

Automatically Set System Time On ARM Devices

Most users know how unforgiving the Slackware operating system is when the date and time is not set relatively accurate, and rightly so. Inaccurate time can give rise to all kinds of problems, and is often the bane of a system administrator's life. Particularly on ARM devices, where this can be a continual annoyance when the system time has no way of setting itself correctly and/or those responsible [often] forget to do it.

Preface

The information on this page aims to enable Slackware users with an Internet connection and/or RTC installed on ARM devices to automatically set the system date and time, via a `/etc/rc.d/rc.local` script, and hopefully provide insight into the different means and methods that can otherwise be employed.

Notes

Most (if not all?) modern PC systems come with an onboard real time clock (RTC) installed on the mainboard, whereas most single board computer ARM devices do not.

For example, the Raspberry Pi (a.k.a. the most common and popular single board computer on the planet) cannot seem to factor-in an onboard timekeeper within its MSRP, even on the more expensive models which feature larger amounts of RAM but still no RTC. The Banana Pi, Orange Pi, and Asus Tinkerboard devices do not feature an onboard RTC. Less common devices, such as SolidRun's HummingBoard Pro, the NanoPi (Neo4 or R1 or M4), the Olimex Olinuxino-A64, and the Rock Pi 4, all come with an onboard RTC installed. So some single board computer ARM devices do have them built-in, while others do not.

On those which don't have a RTC as standard it's prudent, and easily affordable, to buy a RTC module and install it yourself, with some [RTCs being available on eBay for only a few dollars](#) inc. delivery!

Requirements

There are a number of ways to achieve automatically setting the date and time accurately on any computer system. You will require at least one of the following...

- Access to an Internet connection.
- A battery backed-up onboard RTC installed on your ARM device.
- A battery backed-up onboard RTC on another Slackware system connected to your local network.

If you do not have at least one of the above then you're going to have to set the date and time on your ARM device(s) manually, as and when required. Here's a few examples...

```
root@slackware:~# date MMDDhhmmYYYY.ss
root@slackware:~# date -s "DD MMM YYYY hh:mm:ss"
root@slackware:~# date [-u|--utc|--universal] [MMDDhhmm[[CC]YY][.ss]]
```

Automatically setting the system time with an Internet connection

The quickest and laziest method is with the Bash script code below, which is a representation of an '/etc/rc.d/rc.local' file which will set the system date/time on boot-up and also set the date/time on an installed RTC. Everything is done *automagically* and is effortless...

rc.local

```
#!/bin/sh
#
# /etc/rc.d/rc.local:  Local system initialization script.
#
# Put any local startup commands in here.  Also, if you have
# anything that needs to be run at shutdown time you can
# make an /etc/rc.d/rc.local_shutdown script and put those
# commands in there.

# test for network interface connection carrier
for NIF in $(ls /sys/class/net/ | grep -v lo); do
    if [[ -n $(cat /sys/class/net/"${NIF}"/carrier 2>/dev/null) ]]; then
        # set system time from ntp server [remote]
        /usr/sbin/sntp -Ss 0.pool.ntp.org > /dev/null 2>&1 || \
        /usr/sbin/sntp -Ss 1.pool.ntp.org > /dev/null 2>&1
        # set RTC from system time [if detected]
        [ -h /dev/rtc ] && /sbin/hwclock -w > /dev/null 2>&1
    elif [ -h /dev/rtc ]; then
        # set system time from RTC [if detected]
        /sbin/hwclock -s > /dev/null 2>&1
    else
        # output to stderr if time sync fails
        echo "WARNING! : system time sync ERROR ... " 1>&2
    fi
done

#EOF
```

"The Mechanics" : The above '/etc/rc.d/rc.local' script runs when the Slackware ARM system is booting and looks for a network carrier and then, if one is found, it runs the 'sntp -Ss pool.ntp.org' command. Then, if a RTC device is detected, it sets the time on that too. Otherwise, if no network carrier is detected, it looks for a RTC device and sets the system date and time from it. Otherwise, if no network carrier or RTC is found, it sends an error message to 'sdout' and 'sdterr' (echo 1>&2).

I've got this code in the '/etc/rc.d/rc.local' initialisation script on all my ARM devices. It does a remarkably good job and has never failed. This code works when there's no Internet connection because all my ARM devices also have a battery backed-up [DS3231] RTC installed. So, either way, my system is going to have the date and time set by one means or other. Although in reality, I've never seen the error message appear anywhere. Even when the battery depleted on a ChronoDot RTC that I've had for +9 years this script got the date and time from my local network's NTP server anyway. Instead of specifying an Internet-based NTP server (e.g. 0.pool.ntp.org), I have my scripts query the IP address of an NTP server (e.g. 192.168.1.17 on port 123) on my local network, which I have running on one of my Raspberry Pi 2 devices.

Setting the system time with a local NTP server

The SlackDocs [NTP](#) page takes users through the process of how to configure and maintain a Slackware NTP server. It's a very easy and straight-forward exercise. So, if you have an ARM device just laying around, gathering dust or not doing much, it might be advantageous to put it to some use for this purpose.

Once you have a NTP server running on your local network, you are able to set any other computer system or device that uses a time protocol from it; Linux systems, Windows computers, Android devices, tablets, smart phones, etc. Therefore, it's possible to have everything set with an accurate date and time that's capable of communicating with the NTP server.

Setting the date and time on other Slackware [Linux] systems using a local network NTP server is achieved by typing one simple command:

```
~# sntp -Ss 192.168.1.17
```

On Slackware ARM 14.2 this can be achieved using the following command:

```
~# ntpdate 192.168.1.17
```

In the example above, the IP address is our local network NTP server machine. Obviously you would enter the IP address of the machine running the NTP server daemon on your own local network.

Help

Any requests for help/assistance, questions, suggestions, good or bad or indifferent feedback, can be addressed on the [Slackware ARM LQ forum](#).

Thank you for being interested in automatically setting the system date and time on a Slackware ARM installed device, and in this SlackDocs page.

Sources

Documentation which assisted in this SlackDoc project:

https://docs.slackware.com/howtos:network_services:ntp - by [Niki Kovacs](#)

* Originally written by [Exaga](#) - 2021-03-14 19:08:54 [GMT]

[howtos](#), [slackware](#), [arm](#), [system](#), [time](#), [rtc](#), [rc.local](#), [network](#), [ntp](#), [server](#), [author](#) [exaga](#)

From:
<https://docs.slackware.com/> - **SlackDocs**

Permanent link:
https://docs.slackware.com/howtos:hardware:arm:automatically_setting_system_time_on_arm_devices

Last update: **2021/03/22 09:26 (UTC)**

