2.12.2 Finding Linux Commands

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

Find Linux Commands 0:00-0:17

In this demonstration, we're going to look at several different utilities that you can use at the Linux shell prompt to search for Linux commands. First, we'll look at the which command. Then we'll look at the whereis command. And then we'll end this demonstration by looking at the type command.

---

which Command 0:18-3:17

Let's begin with which. which is a handy little utility. It can be used to locate the executable file for a particular Linux command. In order to do this, it searches through all of the directories in your PATH environment variable to locate that particular command. To view all of the directories in our PATH environment variable, I can type 'echo $PATH' in the shell prompt. And when I do, I see a list of directories, each separated by a colon, that are contained in the PATH environment variable.

When I run the which command, it's going to look in each one of these directories and try to find the executable file for the command that I specify. If it finds a match, it'll display the appropriate directory here, on the screen. For example, we could type 'which ln', which will tell which to look through all of these different directories and try to find the executable file for the ln command. Press Enter, and it locates that executable in the /user/bin directory. When we run the ln command, we know now that we're actually running the /user/bin/ln executable file. We can do the same thing with other commands. We can search for 'which vi' to find the executable file for our vi editor. Press Enter, and it's located in the same directory structure, in /user/bin.

It's important to note that the which command only looks in the directories specified here, in the PATH environment variable. It doesn't look anywhere else in the file system. So, if the file we're searching for does not reside in one of these directories, which will not be able to find it. For example, if I do an 'ls' here, we can see that within my /home directory, I have an executable file here called "helloworld". My /home directory, /home/rtracy, is not in the PATH environment variable. There is a subdirectory of my /home directory that is--this /bin directory, right here--but my /home directory itself, /home/rtracy, is not in my PATH. Therefore, the which command will not be able to locate this particular executable file. If I were to type 'which helloworld', it says, "Hey, I can't find it. I can't find an executable named helloworld in any of these directories."

Notice that there is a subfolder of my /home directory here, as we noted earlier, called bin, that is in the path. If I were to use the cp command to copy 'helloworld' to the ./bin directory--cd into bin now, and do an 'ls' make sure it's there. Okay. So, now helloworld is in the /bin directory, and the /bin subdirectory of my /home directory is in the PATH. which should be able to find it now if we do 'cd ..' to go up a level in the file system. Let's run that which command again. Now it's able to find it because it's in one of the directories in the PATH environment variable. Let's clear the screen.

---

whereis Command 3:18-3:31

Another utility that you can use to search for information about a Linux command is called the whereis utility. The whereis utility performs a similar function to which in that it

---

type Command 3:32-8:16

locates the files associated with a particular command, but it displays a lot more information than which does. First of all, it will display the location of the command's executable, just like which, but it will also display the man page location for that command. If you have your sources installed on your system, then it will also tell you where that source code for that command resides, as well.

For example, we could run whereis ln to find out information about the ln command, just like we did with the which command. Notice, here, we see a lot more information. First of all, we have the directory where the ln command resides. We also see that there are two man pages associated with the ln command, and here is where they are located. My source code information is not displayed because I don't have my sources installed on this system. I wanted a nice, clean, trim Linux distribution without a lot of extra things that I'm not going to use. If you have your sources installed, it would then also tell you where the source code for the ln command resides as well.

The last utility we are going to look at is the type utility. The type utility analyzes a particular command--whatever command you tell it to look at--and will tell you what kind of command it is because there actually many different types of Linux commands. First of all, there are commands that are actually built into the shell itself. These commands don't have an executable file in the file system. They're actually just

built into the shell's code itself. In addition to a built-in command, it could also be an external command. ln is an example of an external command because it does have an executable somewhere else in the file system. The command that you run at the shell prompt could also be an alias, in which case, the alias itself doesn't have a file in the file system; it just points to some other command somewhere in the file system. Or it could be a shell function that's been defined. Let's clear the screen here.

Let's use the type command to view information about the cd command for change directory. We use the change directory command all the time. Let's press Enter and see what kind of command it is. As you can see here, cd is actually built into the shell itself. There is no external executable for the cd command. Let's run the type command again. And this time, let's look for information about the ln command. Now the output is different. ln is not built into the shell itself, but instead is an external command with a real executable that resides in the file system. In this case it's, /user/bin and the name of the executable is ln.

For external commands, there is a possibility that you will see different output from the type command that you need to be aware of. Here, when we looked for ln, type reported back the directory where the ln command resides. Let's do something different here. Let's do 'type' again, and let's search for information about the vi command. Again, it tells us where the vi command is located in the file system--/user/bin. Watch what happens if I actually load vi, and let's go ahead just exit out, not doing anything special. Just load the command, then exit. Now let's type the vi command again. You'll see that the output is different this time. It tells us that vi is hashed.

You might be asking, "What on earth does that mean?" It does still supply the PATH to the vi command over here, but it tells us over here that it's hashed. Well, when you see that, it means that the command has been recently run, and its location is now stored in the shell's hash table. That means that type already knows where the command is, so it doesn't have to search through all of the directories in your PATH environment variable to try and find it. You'll notice the first time we ran the type vi command, we did not receive the message that it was hashed. I ran it, ran the type vi command again, and now, because we've run it, the shell knows where vi is, and it's stored in the hash table. If you see hashed in the output of the type command, you know that is an external command that has been recently run.

Let's take a look at one other example. Let's type the 'clear' command here real quick. I want to show you the alias command to look at list of aliases that are defined on the system. For example, we can see ll, and lll, and so on. These are all aliases that are defined when the system starts up. Well, if you use the type command and you specify a command that is actually an alias to another command, then the type command will tell you that it's actually an alias. For example, let's do 'type lll'. Hit Enter. It tells me here that lll is not a real command; instead, it's an alias that points to the ls -al command. We can verify that in the output of the alias command.

Summary 8:17-8:27

That's it for this demonstration. In this demo, we took a look at several shell utilities that you can use to get information about Linux commands. We first looked at the which command. We then looked at the whereis command. And then we ended this demonstration by looking at the type command.
