

## 10.5.1 Time Zone Settings

---

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

Time Zone Settings 0:00-0:30

If a Linux system is moved to a different physical location, you may need to reconfigure your time zone settings, depending upon how far you move the system. At first glance this might seem like a trivial task.

However, you need to understand that time on a Linux system is managed differently than on other operating systems that you might be familiar with. In this lesson we're going to review, first of all, how time works on Linux and then we're going to talk about configuring the time zone.

---

View Linux Clocks 0:31-2:57

Let's begin by reviewing how time works on a Linux system. The first thing you need to understand is that there are two clocks that you need to be concerned with on a Linux system.

The first clock is the hardware clock. This clock runs independently all the time, even when the system is powered off, and that's because it's implemented within a chip on your motherboard. In fact, there's the battery right there that runs the clock.

Because it runs on battery, the clock still runs even if the system is powered off or even if the system is unplugged. This hardware clock is referred to by a variety of different names. You might hear it referred to as the RTC, which stands for Real Time Clock. You might even hear it referred to as the BIOS clock.

You can manage the time of the hardware clock on your motherboard using the `hwclock` command at the shell prompt. For example, you could use `hwclock` to set the current time or even the current date.

There's also a second clock that runs on a Linux system. This is the system time. This clock does not run in a hardware chip on the motherboard like the hardware clock does. Instead, it runs via software inside the Linux kernel itself.

And here's something that confuses a lot of new Linux system administrators, and that is the fact that system time is measured as the number of seconds that have passed since midnight, January 1st, 1970, UTC. Or, in other words, it measures the number of seconds that have passed since the very last day of 1969.

It's important to remember that the hardware clock on the Linux system and the system time on the Linux system may not actually be the same. When you're managing a Linux system, we need to be focused on the system time. We're more concerned about the system time than the time reported by the hardware clock itself.

Really, the rule of the hardware clock is simple. Its job is to just keep time when the system is powered off. The system time is synchronized to the hardware clock time when the Linux operating system initially starts.

After the system has booted, Linux refers only to system time. The hardware clock at that point is ignored until the next time that the system is rebooted.

---

Use Linux Time Options 2:58-6:00

When setting the time on your hardware clock, you have two different options to choose from. The first one is UTC, or Universal Time Coordinated. UTC is the standard by which the world regulates time so that everything stays in sync.

UTC refers to the current mean solar time at the royal observatory in Greenwich, England--right about here. Therefore, UTC is often referred to as Greenwich Mean Time, or GMT as well. Your local time is then determined by either adding or subtracting one hour to or from UTC for each time zone that you are away from Greenwich, England.

So for me, my time is UTC -7, right about there. Therefore, if it's 11:58 AM my local time, it's 6:58 PM in Greenwich, England UTC.

If your hardware clock is set to UTC here instead of your local time over here, then the Linux kernel will automatically calculate your time offset, including daylight savings time, if applicable, for your local time zone in order to come up with the local time. That is one option for setting your hardware clock.

The other option of course is to just set it to the local time. So, which one is best? Well, when you're installing Linux, you will probably be prompted by the distribution installer to specify which way your local clock is set.

You have to say, "I have my clock set to UTC" or "I have my clock set to local time," and it's really important that you select the correct option. Otherwise, your time zone offset calculations will be wrong, and the system will be configured with the wrong time.

In actuality, setting your hardware clock to UTC is the preferred option when you're working with Linux systems. Using UTC makes calculating your time zone differences a lot easier. For example, let's suppose you install a groupware application on your Linux system that manages emails, instant messaging, and so on.

When you do this, you should set your hardware clock to UTC because it will make it easier for the groupware application to coordinate timestamps that are assigned to messages for all the different computers that are going to be using that system, because they may be located in different time zones.

If you don't use UTC, then it's possible for you to be getting messages that are time stamped from four in the morning, when they really weren't sent at four in the morning. They were instead sent at, say, 9:00 in the evening.

But because the hardware clock was set to local time, the time zone calculation is off, and all of your time stamps on your messages are goofed up. On the other hand, if all of your systems-- including your server-- are set to use UTC, then the time stamps will be consistent, no matter what time zone a client system happens to reside in.

---

### Configure Time Zone 6:01-7:04

With this in mind, let's talk about how you go about configuring the time zone on a Linux system. During the initial installation of your Linux system, you're prompted to specify which time zone the system is located in. Where that is saved depends upon the distribution.

For example, if you're using a Debian-based distribution, such as Ubuntu, the time zone you specify is saved in this file right here: `/etc/timezone`. An example of this is shown here, where we use the `cat` command to view the contents of the `/etc/timezone` file. And it tells me I'm in the America/Denver time zone, which is UTC -7.

RPM-based distributions, such as Fedora, Red Hat, OpenSUSE, and so on, will save this information in a different file. It will save it in `/etc/sysconfig/clock`. And you can see the syntax down here. It just says `TIMEZONE=` and then the name of the time zone, which in this case says "America/Boise", which is exactly the same time zone as America/Denver; it's UTC -7.

---

### View the Current Time Zone 7:05-7:21

You can view the current time zone setting by entering the `date` command at the shell prompt. In this example, you can see that my time zone is set to MDT, which stands for Mountain Daylight Time, which again is UTC -7.

---

### Change the Time Zone 7:22-8:38

If you need to change the time zone setting on a Linux system after it's initially installed, you can use one of two different commands. For Debian distributions, you use the `tzconfig` command. Other distributions, like Fedora, Red Hat, openSUSE, and so on, use the `tzselect` command.

Be aware that you cannot run these commands as a standard user. You need to have root-level privileges in order to run them. An example of using the `tzselect` command is shown here.

The first thing the `tzselect` command does is display a list of continents and oceans, and you need to enter the appropriate number for the region where the time zone you want to switch to is located. For example, if my time zone is located in North America, I would enter 2, then specify The Americas. Then I would have to specify the country where the system is located, and then the appropriate time zone.

You can see an example here, where I've specified that I'm in the United States and I want to use Mountain Time. And it tells me that the Time Zone Environment Variable, `TZ`, is going to be set to America/Denver. And I have to specify "Yeah, that's okay" by entering 'Y' and selecting Yes.

---

### Change the Time Zone Environment Variable 8:39-9:58

Because TZ basically just sets the value of the TZ environment variable, you don't have to use it. You can actually change time zones from the shell prompt without using the tzselect command by simply setting the value of the TZ environment variable and then exporting it.

And this can be useful in situations where you don't have access to the root account for the system, or if you want to use a different time zone for just your account without changing the time zone that's used by other users.

The syntax for doing this is shown here. We enter 'TZ=', followed by the name of the time zone file we want to use, and a list of all the available time zones can be found in the directory shown here. And then after setting the value of the environment variable, you would export it.

In this example, I've actually done both at once. I run export, and then I set the value of the TZ environment variable all in one command. But of course, as with any other environment variable, this change is not persistent.

If I were to reboot the system, my time zone change would be lost. If we want to make it persistent for just my user account, I could open up the .profile hidden file in my user's /home directory and enter the command there. For example, export TZ= and then in tick marks 'America/Denver'.

---

#### Change Time Zones in /etc/localtime 9:59-11:04

There is one last way we can change time zones that we want to look at before we end here, and that is changing time zones using the /etc/localtime file.

The thing you need to remember is that this file right here is not a real file. It is instead a symbolic link that points to the appropriate zone file located in the /usr/share/zoneinfo directory.

In this example, I want to switch my time zone to Mountain Standard Time in the United States. I use the ln command to create a symbolic link, and I use the -sf option, followed by the zone file, the time zone file, that I want to point to.

In this case, it is the Boise time zone file located in /usr/share/zoneinfo/America. And then I specify the name of the symbolic link that I'm creating, which is /etc/localtime, and down here I've used the ls -l command to view this file, and you can see that it points to the appropriate zone file in /usr/share/zoneinfo.

---

#### Summary 11:05-11:09

That's it for this lesson. In this lesson, we've discussed how time is tracked on Linux and also how to manage Linux time zones.

---

**Copyright © 2022 TestOut Corporation All rights reserved.**