2.8.3 Directory Management

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

Directory Management 0:00-0:20

In this lesson, we're going to review how you manage directories within the Linux file system. Let's begin by discussing how to create and delete directories. Let's first look at how you go about creating a new directory. Let's look at the shell commands that you can use to manage the directories within the Linux file system.

The mkdir Command 0:21-1:33

Let's first look at how you go about creating a new directory. This is done using the mkdir command, M-K-D-I-R. The mkdir command is a pretty simple command. All we type is 'mkdir' followed by the name of the directory we want to create. In this example, we created a new directory named docs. We used the ls command here to verify that the new directory was, in fact, created.

This is another example of using a relative path. We didn't specify where docs was supposed to go. We didn't provide any path information; therefore, the mkdir command assumed that it would go into the current directory, which, as you can see, is my user's /home directory.

Just as with the other commands that we've looked at in this lesson, you could, if you wanted to, use an absolute path with the mkdir command. An example is shown here. If we specify an absolute path and the mkdir command does not care what the current directory is, it makes the directory we specify in the path that we specify. In this example, we wanted to create a new directory named backup, and it needed to be created at the root of the file system. To do this, we entered 'mkdir /backup'.

The mkdir -p Option 1:34-2:28

Be aware that if you are creating a directory using an absolute path, like we just looked at, but the path that we specified in the command doesn't actually exist yet, then you can use the -p option with the mkdir command. The -p option tells the mkdir command to specify that the entire directory path within the command be created if it doesn't already exist. In this example, I've typed mkdir /backups/monthly/February.

If backups and monthly already existed, then it would just create this directory right here--February. But let's suppose that /backups and /monthly do not already exist. By adding the -p option, the mkdir command will create /backups, and then it will create /monthly, and then finally it will create the /February directory.

The cp Command 2:29-3:51

In addition to creating and viewing directories in the Linux file system using shell commands, you can also copy, move, or delete them. One command you can use to do this is the cp command, the copy command. This utility is used to copy directories or even entire directory structures from one location to another.

It's important to note that if you need to copy an entire directory structure, meaning the directory specified in any files or subdirectories within that directory, you need to use the -R option with the cp command. This specifies that the directory contents be copied recursively. In this example, we run 'cp -R', and then we specify the name of the directory that we want to copy. This is the source.

In this case, we're going to copy my entire home directory. I've specified ~/. I have to specify where I want that directory copied to. In this case, I've specified an absolute path where I want that directory copied. In this case, it's /backups/monthly/February. This command will copy my user's home directory along with any sub-directories and files within it to /backups/monthly/February.

It's important to note that because cp copies files, the original directory, right here, is not modified in any way. It's left intact.

The mv Command 3:52-4:42

In addition to cp, you can also use the mv command to manage directories. The mv command is used much like cp. It will copy a directory that you specify to a new location in the file system. But there is one key difference between mv and cp. The mv command moves the directories. That means that the original directory, the source directory, gets deleted. Use move with caution.

In this example, I'm going to move a directory within my /home directory named /docs. I'm going to move it from my /home directory to the absolute path specified here: /backups/monthly/February. Because we're moving this time instead of copying, this directory will be deleted from its source location after it's been moved to its new location, over here.

Rename a Directory 4:43-5:34

There's an interesting feature of the move command that you need to be familiar with. That is the fact that it cannot only move directories, but it can also rename directories. Basically, when we use move to rename directories, we're moving the directory to the same directory where it already existed and changing its name.

To do this, you simply type 'mv' followed by the source directory--the one that you want to rename--and then the new directory name that you want to apply. In this case, we're going to change the name of the docs directory in my user's home directory to documents.

Notice that we're not changing its location in the file system structure. That directory is going to stay in my user's home directory. Really, even though we're technically moving the directory, because we're keeping it in the same location where it originally was, we, essentially, are just renaming it.

The rmdir Command 5:35-6:15

There may be times when you need to remove a directory completely from the file system. There's two different ways you can do this. The first one is to use the rmdir command. As its name implies, rmdir is used to remove directories. The syntax is pretty simple. You type 'rmdir' followed by the directory name that you want to delete. In this case, I'm going to delete the docs directory from my user's home directory.

Just as with all the other commands we've looked at, you can either specify a relative path to the directory you want to delete, or you can specify an absolute path, as we've done here. In this case, we specified that the docs directory is in my user's home directory.

Empty Directory Requirement 6:16-7:39

There's something very important that you have to remember about rm, or else you're going to have problems: that is the fact that the rmdir command requires that the directory you want to delete must be empty before it will delete it. If there's anything in this directory at all--a file, a subdirectory, it doesn't matter--rmdir will not delete it.

An example of this is shown on this slide. Notice here that I used the rmdir command to try to delete the docs directory in my user's home directory. I got an error message. It says, "rmdir failed to remove the directory because it's not empty." What do you do in this situation? Well, you have two options.

One is to go into that directory using the cd command and then delete all the files and subdirectories that are already within it. Once you do that, then you can use rmdir. That's really not a very elegant solution, because many times, you'll find that the directory you want to remove has files in it and subdirectories within it. Within those subdirectories are more files, and you have to delete those before you can delete the subdirectory. Many times, that subdirectory will have other subdirectories in it with their own files, and you could spend an awful lot of time deleting files, removing subdirectories, deleting files, removing subdirectories, just to remove this directory, right here.

The rm Command 7:40-8:30

A better option is to use the rm command. The rm command removes files and directories from the file system. It's actually a much more powerful command than the rmdir command. I will actually confess that I've never used the rmdir command because of that limitation. If I need to delete a directory, I always just use the rm command.

To delete a directory and all of its contents, including any sub-directories, you run rm using the -r option, and then you specify the name of the directory that you want to delete. Just as with all the other commands we've looked at, you can either use a relative path, or you can use an absolute path. In this case, we've specified that the directory we want to delete is in my user's home directory. It wouldn't matter where I was at in the file system. It would delete the docs directory in my user's home directory.

The rm -r Command Option 8:31-8:59

This -r parameter, right here, is important as well. The -r option tells rm to delete files and folders recursively, which basically means, "Take a look inside the directory we specify, and if there are any sub-directories and files within that directory, go ahead and delete them, too." After

we ran that command, we ran the ls command, and we see that the docs directory, which would have been right about here in the output, is gone. It's not there. It's been removed even though it had stuff in it that rmdir could not delete.

Summary 9:00-9:20

That's how you manage directories from the shell prompt.

In this lesson, we talked about using the mkdir command to create a new directory. We talked about using the cp command to copy a directory. We talked about using the mv command to move a directory. We talked about using the rmdir command to delete an empty directory. Then we ended this lesson by talking about the rm command, which can be used to delete a directory that already has stuff within it.
