On the P3335 Architecture WG telcon of 17 October 2023, I mentioned that there had been a violent time war some twenty years ago on the matter of should *POSIX Time* be UTC, and it seems worthwhile to document the core issues of that day.

The official definition of *POSIX Time* is in the IEEE 1003.1 standard[1], and most of the explanatory details are actually in the Rationale volume, not the normative parts. Look for "Epoch" and "Seconds Since the Epoch". Also look in the time-related header files.

The following is adapted from my 16 October 2000 posting "Re: (pasc-time-study 128) A Reminder (and summary)" to the PASC Time Study Group. In this, "POSIX Time" corresponds to the "Seconds Since the Epoch" of yore, while "broken-down time" is what comes out when one converts POSIX Time into YYYY MMM DD HH:MM:SS.sss form.

This was written as requirements with rationale, the intent being to lay out the full scope of what would be required to make POSIX Time identical to UTC. This proved to be far more than the WG was willing to undertake, so these requirements never made it into the POSIX standard, which instead made it clear that POSIX Time was *not* UTC, that the seconds were nominally the SI kind, and that a day contained *exactly* 86,400 seconds, precisely to reduce confusion and to cut off all further claims that POSIX Time was UTC, or ever would be.

The intent here is to start the requirements and objectives discussion.

After thinking about all the recent time-related traffic on the Austin Group[2] reflector, and the above, I would suggest the following stipulations:

1. The primary timescale of POSIX shall be called "POSIX Time", which is defined below. It shall **not** be called UTC unless it **is** UTC in every respect; that is, completely conforming to ITU TR 460-4 or its successors. (ITU TR 460-5 just became official, but changes nothing relevant to the present discussion.)

We need to make it quite clear that POSIX is defining its own custom timescale, with its own definition, properties, and audience. Any resemblance to timescales historical, living or fictional, is purely coincidental.

2. The first [most important] objective of POSIX Time is to establish causal order of file modification dates, transaction messages in distributed systems, and the like[3]. This implies that POSIX Time must be at least monotonic non-decreasing.

[1] The POSIX documents are available at <https://pubs.opengroup.org/onlinepubs/9699919799.2018edition/>.

[2] The Austin Group is where POSIX standards are developed.

[3] People knew that using timestamps to deduce causal order was not perfect, but when software teams were maybe twenty people, mis-ordering errors were rare. But with teams of at least one hundred, overlaps were common, and

The second objective is that the difference between timestamps shall be an accurate measure of the physical [wallclock] time elapsed between those timestamps. The third objective of POSIX Time is to provide a humanly useful timescale to applications and users. The fourth objective of POSIX Time is to provide to applications and users an accurate implementation of one or more of the internationally standardized clocks such as UTC, UT, UT1 (~GMT), UT2, TAI, etc.

3. POSIX Time must be defined to support any combination of totally isolated workstations, groups of workstations in a network, and workstations connected to the world via the Internet and/or hardware time receivers and distribution systems.

Support for isolated systems implies that the underlying POSIX Time cannot be sensitive to variations in the motion of the Earth, because these motions cannot be predicted in advance, and so cannot be known to isolated systems.

Thus, POSIX Time cannot be UTC, GMT (~UT1), UT, UT1, UT2, or the like, as these timescales are manually adjusted to follow variations in the motion of the earth, based on astronomical observations, and these adjustments cannot be known to isolated systems.

4.   GPS time receivers are the most widely used form of time receiver used in computers. GPS receivers yield "GPS System Time" (which is TAI plus a fixed 19-second offset) and UTC (computed from GPS System Time and other data contained in the GPS signal, including the current count of past leap seconds and schedule for the next leap second).

GPS System Time rolls over every 1024 weeks (19.7 years), so some way to determine which GPS rollover cycle we are in is needed in practical systems; GPS alone isn't sufficient. Future upgrades to GPS itself may solve this problem, but any such solution will be phased in over many years, perhaps decades.

5. Hardware clocks on computers tick at a constant (albeit slightly adjustable) rate, and thus a second on such a computer is of constant physical duration, the intent being that the computer's second be of the same duration as the second used in civil time, which is the SI Second used in UTC. Therefore, the most natural definition of a "second" in POSIX Time is that it equals the SI Second to the (unspecified) accuracy of the computer clock hardware.

In this, we will conspire to ignore the effects of NTP, which will cause the length of the second to vary slightly as the local clock is steered into synchronism with the chosen master clock or clocks. Once NTP has achieved synchronism, the local

the wheels flew off SCCS. A centralized ordering mechanism was implemented to replace SCCSs. <https://en.wikipedia.org/wiki/Source_Code_Control_System>

length of the second will be very accurately matched to that of the master clock or clocks, despite inherent inaccuracies in the local computer clock hardware. So, we are justified in ignoring the effect of NTP here, but should explain this in the 1003.1-200x Rationale.

6. The POSIX Time "Epoch" (timescale origin) is defined to coincide with 00:00:00 UTC 1 January 1970 AD. This epoch is chosen for backward compatibility. While the origin (zero) of POSIX Time is defined as a particular instant specified in UTC, it does not follow that POSIX Time is in any sense UTC.

We may want to adjust the Epoch slightly, so that POSIX-TAI is exactly an integral number of seconds. There are reasons to make the difference eight seconds, and reasons to make it ten seconds, a matter to discuss and decide later[4].

7. POSIX Time is therefore a uniform and continuous timescale, being mathematically smooth (to the granularity of the underlying clock) and monotonic non-decreasing, with a constant rate of progress of one second of indicated time per second of physical [wallclock] clock time (the SI Second).

There is a name for clocks having the above properties: TAI. In short, POSIX Time (expressed as a broken-down time without leap seconds) differs from TAI by an arbitrary constant offset. If the computer has access to an external source of time, even via a human, and its time is set correctly, the offset will be an integral number of whole seconds, being the historical value of TAI at the POSIX Time Epoch. If the computer happens to be isolated, the offset is unknown.

Another way to say the same thing is to say that POSIX Time is defined as the number of SI Seconds since the Epoch, ignoring leap seconds.

A proposed formal definition: "POSIX Time is a uniform and continuous timescale counting SI Seconds and fractions since the POSIX Epoch, 0 hours 0 minutes 0 seconds UTC 1 January 1970 AD, also known as Julian Day Number 244_0587.500 (UTC), or Modified Julian Day 40,587. Leap seconds are not applied. This timescale is defined from 1970 AD to 2100 AD, and is unambiguous over that range." (I'm assuming we will also fix the Y2038 problem.)

8. UTC is still needed for input and output involving humans. For this purpose, suitable conversion library functions should be provided. For the most part, these functions already exist, except for the fact that they don't all handle leap seconds correctly. For converting current or near future times, the number of leap seconds since the Epoch, or planned, both of which can be derived from the GPS signal, is all that's needed. For dates in the past, the list of past leap seconds (available from

[4] The offset later ended up being ten seconds.

various time standards organizations) is needed. For dates too far in the future to know the leap seconds, some convention must be adopted. Likewise, a convention for handling UTC conversions in isolated systems is needed.

The fundamental problem with time in POSIX was not the “seconds since the epoch” uniform timescale, it was the fact that the equations for conversion between broken-down time and seconds since the epoch fell apart in the presence of leap seconds. So, the solution is simply to provide conversion functions that work with leap seconds, specifically a table giving all past (to the Epoch) and future leap seconds and when they happened or will soon happen.

Leap seconds must be announced at least eight weeks in advance, and are often announced 5 or 6 months in advance. There are no double leap seconds, although one can have more than one leap second in a given year.

In this table, no distinction between past and future leap seconds need be made, and the table can be as simple as a slowly growing list of “seconds since the epoch” values that are leap seconds. This list grows by one value every 1.5 years on average, so from 1970 to 2000 there are about 24 values; and in 2100 AD we would have a total of 87 values, call it 100 values.

Why does this work? It's a bit confusing to explain, but because uniform timescales like POSIX Time themselves have no leap seconds, one can identify (color in) the physical seconds that are handled as positive leap seconds in UTC. One can think of this in another way, that a leap second is nothing more than an arbitrary abrupt adjustment in the mapping from uniform (atomic) time to civil time (UTC), done merely to keep civil time more or less aligned with solar time (so noon always happens when the Sun is at its zenith, despite the gradual slowing of the Earth's rotation). Given this list of the colored-in physical seconds (leap seconds), one can convert in both directions without difficulty.

Acronyms: **AD** = Anno Domini, **ITU** = International Telecommunications, Union, **GMT** = Greenwich Mean Time, **GNSS** = Global Navigation Satellite System, **GPS** Global Positioning System (a form of GNSS), **NTP** = Network Time Protocol, **PASC** = Portable Applications Standards Committee, **POSIX** = Portable Operating System Interface (the UNIX is silent), **SI** = International System of Units (commonly known as the metric system), **TAI** = Temps Atomique International (French for International Atomic Time), **UTi** = Universal Time, **UTC** = Coordinated Universal Time (the abbreviation is from French), **WG** = Working Group