3.3.1 Locale Settings

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

Locale Setting 0:00-0:34

When you install a new Linux system, you typically configure the system's locale during the installation process. This is really important because the system's locale determines several key settings, such as the language and the encoding of text that's displayed on the screen. It also determines the default number format. It determines the default currency format, and it also determines how the date and time are displayed. After you initially install the Linux system, your locale settings are saved in values assigned to several different environment variables.

Locale Environment Variables 0:35-4:53

Let's review what they are.

The first one is LC_CTYPE, which configures the default character type and encoding used in the system. There's also a variable called LC_MESSAGES, which configures your natural language messages. There's LC_COLLATE, which configures your sorting rules. There's LC_NUMERIC, which configures your number format. You have LC_MONETARY, which configures your currency format. There's LC_TIME, which configures your date and time display. There's LC_PAPER, which configures your default paper size, because different paper sizes are used by default in different parts of the world. LC_NAME configures the default personal name format because in some areas of the world, your last name comes first, and your first name comes last. LC_ADDRESS configures the default address format.

LC_TELEPHONE configures the default telephone number format. LC_MEASUREMENT configures the default measurement unit, whether you're using the metric system or whether you're using the English measurement system.

The last three environment variables that you see on this screen are very important. There's another special environment variable here called LC_ALL. It's special because anything you assign to LC_ALL actually overrides all of these other LC environment variables we just looked at. Likewise, LANG is used to specify the default locale value for all of your LC variables. There's also an older environment variable called LANGUAGE, which overrides LC_MESSAGES. If necessary, you can customize these variables, altering their values, in order to change your system's locale.

For example, let's suppose you have a computer, and it has Linux installed on it. During the installation process it was set up to use US English.

Since being installed, that computer has been moved to French-speaking Canada. In this situation, we need to change the system's locale. We need to modify our locale environment variables to reconfigure the system to use French, for example, instead of English.

We also need to change our other locale settings to make sure that we're using the metric system now, instead of the default English measurement units.

In this situation, we need to, first of all, reconfigure the character type and encoding used by the system.

This is done by modifying the LC_CTYPE variable. The LC_CTYPE variable can be assigned a locale value using the syntax shown here.

First, we specify the language.

This specifies an ISO 639 language code to be used and is specified in all lowercase.

We specify an underscore, and then we specify the territory.

The territory specifies the ISO 3166 country code to be used and is specified in all uppercase.

Then we insert a period. After the period, we identify the code set that we want to use. This specifies which character set, essentially, that we're going to use on the system. Then we have an at sign (@), which is optional. And after the @, we have a modifier, which isn't always used. It's used to specify other locale attributes, such as a dialect, or maybe a special currency.

Here are some example values you could use for the language and territory part of the LC_CTYPE variable.

First of all, en_US specifies the language and territory for United States English.

On the other hand, en_CA specifies the language and territory for English in Canada.

Remember, with the computer we're talking about this scenario, it's in French-speaking Canada. Therefore, we would use a different language and territory code. Instead of en_CA, we would use fr_CA to specify the language and territory for French Canadian.

We also need to specify in the LC_CTYPE variable the default encoding, or character set, that we're going to use.

For example, we could use UTF-8 character encoding. This is also known as Unicode.

For our French-Canadian computer, we would specify a language and territory of fr_CA.UTF-8 for the code set.

View Locale Environment Values with Local Command 4:54-10:09

When you initially installed your Linux system, your locale settings were automatically defined for you based on the selections that you used during the installation process. If you want to view them at the shell prompt, one way to do so is to use the echo command. We just type 'echo' followed by a dollar sign ($) and the name of the environment variable that you want to look at, such as LC_CTYPE. You can see this system was set up to use US English and Unicode UTF-8 encoding.

If, for some reason, you need to localize a Linux system to a locale other than the default that was configured during the installation process, all you have to do is change the value of the appropriate environment variables. Usually, what you will see is that all of these environment variables will be set to the same value.

On some systems, it may not be that way. You may see that most of the environment variables are blank, and all that's populated are the LC_ALL or the LANG variables. It's done this way in order to specify the default locale that will be automatically used by all of the other variables. We'll talk about that a little bit more in just a minute.

You don't have to use the same values for all of your environment variables. There may be situations where you can set them independently of each other to use different values. If you do this, however, be aware that not all of the LC environment variables have the same level of precedence. Linux uses the rules that you see here to determine which locale settings to use. This is done in case there are conflicting values in locale variables.

First of all, if the LC_ALL variable is defined, then that value is used, and any values that are assigned to other LC environment variables will not be checked. Remember, I said just a minute ago that on some Linux systems, you'll see that all of the environment variables are blank except for LC_ALL, or maybe LANG, in which case the value assigned to LC_ALL will be used by default by all of the other local environment variables.

If LC_ALL is undefined, meaning it's blank or has a null value, then the specific LC variables will be checked, such as LC_CTYPE. If a particular LC environment variable has a value assigned to it in this situation, that's the value that will be used.

If LC_ALL is blank and the LC environment variable in question is also blank, then we're going to use the LANG environment variable. Whatever value is assigned to LANG will be the value that we used for all of the environment variables that are blank on the system.

If you want all of your LC environment variables to use the same value, which is very common, then, really, all you have to do is set the LC_ALL environment variable. This eliminates the need to go through and set each individual LC environment variable to the same value. However, there may be situations when you do truly need to use different values for different LC environment variables.

Remember, LC_ALL overrides all the other LC environment variables. Therefore, if you want to use different values for different LC environment variables, you first have to set LC_ALL to a null value. In fact, many distributions leave LC_ALL undefined for this very reason. Instead, what they do is set the value of LC_CTYPE in order to define the default encoding that will be used on the system. And then they set the value of the LANG environment variable to define a default value for all the other LC environment variables. However, because LANG is third in priority, you can actually manually set the value of an LC environment variable without impacting any of the other LC environment variables if the variables are configured in this manner.

As I said earlier, if you need to view the value of an LC environment variable, you can use the echo command if you want. It's really not the most elegant way to do it, though, because there are so many locale variables. A better option is to use the locale command. You can use the locale command to view all of your current locale settings all at once. You can see the output of the command lists all of our LC environment variables and what their values are currently set to. In this example, you can see that almost all of the locale environment variables have been assigned a value except for LC_ALL. Remember, if we assigned a value to LC_ALL, it would completely override all the rest of these environment variables.

If we wanted to, we don't actually have to populate all of these environment variables. Instead, all we have to do is set our character encoding using LC_CTYPE and then set our language using LANG. The value we specify for LANG will be automatically used for all of the other environment variables. That way, you only have to manage two of them, instead of a whole list. You can also use the commands shown on this screen to view the type of encoding that's being used on your system. You just type 'locale charmap', and it tells you what character mapping you're using. In this case, UTF-8, or Unicode.

Before we end this lesson, be aware that if you have a Linux system that is set up to use one type of character encoding, but then you change it at some point, you could end up having trouble viewing files that were created previously, when the system was using its old encoding. Fortunately, all is not lost.

Converting Encoding with Iconv 10:10-10:54

You can convert a file's encoding from the old encoding to a new one using this command right here, iconv. The syntax is to enter 'iconv' at the shell prompt followed by '-f'. And then enter the original encoding that the file used, and then '-t' followed by the new encoding that you want the file to use. Then you specify '-o' followed by an output filename where you want to write the new file to. And then, of course, you specify which file needs to be converted.

Basically, it doesn't touch the original file. It just takes that original file, runs it through the conversion process using the source and destination encodings, and writes it out as a new file.

Summary 10:55-11:15

That's how you work with locale settings on a Linux system. Remember, your locale is defined by your locale environment variables, and these variables are populated using a language code, a territory code, a code set, and then an optional modifier.

We also talked about using the locale command to view the various values of your locale environment variables.

And then we talked about using iconv in order to change the encoding of a file.
