2.8.1 Directory Navigation

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

Directory Navigation 0:00-0:14

In this lesson, we're going to review how you navigate within the file system. There are two different shell commands that you use to do this. The first is pwd, and the other one is cd. Let's begin by looking at how you use pwd.

The pwd Command 0:15-1:20

The pwd command is a relatively simple utility, but it also can be kind of useful at the same time. The name of the command, pwd, stands for present working directory, although some people call it print working directory. Either way, it does the same thing. The command is used to display the current directory you're working in on the screen. This can be useful if your shell profile hasn't been configured to automatically display the current directory as part of the shell prompt.

The syntax for using pwd is very simple. As you can see here, all you have to do is type 'pwd' on the screen. When you do, it displays your current directory on the screen. As you can see here, I entered the pwd command, and it tells me that my current directory is /home/rtracy. My prompt is configured on this system to display three different pieces of information. First, it tells me my username, who I'm currently logged in to the system as. It also tells me the hostname of my system. And then, finally, over here, it tells me what my current directory is.

The Tilde (~) Character 1:21-1:59

Notice, here, that pwd says I'm in /home/rtracy, but my prompt says that I am in tilde (~). Remember that the tilde character is a shortcut. What tilde references is the /home directory of whatever user is currently logged in. It's just kind of a shortcut abbreviation for indicating the home directory. In this case, I'm logged in as rtracy, so tilde points to /home/rtracy. If I were logged in as a different user--say I were logged in as the ksanders user--then I would still see tilde in the prompt, but it would point to an entirely different directory. It would point to /home/ksanders because that's that user's home directory.

The cd Command 2:00-2:19

Once you know where you're at in the file system, you also need to know how to change to a different directory in the file system. You do this using the cd command at the shell prompt. To use this command, you simply enter 'cd' followed by the name of the directory that you want to switch to. Understand that there are actually two different ways that you can do this.

The cd Command with a Relative Path 2:20-3:27

First of all, you can enter cd and then the name of a directory without specifying the full path to that directory. When you do this, cd will assume that the directory that you're specifying is a subdirectory of whatever the current directory is. That's shown here in this example.

Here. we've entered the cd temp command. Because we did not specify a full path, the cd command is going to switch us to a sub-directory of the current directory, whatever that may be, named temp. You can see here that we use the pwd command to verify that we truly did switch to the right directory. We switched to /home/rtracy/temp.

This is called using a relative path. We call it a relative path because the path specified within the command is relative to some other point in the file system, which, in this case, is the current directory. How do you know that this is a relative path? Notice that we did not specify, right here, the full path to that directory. There's no context clues for the cd command to tell us where temp is, so it has to assume that temp is in the current directory.

The cd Command with an Absolute Path 3:28-4:21

You can also use absolute paths with the cd command. When you use an absolute path, you specify the full path to the directory that you want to change to, starting with the root directory, which is forward slash (/). You can see an example of that right here. In this example, we enter

the cd command, and then we start at the top of the file system, '/', and then we provide the full path to the directory that we actually want to change to.

We're telling cd to switch us to the /home/rtracy/temp directory. This command would work no matter what the current directory is because the cd command really wouldn't care what the current directory is, because we've specified the full path to the directory that we want to change to. The path that we want to change to is calculated from the /root directory, and here you can see that the current directory was changed to that path.

Move to a Higher Directory 4:22-5:55

You can also use the cd command to move up a level in the file system hierarchy. You do this by entering 'cd ..'. This will change the current directory to the next level higher in the hierarchy. You can see that here. We were in the /temp directory. We type 'cd ..', and we go up one level to my user's /home directory. We were in /home/rtracy/temp. We typed 'cd ..'. Now we're in /home/rtracy.

I do need to point out that you need to make sure you insert a space right here, between the cd command and the '..'. If you forget that space and you type 'cd..' with no space in between, this bash shell is going to think that you want to run a command named cd.. It will look through the file system and say, "I don't have a command named cd.." Always remember to put that space in there. Do be aware that some distributions will actually create an alias named cd.. with no space, just to address this issue. If you do accidentally forget to type that space in there, it will run the alias, which will then run the 'cd ..' command for you automatically.

In the second example, here, we've moved up two levels in the file system hierarchy. Notice that we typed 'cd ../..'. That says go up one level, and then go up one more level. Because we were in the /home/rtracy directory after we typed the second command, we're taken up two levels to the root of the file system, just forward slash (/).

The ls Command 5:56-6:41

To this point, we've discussed how to view the current directory and how to change to other directories in the file system. Now we need to shift gears.

We need to talk about how to list the files and subdirectories that reside within a given directory. This is done using the ls command. Let's take a look at how it works.

If you just enter 'ls' at the shell prompt, as in this first example, then the contents of the current directory are listed on the screen. In this situation, if you look at my prompt, you can see that my current directory is the /root directory of the file system, forward slash (/). When I type the ls command, all of the files and all of the subdirectories located in the /root directory of the file system are displayed. These are listed right here.

Relative vs Absolute Path 6:42-7:24

This is an example of a relative path. Because we just typed the ls command with no other directory listed as an argument for the ls command, it assumed that we wanted to look at the current directory.

In the second example, though, we provide an absolute path of the directory we want to look at with the ls command. In this case, we view the contents of the /home/rtracy directory, which are then displayed on the screen. Because we specified the full path to the directory that we want to look at, the ls command doesn't care what the current directory is. We told it to start at the root, go down to /home, and then go to /rtracy and display whatever is in there.

---

ls Command Options 7:25-8:27

The ls command includes several useful options that you should be familiar with. You can see what they are by viewing the man page for ls. However, there are two critical ones that you have to know. The first one is the -a option, which causes ls to display all files.

Why is this useful? Well, because by default the ls command, as run in these two examples, will not display any of the hidden files within the directory specified. If you add the dash -a option, then it will.

Also, -l is extremely useful. When you run the ls -l command, it will display a long listing of the directory contents. Notice here we did not use the -l option, so the only thing we saw was the name of the file or directory within the specified directory. Using dash -l, we could see much more information, such as the permissions that are assigned to that file directory. We could see which user and group owns the file or directory. We could see how big the file is, and so on.

Summary 8:28-8:36

In this lesson, we talked about how to use the pwd command to view the current directory. We talked about using the cd command to change the current directory, and then we ended this lesson by talking about the ls command.

Copyright © 2022 TestOut Corporation All rights reserved.