to get information about a file. The output of the command will tell you the general file type--whether it's a directory, whether it's a named pipe, whether it's a standard file, a link file, or so on. Then it also provides detailed information about the file. To do this, it compares the file that you specified in the command with file signatures contained in the /usr/share/misc/magic directory, the /usr/share/misc/magic.mgc file, and the /etc/magic file.

The rm Command 3:21-5:10

With this in mind, let's now look at how you go about deleting files from the Linux file system.

You delete files from the file system using the rm command. You'll note that the rm command is the very same command that we use to delete directories; that's because rm is a very powerful deletion utility. It can be used to delete a file. It can delete an empty directory. It can even delete a populated directory. To use it to delete a file, we simply enter rm followed by the name of the file that we want to get rid of. Notice, in this example, that we had a file within my user's /home directory called logfile. I entered 'rm logfile' at the shell prompt, and then we ran the ls command again. The logfile file is missing; it should have been right there.

As with all the Linux commands, notice that, because I did not specify a full path to logfile, the rm assumed that the file I wanted to delete was in the current directory, which is my user's home directory. In fact, it was there, and it deleted it. If that file resided somewhere else in the file system, I would have had to provide the full path of that file. Be aware that you have to be very careful when you use the rm utility to delete a file, and that's because, by default, it does not prompt you to confirm a file deletion operation. It assumes that you really did mean to delete that file, and it does the dirty deed.

By the way, any file that you delete with rm does not get moved to the Recycle Bin. You can't get it back, in other words. If you delete a file or directory with rm, it is gone. If you want rm to prompt you before deleting file or a directory, include the -i option in the command. If you do, then the rm command will prompt you whether or not you want to really delete the file or not, and you'll have to respond before it will actually delete it.

The cp Command 5:11-6:02

Let's next discuss how you can copy and move files from the shell prompt. It's pretty straightforward because you use the exact same utilities to copy or move files that you use to copy or move directories. If you need to copy a file, you use the cp command. It will copy the file from one location in the file system to another.

In this example, I'm specifying that we copy a file named widgetdesign that's located in the /mnt directory to my user's /home directory, represented by the tilde (~). In this case, the widgetdesign file will be copied from /mnt to /home/rtracy. I run the 'ls' command and, sure enough, the file is now there. Remember, cp makes a copy of the file. That means this original copy of the file remains intact in the /mnt directory.

The mv Command: Move a File 6:03-6:48

In addition to copying files, you can also move files. To do this, you use the mv command. The mv command copies a file from its current location in the file system to its new location, and then deletes the original copy of the file.

In this case, we're going to work with the same file that we were working with previously with the cp command. It's the widgetdesign file. It's located in the /mnt directory. This time, instead of copying it, we're going to move it from this directory to the directory that we specify. In this case, I just put a tilde (~) in there to indicate my user's /home directory, /home/rtracy, in this case. When I do, the file is copied from /mnt to my user's /home directory, and then the original version of that file from /mnt is deleted.

The mv Command: Rename a File 6:49-7:24

It's important to note that the mv command can not only move files, but it can also be used to rename files. Essentially, what you do is you move the file from the directory where it currently resides to the same directory, but just with a different filename.

To do this, we enter 'mv' followed by the name of the file that we want to rename followed by the new name that we want to assign to it. In this case, I'm using relative paths. I want to rename the widgetdesign file in the current directory, which is my user's home directory, and I want to rename it twidgetproject from widgetdesign. I run 'mv widgetdesign widgetproject', and it renames the file.

The shred Command 7:25-8:35

There's one other command that we need to talk about before we end this lesson, and that is the shred command. The shred command is extremely important in today's security-conscious world. Here's the important thing that you need to remember: if you delete a file with the rm command, it will delete that file's inode, but it does not actually delete the file's data. This is a serious issue.

Suppose you use the rm command to delete a very sensitive file that contains proprietary information, and you don't want anybody else to get a hold of that information. Well, if you use the rm command to delete it, the file contents are actually still on the hard drive. Only the file's inode is actually removed. Therefore, if somebody were to gain access to that hard drive, they could still theoretically go in and pull that data off of the disk and reconstruct it. The tools needed to do this are widely available on the internet.

If you need to delete proprietary or otherwise confidential information from the Linux file system, then you should use the shred command instead of the rm command.

shred Options 8:36-9:59

To do this, you enter the shred -u' command followed by the name of the file that you want to remove from the hard disk drive. In this case, we're going to use shred to delete that widgetproject file that we just talked about. When you do this, shred will, of course, delete the inode of the file in question--just like the rm file does--but then it goes one step further. It will take a look and identify the area where the data that was associated with that inode actually physically resides on the hard disk drive, and then it will overwrite that area over, and over, and over in order to obliterate the data that was associated with the filename.

By default, it will do this 25 times. But you can also use the -n option in the command, as shown here, to specify a custom number of passes. In this case, we increased it to 35 passes. You can crank it up to a thousand passes if you wanted to, just to make sure that that file is completely obliterated.

Notice, in this command, we also included the -z option. The -z option specifies that shred overwrites that data area with a bunch of zeros. By doing this, it basically makes that area look like a blank spot on the hard disk drive. It essentially hides the fact that the file was shredded. Therefore, if some nefarious person were to poke around on the hard disk drive, it won't be obvious that the data area was shredded.

Summary 10:00-10:23

That's how you manage files in the Linux file system.

In this lesson, we talked about using touch to create a file. We talked about using the file command to determine a file's type. We talked about using the cp command to copy files. We talked about using the mv command to move files. We talked about using the rm command to delete files. Then we ended this lesson by talking about the shred command to not only delete a file, but also to erase any evidence that the file was even there, on the hard disk, in the first place.