2.12.5 Content Search Utilities

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

Content Search Utilities 0:00-0:20

In this lesson, we're going to look at content search utilities that you can use on a Linux system. You can use the find utility and you can use the locate utility to find a specific file within the Linux file system. But what do you do if you need to find specific content within a file or files in the Linux file system?

The grep Command 0:21-1:42

Linux provides a very useful utility called grep that you can use to look for specific content within a file. grep is one of those tools that most Linux administrators can't live without. In fact, I dare say that most administrators use it on a daily basis. That's because using grep, you can search through a file to locate a particular text string. This can come in really handy when you need to search through a very large log file for just a specific message. Or maybe you need to find a specific directive within a configuration file. In fact, you can even use grep to search through multiple files all at once to try to find a particular text string.

To use grep, you run the grep command at the shell prompt, and then you specify the search string or the expression that you want to search for. In this case, I'm running grep, and I'm looking for the text string firewalld. And then you specify which file you want to find that expression within. In this example, I'm looking for the text firewalld within a file named boot.log located in my /var/log directory. Basically, this will display any log entries within this log file related to the firewalld daemon on my system. As you can, see grep returns each line within that file that contains an instance of that search term.

The -i Option 1:43-2:17

When you're working with grep at the command line, there's a couple of very useful options that you can use. The first one is the -I option, which causes grep to ignore case when searching for the text string. Be aware that, by default, grep does consider case when it's performing its searches. If you use, say, a capital letter in your search string, then any matches within the file specified have to use that exact same case. That might be appropriate in some situations. But many times, you're not exactly sure what the correct case is for your search term, in which case you can use -I so that grep ignores case.

The -l and -n Options 2:18-2:50

Another useful option is the -l option. Previously, we saw that when grep finds a matching line in the file that it's searching, it displays the contents of the entire line on the screen. Well, there may be times when you don't actually want this to happen. Instead, the only thing you care about is which file contains the matching text. If you use the ‘-l' option, then whenever it finds a match within a file, grep will output the name of the file itself, but not the matching text.

You can also use the -n option to display matching line numbers. I don't use that option very often.

Prepare and Maintain locate 2:51-3:23

One that I do use all the time is the -r option. This causes grep to search through multiple files. Now, r stands for recursive. Basically, using the -r option causes grep to search recursively through all of the subdirectories of the path that you specify. This is really useful in situations where you're looking for a particular term, but you're not 100 percent sure which file it's actually in. Using this option allows grep to go through all of the files within the path you specify and try to find any matching instances of your search term.

The egrep Command 3:24-3:59

The grep utility works great when you need to search for a specific text string within a file. However, be aware that there may be times when you need to search for more complex patterns.

In this situation, grep really isn't the best option. Instead, you need to use the egrep command. Now, I just said that grep isn't the best option, but it's not entirely true Because the egrep command is actually the same thing as running grep with the -E option as well. The key advantage of using egrep or grep -E is the fact that instead of searching for a specific text string, it can instead search for files using regular expressions.

Regular Expressions 4:00-4:33

Regular expressions are strings consisting of metacharacters and literals. When I say literals, I'm talking about regular letters A through Z as well as numbers.

Understand that metacharacters are not literals. Metacharacters do not represent themselves; they instead represent other characters. This can be really useful because it can be used to specify a character's location or matching text within a text string. Using egrep, you can use regular expressions to create some very complex search patterns.

The Star (*) and Period (.) Metacharacters 4:34-5:43

There are many different metacharacters you can use in a regular expression. A few of the more commonly used ones are shown here.

First of all, we have the star (*) metacharacter. The star tells the grep utility to match on any number of characters. You might be asking, what does that mean? Let's look at the example over here.

We enter a search term of ‘Stuff*'. Then grep will match any word that begins with stuff, such as Stuffing, Stuff1, Stuffy, and so on. Basically, the star character is a wild card. It says, "Match anything."

On the other hand, you can use a period. A period matches just one single character. Where a star matches any number of characters, the period only matches one single character. For example, if we search for ‘Stuff.' with grep, it'll match text such as Stuff1 and Stuffy, but it will not match Stuffing. The reason why is because Stuff1 and Stuffy both match one single character where the period is located. Stuffing consists of three characters where the period is, and so no match will be made.

The caret (^) and dollar ($) Metacharacters 5:44-6:49

You can also use the caret symbol (^). This matches an expression if it appears at the beginning of the line. In this example, we use ‘^Stuff'. If we search for caret stuff, it will match any instance of the word stuff as long as it comes at the beginning of the line.

By way of contrast, you can also use a dollar sign character ($). This will match an expression only if it appears at the end of the line. For example, if we search for ‘Stuff$', it will match any instance of stuff as long as it appears at the end of the line.

These are just a few of the metacharacters that you can use in regular expressions. There's actually many, many more, and we can't cover them all here. I encourage you to open up the Linux system, go to the shell prompt, and type ‘man regex,' and you'll see the man page for regular expressions. There's a lot more information available there about how you can use regular expressions. It also provides you with a lot of examples. In addition, if you want to go on the internet and search for regular expressions, you'll see tons and tons of different ways you can construct very powerful and very flexible search terms using regular expressions.

Escaped Characters in Regular Expressions 6:50-8:07

Before we go on, we need to point out that when you're using egrep, you need to be familiar with the concept of escaped characters. This is kind of confusing. To escape a character, we put a backslash (\) in front of it. This is very important because that backslash character tells egrep that the character that follows is not a regular expression metacharacter, but is just plain old text.

For example, let's suppose we need to search through a log file, and we need to find any instance of the text string www.spam.com. We've got a problem because the search term contains periods. Remember, as far as egrep is concerned, those periods are metacharacters which means match one character in this spot. That's not what we're trying to do. Instead, we want egrep to see these periods as just plain text within a URL.

To do this we have to escape those periods. This tells egrep that these are not metacharacters; they're just regular old text periods. To do this, we would put a backslash in front of each one. That way, egrep will search for www.spam.com and not www something spam something com.

The fgrep Command 8:08-8:55

In addition to egrep, there's another variation on the grep command called fgrep, and that stands for fixed string grep. The fgrep utility is used to search for content within files, just like egrep and just like grep. But the fgrep utility searches files for lines that match a very specific fixed string. Unlike egrep, it does not perform searches for regular expressions. Instead, it uses direct string comparisons to find matching lines of text in the file being search.

The syntax used is, basically, the same as that with grep and with egrep. You type ‘fgrep' at the shell prompt followed by the pattern that you want to search for and then the file name you want to search within. I might point out that just like egrep is the same as running grep with a -E option, running fgrep is basically the same as running grep with the -F option.

Summary 8:56-9:06

That's it for this lesson. In this lesson, we talked about searching for content within Linux files. We first looked at the grep command. We then looked at using the egrep command with regular expressions. And then we ended this lesson by looking at the fgrep utility.
