2.8.4 Manage Directories

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

Manage Directories 0:00-0:11

In this demonstration, we're going to spend some time looking at several shell commands that you can use to manage directories in the Linux file system.

Create New Directories 0:12-1:16

The first one we need to look at is the mkdir command. M-K-D-I-R, as it name implies, creates new directories. Let's suppose we'll want to create a new directory in my home directory called mynewfiles. We would type 'mkdir mynewfiles'. I hit Enter, do an 'ls' command and we see that a new directory is created called mynewfiles, and we can put content in that new directory. Again, because we did not explicitly specify a path for mynewfiles, the mkdir command assumed we wanted to create it in the current directory. If we wanted to create it somewhere else in the file system, we would have to explicitly specify the full path, starting at the root, to that particular location. For example, we could type 'mkdir /tmp/rtracey'. Notice that we've used an explicit path here because we started at the root, and we worked our way down the file system to the specific directory we wanted to create. That created a new directory in the /tmp directory. We can verify this by typing 'ls /tmp', and we see a directory here called rtracy now created.

Remove Directories 1:17-2:52

In addition to creating directories, we can also remove directories. There are two different utilities that you can use to remove directories. One is called rmdir. As its name implies, this is used to remove directories. There is something very important that you need to remember about rmdir, and that is the fact that it will only delete a directory if that directory is empty. It can't have any files in it. it can't have any subdirectories in it. In our file system, we have a directory named temp. We do an 'ls temp'. we see it has five files within it. So, if I were to type rmdirtemp, the command fails. Why? Because the directory is not empty.

At this point, you have two options. One would be to go into that directory and manually delete all the files and subdirectories and then run rmdir again, which would take a lot of time, or you can use a different utility called rm. The rm command will remove a directory with contents in it if you specify the -r option with it. You have to be very careful because it will not prompt you by default before removing all those subdirectories and files within the directory you specify. There are options you can use with rm to change that to require it to prompt you before removing anything. but if you just specify -r, it won't prompt you, so be really careful before you use it. You could delete stuff that you don't want to delete. Let's try it. 'rm -r /tmp'. Now, if we do an ls command, the temp directory is gonE that was up here before.

The next utility that we need to look at is the cp utility.

Copy Directories 2:51-5:27

As its name implies, this is used to copy directories. Within my home directory, I have a directory called mynewfiles, right here. We just created that a minute ago with the mkdir command. If we go into mynewfiles, we can see that it's empty (obviously, because we just created it). Let's use the touch command to quickly make several empty zero byte files in this folder. If we do an ls command, we see that we have five files within mynewfiles. We could also use the mkdir command, here, to create a new subdirectory within the mynewfiles directory that we just created. Let's create one called 'otherfiles', and let's do that again to create a directory called 'someotherfiles'. If we do an 'ls', we see all the files we just created, as well as these empty subdirectories.

Go up to the parent directory by typing 'cd ..'. Let's suppose we need to copy mynewfiles along with all the files within it and all the subdirectories within it to a different location in the file system. To do this, we use the cp command. But, because we want to copy all the contents of mynewfiles, we have to use the -r option, which configures the cp command to copy files and subdirectories recursively. In this situation, suppose we want to copy mynewfiles from this location to the /tmp directory. To do this, I type 'cp -R mynewfiles', and we want to copy it to /tmp. We do an 'ls /tmp' now. We see the mynewfiles folder that was created. We could run the ls command again and this time specify mynewfiles, and we see that all of the files and the subdirectories that we created earlier in the home directory have been copied over to the /tmp directory as well.

Let's go ahead and delete that folder because we don't actually want to keep it over there in /tmp. We use 'rm -r /tmp/mynewfile'. If do an 'ls' command, we should see that mynewfiles has been deleted from the /tmp directory. We do an ls of the local directory. It's still here because, remember, we copied, we didn't move, so we kept the original source directory and files and subdirectories in place and just made a copy of them over in /tmp.

You can also move a folder from one location in the file system to another.

Move Folders 5:23-6:09

When you move, the original source directory is deleted and is moved over to a different location. This is done using the mv command. As you might expect, mv is short for move. We can enter move. In this case, let's move the mynewfiles directory to the /tmp directory. If we run the ls command, notice that mynewfiles is gone from the local directory. We do an ls of the /tmp directory. We see that mynewfiles has been moved over there. Let's do an ls of mynewfiles in the /tmp directory, and we see that all the files and subdirectories that we originally had in the home directory have been moved over to the /tmp directory.

That's it for this demonstration.

Summary 6:09-6:11

In this demo, we looked at several different commands that you can use to manage directories in the Linux file system.
