3.3.2 Configuring Locale Settings

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

Configure Locale Settings 0:00-0:47

In this demonstration, I'm going to show you how to configure locale settings.

Locale settings are used to localize the shell, and sometimes GUI environments, such that the time, date, and the currency formats, as well as the language, all match what is typically used in the location where you live or work.

For example, if you live in the United states, your monitory system is based on dollars and is represented by a dollar sign ($). However, if you live in England, your monitory system is based on the pound and is represented by a pound sign (£).

Likewise, if you lived in Canada, you may speak French and would probably find it more convenient to view the man pages, for example, if in your native language.

All of this can be adjusted by configuring the desired locale settings.

Using the 'locale' Command to View Variables 0:48-1:11

Locale settings are controlled by environment variables.

To view these variables simply type 'locale' at the shell prompt.

As you can see, a list of the various locale environment variables are displayed, along with their values.

For example, LC_TIME is an environment variable that determines if the time will be shown in 24-hour format or in 12-hour format.

Variable Syntax 1:12-4:53

Each locale environment variable consists of two parts, the variable name followed by the assigned value.

To see how these variables work, let's first look at how the current time for this system is configured. When I type date, I see that the date and time are shown in the typical English US format.

But if I wanted to temporarily change the time format for my system to what a French Canadian might like, I'd begin by first typing the name of the variable - LC_TIME, followed by an equal sign.

Everything after the equal sign is the value assigned to that variable.

When assigning a value to a variable, you first specify the language you want that variable to use. This is done by specifying the language code such as ‘en' for English for our case ‘fr' for French. Notice that the language code must be entered in lowercase and is based on the ISO 639 standards.

Next comes the country code. This is separated from the language code using an underscore. For example, you may want to specify the country code of ‘US' for the United States, or ‘CA' for Candida. Note that the country code must be specified in uppercase and it's based on the ISO 3166 standards.

Next, we add a period which is followed by the character set. Although there are several character sets available, UTF-8 is the most common one used.

Additional option can also be added if needed by adding an @ sign after the character set and then include a modifier. This modifier is used to specify other locale attributes, such as a particular dialect, or maybe a different type of currency. For example, the modifier is frequently used in Europe to differentiate between countries that use the euro, and those that don't, but still speak the same language.

Now let's export the LC_TIME variable and run 'locale' again.

Notice that LC_TIME is now set to the French language and will be shown using the Canadian format.

To see this, I'll type ‘date' again, and you see that the date and time are now in French and in a format that was different to the US style.

An important thing to remember is that even though I could go through and set each variable one at a time, if I know that I want all of my variables to be set using the same options, I could have just changed the L A N G or LANG variable, which in turn would then be applied to all of the variables accordingly. The nice thing about using the LANG variable is that you still have the option to then change individual variables as needed.

For example, using the LANG variable lets try setting everything to use German.

To do that, we once again type the name of the variable LANG, but then use the language code of ‘de' and a country code of ‘DE' as well, followed by the UTF-8 character set. Now, we'll export our variable and view the changes.

Notice that the German settings have been applied to all of the variables, except the TIME variable. This variable was not changed since I explicitly assigned that variable earlier.

In addition, if you look close, you can tell which variables were inherited from the LANG variable, as those all have quote marks surrounding them, while the variable that was explicitly assigned does not.

Also notice that as I do something that generates text, like entering an invalid command, that the error message is being displayed in German.

Making Permanent Locale Changes 4:54-7:36

Since this method only changes the variables temporarily, you also need to know how to make the same type of changes permanently.

This is done using the ‘localectl' tool. This tool supports an option and a command.

For example, to see what my current permanent locale settings are, I can type localectl status. Notice that even though we changed most of the variables to German just a minute ago, that since that was only temporary, this command shows what the system would go back to if I were to reboot my computer.

By entering localectl list-locales, the tool lets me see what language and country codes are available to my system, with the ‘less' command letting me see them one page at a time.

Notice that I have some German options, some English options, and some French options.

To make the changes to my locales permanent, I'll use the ‘set-locale' option, followed by the desired change.

Before doing that, I'm doing to restart my computer to set everything back to the default of US English.

Now that the computer had been restarted, notice that when I type locale, everything is set back to the default stetting's.

These default settings are saved in the /etc/default/locale file and can be seen using the cat command (cat /etc/default/locale). Notice that the default for this system is English and US.

To make a permanent change, simply type ‘localectl', and the option to change a variable, which is ‘set-locale' and from this point on, it's the same as we did earlier.

In this example, I know this computer is going to be used by a French speaker in Canada, so I'm going to use the LANG variable to change all of my variables at once, so I'll type:

Localectl set-locale LANG=fr_CA.UTF-8

And then I'll reboot my system to the activate the change.

Now that my system has been rebooted, you can see the change, starting at the login dialog.

After logging on and opening a terminal, you see that the headings for the terminal are in French and when we look at our locale settings, everything set French Canadian.

Likewise, if I run a command such as ‘man', that the results are in French as well.

Summary 7:37-8:00

That's it for this demonstration.

In this demonstration, we talked about configuring our locale environment variables.

We first used the 'locale' command to view the value of our environment variables. We showed how to temporarily change one or all of these variables and explained the precedence rules associated with locale variables.

And we ended this demonstration by showing how to permanently set the local variables using the localectl command.