2.12.1 File Location Utilities

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

File Location Utilities 0:00-0:28

In this lesson, we're going to review how to find files in the Linux file system. If you're like me, this is a useful skill to have, because I create files all the time and then can't remember where I saved them. Fortunately, Linux includes a variety of different utilities that you can use to search for a specific file or directory in the file system, so in this lesson, we're going to review the find command, the locate command, the whereis command, and the which command.

The find Command 0:29-3:03

Let's begin by looking at the find command. The find utility is a very useful tool that you can use to search for files and folders within the Linux file system. The syntax is shown here. First, we enter find at the shell prompt, and then we have to specify where we want the find command to start looking for the file in question. In this example, I specified /home. Now here's the important thing you need to remember, by default the find command will look for the file in question in /home if we specify that here, but it will also look in all the subdirectories of /home and all the subdirectories of those subdirectories. It will search recursively through the directory tree starting at the directory we specify trying to find the file name in question, and then we use the -name parameter to specify what file we're actually looking for.

In this case, we're looking for a file named files.txt. When I run the command, find will start in home and it'll look for any file that matches the search string that we specify, and as you can see here the output will report where it found that file at. In this case, it found the file in /home/rtracy and the file name as we specified is files.txt. Be aware that you can use regular expressions with find to make your searching a little more flexible. For example, let's suppose you need to find all of the log files that have been stored on your system and we're going to assume that all the log files have an extension of .log.

To do this, we would enter find, and then we'd specify where we want to start searching. In this case, we want to find any file anywhere in the file system that ends in .log, so we're going to specify the start search location as just / that's the top of the file system tree, and remember that by default, find searches recursively so it'll start looking in the root directory and then it'll start looking in every single subdirectory of the root directory, and then it'll start looking in every subdirectory of every subdirectory of every subdirectory for the file name that we specify. Then we use the -name parameter to specify what file names we're looking for and then we specify *.log. Now star is an example of the regular expression. That means match anything so it doesn't matter what the file name is as long as it ends in .log, and then you can see here the find command started reporting what it found. It found several different files that end in .log, and it outputs their location and file name on the screen.

The find Options 3:03-3:37

Be aware that the find utility includes many other options that you can use to make your searching more flexible, and I suggest you go to the find utilities man page to find out what all of them are. Two of the ones that I find very useful are listed here. First of all, you can use the -user option to search for files that are owned by a specific user.

You can also use the -size option to search for files of a specified size. You can specify a fixed size, you can specify files that are all smaller than a certain size or larger than a certain size and so on.

The locate Command 3:37-4:46

In addition to find, you can also use the locate utility to search for files within the Linux file system. The locate utility functions in much the same manner as find; however, it has one distinct advantage over the find command. Here's the key thing that you need to remember. Whenever you run a search with the find command it has to manually walk through every single directory in the path that you specify in the command looking for files that match the pattern that you specified, and this process can take a really long time especially if you're doing a top down search say from the root of the file system.

Now the locate utility on the other hand works a much more efficient way. Instead of manually walking the file system every time you run a search, the locate utility actually first builds an index of all the files in the file system. Then when you actually do run a search instead of manually walking through all the different directories the locate utility simply looks at its index and says, "Oh, that file is located here." It doesn't actually look in the file system directory. As a result the locate utility usually runs a lot faster than the find utility in most situations.

Install locate 4:47-5:38

Before you can actually use the locate you do have to install it on the system first. Most distributions don't include the locate utility by default. What you need to do is make sure that the find utils package is installed on your system and most of the distributions that I've worked with do not include the locate utility by default. You need to install it manually using RPM, dpkg, Zipper or Yum.

Then with that package installed you need to create an index of all the files in the file system, and the name of that index is called locatedb, and it's stored in the /bar/log directory. Now this index as I said contains a listing of every single file that locate's been able to find in the file system, and as you know, that file system is going to change periodically therefore that index has to be updated, and by default, it's updated every day with the latest changes to the file system.

Update the Database 5:39-9:01

Now as you might guess updating that database every day may not be soon enough. Especially if it's a heavily used system where files are being created and deleted all the time. In this case, you can manually update locate's index using the updatedb command in the shell prompt. Be aware that updatedb does all the legwork for the locate utility, and as such, it can actually take a lot of time to complete, and it'll also use up a lot of system resources during the update process.

Once the index has been initially populated and then updated, you can then use the locate command to search for files in the file system. The syntax for searching for files with locate is shown on the slide. First of all, we run the locate command, and then we specify the file name that we want to search for. In this example, I'm searching for the sshd_config file. That's really all there is to it at which point the locate command will look through its index and try to find any file or directory that has this text string somewhere in its name.

As you can see two files were found that match. We have the /etc/ssh/sshd_config file, and it also found the man page for that file because its file name contains the search term we were looking for sshd_config.

You can use the /etc/updatedb.com file to configure the behavior of the updatedb utility. For example you can use the PRUNEFS directory within this file to specify specific file system types that you don't want updatedb to scan. By excluding them from the scanning, any files located on these file system types will not be included within the index.

Notice right here that we have for example cifs file system. cifs file systems refer to Windows shares that have been mounted locally in the file system using the Samba service. Essentially, we're mapping a drive if you will to a share on a Windows server. Well, we don't want the locate utility going out and indexing the files on that share, and because the file system used to mount that share is the cifs file system we're going to exclude that from the list of file systems that the updatedb command will include in the index.

Also notice, we have iso9660 listed in the list. That will prevent updatedb from indexing files that are on removable optical disc drives, because obviously if we're going to be removing that disc from the drive and putting new ones in or moving and putting them in, the file systems are going to change frequently; therefore, we don't want to include them in the index. We can also use the PRUNEPATHS directive down here to create a list of directories in the file system that you want updatedb to skip when it's scanning.

Notice that we have the /media and /mnt directories for example included in this list. That's because we mount removable external devices into these two directories. So those file systems are going to be changing all day as we plug in USB drives, as we unplug USB drives, as we insert optical discs, as we remove optical discs, and therefore, it doesn't make a lot of sense to keep a listing of the files on those devices in the index, so we exclude them.

Same down here with /var/spool/cups. This is where our print jobs are queued up waiting to go to go the printer and those are going to be changing all the time, too, and again we don't need to keep the file names for those print jobs stored in the index.

The which Command 9:02-9:33

With this in mind, let's now switch topics and talk about using the which command. Now the which command is used to display the full path to a particular shell command or utility. For example, let's suppose we want to know where the ls command actually resides in the file system. We run ls all the time, but we don't actually know where the executable for ls is. To do this we just type "which ls" at the shell prompt and it returns the full path to the ls command which as you can see here is /bin/ls.

The whereis Command 9:33-10:14

In addition to the which command, you can also use the whereis command to find files. Now the whereis command is a little bit different. The whereis command is used to locate several key pieces of information about a particular file. For example, it will locate the source code for the file if it's installed on the system. It'll locate the binary executable file itself, and it'll also locate the manual page for that particular

file. In this example, I've run the whereis ls command at the shell prompt and it provide several pieces of information. It tells me where the executable is located, and then it tells me where the man pages for that command are located as well.

Summary 10:15-10:27

So that's how you find files in the Linux file system. To locate just any file you can use either the find command or you can use the locate command. If you need to get information about a particular shell command or utility, then you can use either the which command or the whereis command.
