

## 10.5.5 Managing System Time

---

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

Manage System Time 0:00-0:12

In this demonstration, we're going to discuss managing time on a Linux system. You can view the system time by using the 'date' command at the Linux shell prompt.

---

View System Time 0:09-1:01

Here we see the date and the current time. Viewing the time on the hardware clock requires that we use different options. One option is to go into the '/proc/driver' directory.

We do an 'ls' command here. We see a file named rtc. rtc stands for real time clock, and that refers to the hardware clock on the motherboard of the system. We want to see what time it is on the hardware clock.

We can just use the 'cat' command to view the 'rtc' file. Here we see the date, and here we see the time on the hardware clock in the system.

Notice something very significant here. The hardware clock is 6 hours ahead of the system clock. Why would that be the case? Why would they be so different? It has to do with time zone offsets.

---

Use Time Zone Offsets 1:02-2:04

My hardware clock is set to Greenwich mean time, but because I'm several time zones behind Greenwich mean time--my local time is actually 12 noon, 12:20 PM, while Greenwich mean time is 6:20 in the evening.

You can manage the time on the hardware clock using the hwclock command. You can use hwclock to view the current hardware clock time. You can use it to set the hardware clock to match the system clock, or vice versa.

You can set the system clock to whatever time it is on the hardware clock. If you just want to view the current time on the hardware clock, you use 'hwclock -r'.

Notice something very interesting down here. In the output, it's basically the same as the date command. In other words, the time zone offset is being calculated. The system knows that we are 6 hours behind Greenwich mean time.

Hence, the output is adjusted 6 hours backwards to display the current local time, which as you can see is not really the same as what is really on the hardware clock.

---

Set the System Time 2:05-3:10

If we wanted to set the system time to match the hardware clock time, we could use the 'hwclock' command again, but this time we would use the '-s' option. We could also do just the opposite. We could set the hardware clock to match whatever time we have on the system. To do that, we would change it to the '-w' option.

You can also use the hwclock command to directly set the time and date on the hardware clock itself. To do that, we use the '--set' option, and then we specify '--date', and then in quotation marks we enter in the date and then the time that we want to set the hardware clock to.

So in this case, it would be '09/09/15'. That's today's date. Space, and then what is it? It's '12:23:00'. Hit Enter, and that would set the hardware clock. I'm not actually going to do that, because I would then be several seconds off because I don't know exactly what time it is right now.

But if you needed to set the hardware clock without powering the system down and going into your CMOS setup program, you could do it here from the command line.

---

Synchronize Time Between Network Hosts 3:11-4:45

With that in mind, let's shift gears a little bit and talk about synchronizing time between network hosts. Here's the problem we run into. We may have all the hardware clocks in our Linux systems in our network all set the same, so when our Linux kernel first boots the system time on all of our hosts are identical, but system clocks are not very accurate.

They're implemented as software, so there's a lot of variables that can affect how accurate they are. In fact, it's very common to see system time drift between different systems that were initially synchronized. The longer you let the systems run, the greater the drift is going to be. I've seen some really radical amounts of drift between network hosts.

This can be a problem because it causes some bad things to happen. For example, it can cause inconsistent time stamps. If you have two systems performing operations in a database, and the database is programmed to say, "Well, the first command I receive is the first one I write, and the second command I receive is the next one I write."

If it receives these commands that have different time stamps, you could end up having the second command being written before the first. One option for synchronizing time between network hosts is to use the network time protocol.

That's actually the preferred option. It's the best way to do it. However, there is an older command that you could also use called the `netdate` command. `netdate` is older and it is not available on all distributions. It is available on this particular openSUSE Linux distribution, so that's the one we're going to use here.

Basically, we tell `netdate` to go out on the network, contact a time server, get the time from the time server, and then set the local system clock to that same time.

---

### Run the Time Service 4:46-8:00

In order for this to work, the system we want to get the time from must have the time service running, so let's actually go do that first before we run the `netdate` command. I have another Linux system on the network that we're going to use as a time server.

We need to go over there and make sure that the time service is actually running on that system. If the time service isn't running, you can't sync time with it. So I'm going to run the `vncviewer` application and the IP address of that remote host is '10.0.0.3'.

I want to connect to display port ':1', and I'll come down to Computer. Then I'll go to YaST, and we want to go to Network Services. Then I want to go to Network Services (xinetd).

What we want to do in the list of services is scroll down to the Time Service. Here's time right here. Notice that the time service can run using either TCP or UDP. Let's go ahead and enable both. First I'll select TCP, turn it on.

Then let's do the same thing for UDP. I'll hit Finish, and the time service is now running on this system. Before we go any further, you do have to open up the appropriate network ports on this system. Otherwise, it can't answer time synchronization requests.

You have to have port 37 open, which I've already done, so we don't have to worry about it here, but be aware that is something you have to do. Otherwise, the request will be blocked. I'm going to go ahead and disconnect here, Log Out.

Now that the time service is running on that server, I can type `'netdate'`, followed by the IP address of that server. In this case we'll do '10.0.0.3'. Enter. It goes out and finds out what time it is on the remote server, and then immediately sets my local time to that same time.

As you can see here, we weren't actually very far off. You can see that we were slightly less than one second off from that remote system, so the time shift wasn't dramatic. Now we've got a problem.

If I down this system and boot it back up, it's going to go look at the hardware clock again and set the system time to the hardware clock, and so I'm going to be one second off again from that other server.

If we wanted to, we could use the `'hwclock'` command with the `'-w'` option to go ahead and now immediately set the hardware clock to the system time, which we just barely got from the server. That way if I were to reboot the system, they should be really close to each other. It's done.

There's an important thing you've got to understand about the `netdate` command, and that is the fact that it syncs time only once, when I run the command at the command line.

It doesn't continually keep time in sync, so if you wanted to use `netdate` for time synchronization, then you'd have to create a cron job that runs the `netdate` command on a particular schedule to keep your system time in sync with the time server.

Depending on how badly the software clock drifts on your system, you might need to run it once a week. You might even need to run it once a day.

**Summary 8:01-8:16**

That's it for this demonstration. In this demo, we talked about managing time on Linux. We first talked about how to view the hardware clock time. We then talked about how to manage the hardware clock time with a `hwclock` command, and then we ended this demonstration by talking about synchronizing network time using the `netdate` command.

---

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