10.5.4 Time Maintenance

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

Time Maintenance 0:00-1:21

In this lesson, we're going to discuss time maintenance on Linux. When you initially set up a Linux system, you're prompted to configure your system time along with your time zone. When you do this, you're probably not overly concerned about these settings, because they don't really seem all that critical to the functioning of the system.

However, understand that when you're working with Linux it's actually really important that you maintain the correct system time, especially if the system is going to provide network services.

For example, let's suppose you're implementing a file server. Well in this situation you need to ensure that the time stamps on each of the files are accurate. Or maybe the server system's going to be providing some type of directory service. Or maybe it's a database server.

In these situations, maintaining the correct system time is absolutely critical.

For example, let's suppose you're running a database server, then you need to make sure that each change that's submitted to the database server from client systems has an accurate time stamp.

Imagine what kind of problems would arise if database modifications get applied out of sequence because the time stamps are inaccurate. The integrity of your data would be toast. So in this lesson, we're going to look at setting the hardware clock with the hwclock command and then we're going to talk about synchronizing time with the netdate command.

Using Linux Time Sources 1:22-3:01

Let's begin by discussing how to manage time with hwclock. Now remember, there are two time sources that are used on a Linux system, the first one is the hardware clock, which is integrated into your systems motherboard and it runs all the time even when the system's powered off or even when it's unplugged for that matter.

We also have the system time. Now the system time runs via software inside the Linux's kernel self, therefore it runs only while the system is up and running. Also remember that system time is measured not as the current date and time, but instead as the number of seconds that have elapsed since midnight, January 1st, 1970 UTC.

It's important that you remember that the hardware clock and the system time may not be the same. Therefore, when we're managing a Linux system, we are actually most concerned about the system time, than the time that's currently being reported by the hardware clock.

Really the role of the hardware clock is simple. It's job is to keep time while the system's powered off. When the system is powered on, what will happen is that the Linux operating system will go out and grab the time from the hardware clock and use that to initially set the system time.

Once that's done, Linux cares only about the system time, it could care less about what time the hardware clock has. The only time the hardware clock comes back into play is if you reboot the system at which point the kernel will grab the hardware clock time again to initially set the system time.

Managing the Hardware Clock 3:02-4:43

So to manage the hardware clock on your system, you use the hwclock command. It can do a lot of different things, as noted here. It can display the current time. You can use it to set the hardware clock time. You can use it to synchronize the hardware clock to the system time, or you can go the other direction and synchronize the system time to the hardware clock.

And just a quick note, in addition to using the hwclock command, you can also view the current time of the hardware clock, not the system time, but the hardware clock time by simply displaying the file that you see here, say with the cat command, /proc/driver/rtc. rtc stands for real time clock.

There are several different options you can use with the hwclock command. These are available on most Linux distributions, you can use the -r command, to read the current time from the hardware clock and just display it on the screen.

If, on the other hand, you need to set the hardware clock time, you can use the --set --date command. Be aware that whatever time you specify right here will be set in local time. And this is true even if you've got your hardware clock set to UTC, the kernel will take care of

calculating the appropriate offset.

The next option is -s, this is used to set the system time to the current time of the hardware clock. -w goes the other direction and sets the hardware clock to whatever the current system time is. And then we have the --UTC option which specifies that the hardware clock time is configured to use UTC, or if we want to set it to use local time we use --localtime.

Synchronizing Time 4:44-8:03

Be aware that the scope of the hwclock command's functionality is limited to just your local computer system. That's fine and dandy but if your computer system is operating on a network, then it's very likely that you're going to need the system time on all of your Linux systems to be synchronized.

You don't want one system having a radically different time than other systems, because as we talked about earlier, that screws up time stamps.

One tool that you can use to synchronize time between Linux systems is the netdate command. You can use the netdate command to synchronize the time on the local system with the time on a time server somewhere on the network. And the syntax is shown here where we enter netdate, followed by the IP address of the time server.

And you could also use the DNS name as well. The key thing is that the system that you specify has to be running the time service.

So in this example, I have a time provider over here. It's a server system. The time daemon has been enabled on this server, and it is running. And it's synchronizing time over UDP port 37. That's the default port, you can of course change it. What the time server's going to do is listen on port 37 right here, for time synchronization requests.

So let's suppose we run the netdate command over here on the time consumer, this workstation. It'll send the time request from the netdate command over here to the time provider.

The time service on this server running on port 37 hears the request, and then sends the appropriate time back to the time consumer over here. At which point, the time on this system is set to whatever time it received from the server system. As a result, the time on these two systems is really, really close, within a few milliseconds of each other which for all intents and purposes, we consider to be synchronized.

It's important that you remember that when you are synchronizing time with the netdate command you are synchronizing system time, not the hardware clock time.

Therefore, if you want to make sure that your system time comes up reasonably close to the synchronized time the next time you reboot the system, it's not a bad idea to run the hwclock command with the -w parameter to synchronize the hardware clock on your system to the system time.

That way, the next time you boot, your system time should be fairly close to the time over here on the time provider, because remember the Linux kernal will read the initial time for the system from the hardware clock which should be fairly close to the time provided by the time provider.

There's one more thing that I want you to remember here with regards to system time. Remember that we said that system time does not run on hardware, it runs as software. What this means is, after you synchronize time with the netdate command, eventually the time on this system will drift away from the time on this system.

Therefore, it's not a bad idea to run the netdate command on a regular basis using a cron job, that way we keep the time synchronized regularly between these two systems.

Summary 8:04-8:10

That's it for this lesson. In this lesson, we discussed system time maintenance, we looked at setting the hardware clock with the hwclock command, we also looked at synchronizing the system time with the netdate command.
