Getting TAI time on a Debian machine

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

First publication of this article on 16 April 2024. Last update on of 17 April 2024

https://www.bortzmeyer.org/tai-on-debian.html

It should work by default but, apparently, on some operating systems like Debian, it does not : to get the TAI time, you need a small configuration change. I document it here for myself or for people which will use a search engine and find this page.

TAI is useful because, unlike UTC, it never adds an extra second, neither it misses one (UTC does, because of leap seconds). This makes it convenient, for instance for Internet servers. But how to get TAI time on a Debian machine ?

The official answer is that when you use `clock_gettime` in a C program or `time.clock_gettime` in a Python one, you need to pass the option `CLOCK_TAI`. One can easily check that, on a Debian stable machine (version 12.5), it does not work : you get the same value with `CLOCK_TAI` or `CLOCK_REALTIME` (the typical clock, set on UTC). Unfortunately, no error code will tell you that something was wrong.

It seems that the kernel (which manages the clock and answers to `clock_gettime`) knows only UTC and, to convert to TAI, it needs to know the offset (currently 37 seconds). Debian has a file to do so, a leap seconds table, in `/usr/share/zoneinfo/leap-seconds.list`. This file contains all the information necessary to get TAI from UTC. But someone has to read it and to inform the kernel. This is typically done by ntpd. But it is not done by default, this is why the above test failed.

So, the system administrator needs to configure ntpd to load this file. This is done in `/etc/ntpsec/ntp.conf` (or `/etc/ntp.conf` depending on the version of ntpd you use) by adding this line :

```
leapfile  /usr/share/zoneinfo/leap-seconds.list
```

and restarting ntpd **and** waiting some time for the kernel to synchronize, it is not instantaneous.

If you see in the log file (for instance with `journalctl -n 10000 -t ntpd | grep -i leap`) something like :

```
Apr 16 08:25:39 mymachine ntpd[29050]: CLOCK: leapsecond file ('/var/lib/ntp/leap-seconds.list'): open faile
```

(note the file name, which is not the default one), it means you need to check the permissions of the file **and** that systemd or AppArmor are not adding some restrictions (the default AppArmor profile of ntpd on Debian includes `/usr/share/zoneinfo/leap-seconds.list` but may be you changed something).

You can check that the kernel now knows the truth, for instance with a simple Python session :

```
% python
Python 3.11.2 (main, Mar 13 2023, 12:18:29) [GCC 12.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.

>>> import time

>>> time.clock_gettime(time.CLOCK_TAI)
1713284374.8322737

>>> time.clock_gettime(time.CLOCK_REALTIME)
1713284337.8329697
```

You can see that there is indeed 37 seconds of difference (plus a small value because of the delay between the two commands).

That's all. You can now use TAI in your programs. The file `/usr/share/zoneinfo/leap-seconds.list` is automatically managed by Debian (it is part of the package `tzdata`, and the reference version is , itself a copy of , itself made by the Paris observatory) so you don't need the programs like `ntpleapfetch` which are necessary on other operating systems.

For instance, on a Slackware system, the file `leap-seconds.list` is not provided by default (there is a file named `/usr/share/zoneinfo/leapseconds`, with a different format, and that ntpd cannot use), so you will need to configure cron to download the proper file.

An alternative is to handle time through a library that will do it for you, such as hifitime `<https://nyxspace.com/hifitime/>` for Rust and Python.

Thanks to Nicolas Sapa, Matthieu Herrb and Kim Minh Kaplan for the useful help.