ELPROMA ELEKTRONIKA

Release: May 2015

Leap second

A **leap second** is a one-second adjustment that is occasionally applied to Coordinated Universal Time (UTC) in order to keep its time of day close to the mean solar time, and UT1 time. Without such a correction, time reckoned by Earth's rotation drifts away from atomic time (TAI) because of irregularities in the Earth's rate of rotation. Since this system of correction was implemented in 1972, the 26 leap seconds have been already inserted, the most recent on June 30, 2015 at 23:59:60 UTC. Together with first 10 initial seconds total amount of leap seconds is now (July 2015) equal 36s, and therefore the formula for calculating is:

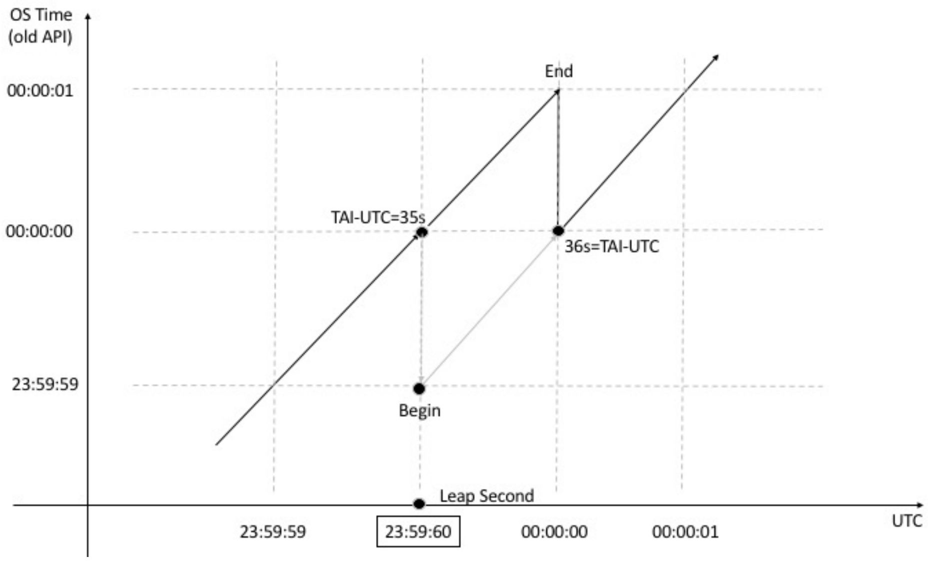
TAI-UTC= 36s

Adding leap second procedure bases on special announcement NTP flag set by decoding special message file from NIST or IERST. The implementation procedure theoretically should give a perfect 60 second and the UTC time clock effect should show as follow:

23:59:59 ➔ 23:59:60 ➔ 00:00:00

But there are a couple o problems why above structure needs an attention during IT system deployment. According to D. Mills article "*A kernel model for precision timekeeping*", there are possible several side effects of getting time deviations depends on end-user operating system (OS) and its kernel version. We would like to point attention to possible several scenarios of supporting leap second depends on OS version and its kernel (e.g. those implemented in old API generation POSIX Linux system but not only limited to). Possible implementations of leap second support at OS kernel are:

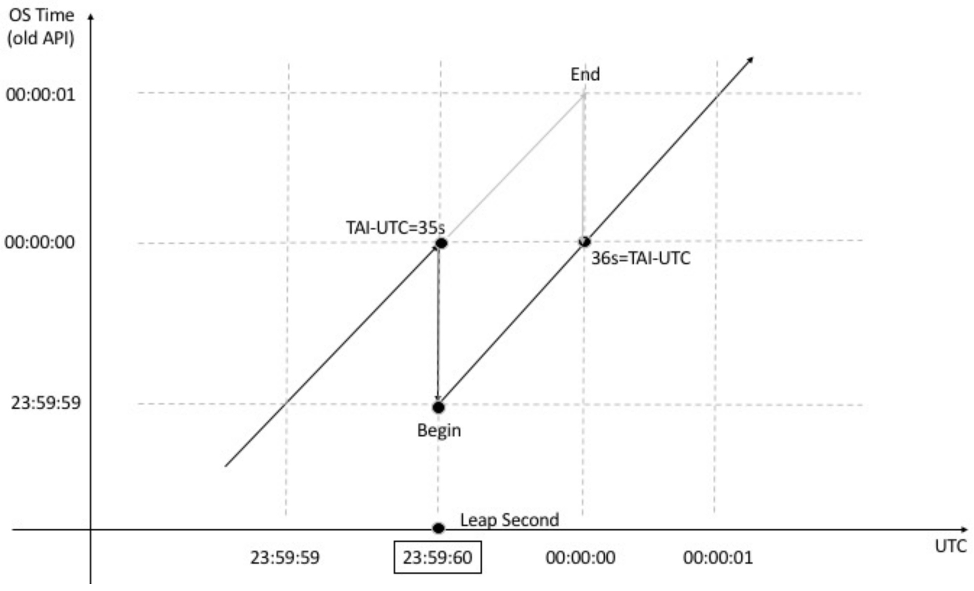
1. Step OS Clock Time Back at the End of the Leap Second



Leap Second - Step Back Clock at the End of Second

In this case time is simply stepped back at the end of an inserted leap second as shown in above figure. Therefore, OS time can not be monotonic, and thus duplicate time stamps occur after the leap second (e.g. at the be ginning of the next UTC day). Mills claims as a result, there can be later time stamps assigned to events which occurred earlier, which can heavily mess up applications using time stamps to order the sequence of events or transactions.

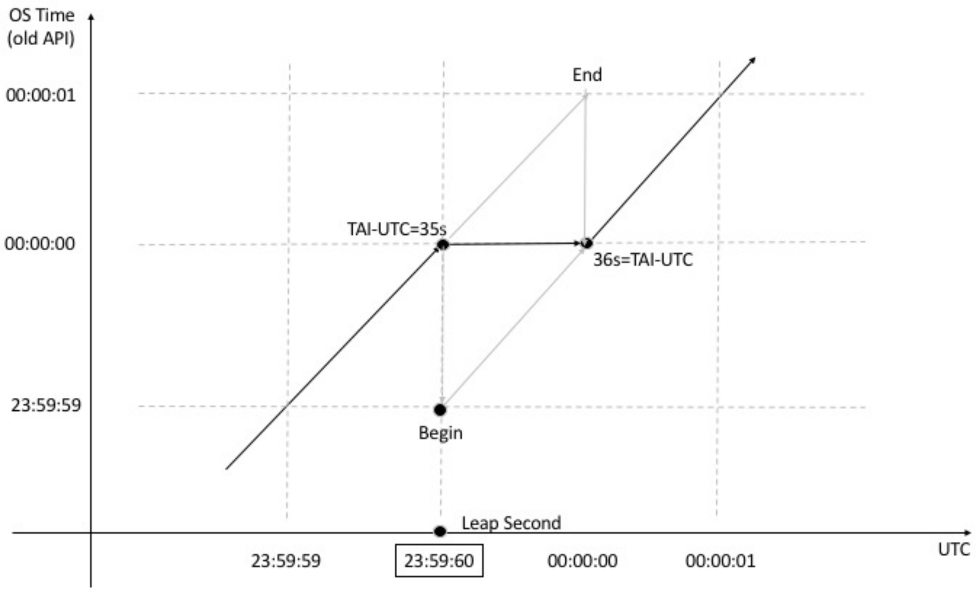
2. Step OS Clock Time Back at the Beginning of the Leap Second



Leap Second Time Back at the Beginning of Second

Time is simply stepped back at the beginning of an inserted leap second as shown above. In this case time is also not monotonic. Mills points the difference from the previous case is that duplicate time stamps occur during the leap second, i.e., at the end of the UTC day. Similarly, there can be later time stamps assigned to events which occurred earlier, which can cause the same confusion as the previous case.

3. Stopping OS Clock Time Counting for Exactly One Second

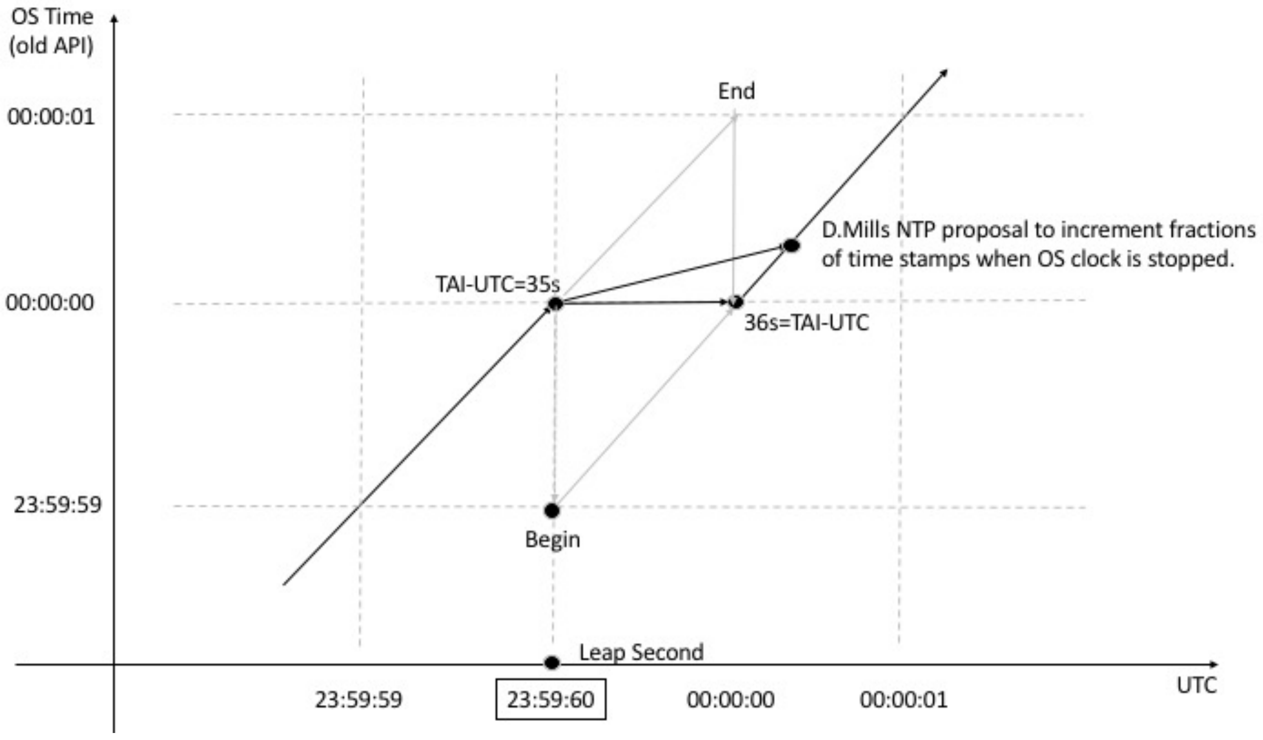


Leap Second Stopping Clock Time for 1 Second

A modified approach which guarantees strictly monotonic time stamps has been proposed Mills, the inventor of the Network Time Protocol (NTP), who suggested stopping the clock during an inserted leap second, but incrementing the fractions of time stamps by the smallest possible time increment whenever the time is read by an application. This technique gives another the 4th minor scenario (see below).

Some operating systems like Microsoft Windows are not aware of leap second and thus are not prepared to handle it. In such case it may be possible to slew the system time over the leap second. Windows slows the system clock down to half the nominal speed for 2 seconds, so the Windows OS time is again aligned to UTC. This method is not optimal, but at least after the leap second the OS time is correct again.

Nevertheless, there is an important conclusion going out of above discussion. Depends on OS version and its kernel, the leap second can be supported on different ways giving offset error to UTC for seconds, and in some cased for hours. Many vendors therefore recommend to not use synchronization a couple of hours before and after leap second UTC midnight.



Dave Mills NTP proposal to Increment Time LSB

OS clock is just put on hold for one second as shown above and time stamps are all the same during the inserted leap second. OS time does not increase monotonically, and time stamps can't be used to order events.