2.12.3 Finding Files

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

Finding Files 0:00-0:15

In this demonstration, we're going to spend some time looking at several Linux shell utilities that you can use to locate files within the Linux file system. First we're going to look at the find utility, and then we're going to look at the locate utility.

find Utility 0:16-6:20

Let's begin with find. The find utility is used to search the Linux file system to locate a particular file. For example, let's take a look at the contents of my home directory. I'm currently in the /home/rtracy directory. Here, you can see a listing of all the different files within this directory. Let's suppose we need to search for the logfile file, right here. We've created this log file at some point, and now we can't remember where it's located. We can't remember where we saved it, so we're going to try to locate it using the find utility.

The syntax for using find is to type 'find', and then you have to specify where you want to start searching from. By default, the find utility will begin at the directory you specify and then search through all of the subdirectories beneath that search location that you specify. For example, if I wanted to search the entire file system, I could specify a / and then specify the name of the file that I want to search for. This search will take a long time to complete because it's going to have to go through lots and lots and lots of directories. However, in this case, let's suppose we know that we created it in some home directory on the system; we just can't remember which one we put it in.

Instead of searching the entire file system, let's constrain our search to /home. Using this syntax, the find command will start its search in /home and will look in all the subdirectories of /home as well. This will greatly decrease the amount of time it takes for this command to run. Now we need to specify what we want to search for. find can search for a lot of different file attributes, one of which is filename. It can search for other file attributes as well. If we want to search by name, we use the -name option. Then we specify the name of the file that we want to look for, such as logfile. This will cause find to begin searching in the /home directory and then search through all the subdirectories of /home trying to find a file named logfile.

I need to point out one thing here for you. As a best practice, it's not a bad idea to put the name of the file that you're searching for in quotes. Here's the issue: if the filename you're looking for does not contain spaces, you can actually get away without using the quotes. But if the filename you're looking for does have spaces in it, then you do have to enclose the entire filename in quotation marks. Otherwise, the find utility is going to be confused as to what it is you're actually asking it to do. As a best practice, use quotation marks around the filename. Press Enter, and it goes through and searches through /home and all the subdirectories of /home, and it locates the file in question. It's in /home/rtracy/logfile.

In addition to searching by filename, the find command can also be used to search for files using regular expressions, meaning that we're going to create a regular expression, and the find utility will try to find all the files that match that regular expression for us, instead of searching for just one particular filename. Let's go ahead and switch to my root user account using the su- command. Oops. I typed it wrong. We'll try it again. I have to switch to root because I want to search for files located in the /var directory. The rtracy user account has limited access to the /var directory, so I'll switch to the superuser root account. Now I can search for whatever I want within this directory.

Within the subdirectories of /var, there are a variety of different log files. I want to search for all of the files that end with a .log extension, meaning I'm looking for a lot of different files that all have the same extension. I'm going to go ahead and switch back to my root user's home directory, and we're going to execute the find command from there. We're going to specify, first of all, where we want the search to start. Let's start in /var. Then we specify -name to specify that find search for a particular filename. Then we specify a regular expression instead of the filename itself. In this case, we'll do a very simple one. We'll use a '*.log'. The star that we used right here matches anything, meaning that any filename that ends in .log will be matched, and find will display the results on the screen.

Let's go ahead and run it and see what happens. You can see that several files in the /var directory, in the various subdirectories of /var, match because they all end in .log. The results were printed on the screen. In addition to finding files based on their filename, the find command can also be used to find files that are owned by a particular user. To do this, we enter 'find' again, and we, once again, specify where we want to start searching. Let's look in the /home directory, which will search /home and all of its subdirectories, and then specify, instead of -name, -user to specify that now, we're looking for files that are owned by a particular user. We're not looking for a particular filename. We're only concerned about the file owner. We'll enter the name of the user that we want to find. We'll find all the files that are owned by the rtracy user in /home and all of its subdirectories.

Press Enter. And here, you can see all of the files that are owned by the rtracy user. Go ahead and clear the screen. In addition to find, you can also use another utility called locate to find files in the Linux file system. Here's the issue: the find utility works great, but in order to find the files that you're looking for, the find utility actually has to explicitly search through the file system structure starting at the directory you specify in order to find matching files.

locate Utility 6:21-9:19

The locate utility, on the other hand, works quite differently. Instead of explicitly searching through the file system each and every time you run the command, locate, instead, builds its own initial database that contains an entry for every single file in the Linux file system.

Then, when you run a search using locate, it will actually search within the database itself, instead of the actual file system, which runs much faster. Be aware that before you can use the locate command, you have to have it installed on your system. Some distributions will install locate by default; others will not. You have to have the findutils/locate package installed in order for you to be able to run the locate command. Once you've installed it, you do have to build that initial database. By default, the locate database will be empty. By default, the database used by the locate command is updated regularly with all the changes that are made to the file system. However, there may be situations when that database does not contain the information that you need to search for.

For example, if you just barely installed the locate utility, the database is going to be empty. You need to run it once in order to create an entry for every file in the file system in that database. In addition, if you just made several different files, and those changes have not been picked up yet and added to the database, then they won't be listed, and you won't be able to search for them. In these situations, you may need to manually update the database. In order to do this, you can manually update the database using the updatedb command. The updatedb command will look through the file system for any files that have been added or removed since the last time it was run and make the necessary changes in the database.

If this is the first time you've run this command, it's going to take some time to complete because it has to go through every single directory in the file system and catalog every single file. If, on the other hand, it's been run before, all it has to do is go and pick up the changes, and it will take quite a bit less time. I'm going to run it right here. It ran fairly quickly. I only had to make a few changes. With my locate database updated, I can then use the locate command to find files in the file system.

For example, let's try to find files that contain 'vim' in the filename. These are files that are associated with our vi editor. Press Enter. You can see that there are many files that have 'vim' in the filename somewhere. You can also see that it ran a lot faster than the find command did. Because it only had to query its database instead of having to search through every single directory in the file system, it was able to provide us with almost instantaneous output. Let's try running it one more time. Let's do 'locate', and let's look for a very specific file this time. Let's locate the sshd_config file. Press Enter. You can see here that here's the sshd_config file, and here's the man page associated with the sshd_config file.

Summary 9:20-9:27

That's it for this demonstration. In this demo, we've talked about how to find files in the Linux file system. We first looked at using the find command to find files. Then we looked at using the locate command to find files.