NAME

time2posix, **posix2time** - convert seconds since the Epoch

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

#include <time.h>

time_t
time2posix(*time_t t*);

time_t
posix2time(*time_t t*);

DESCRIPTION

IEEE Std 1003.1-1988 ("POSIX.1") requires the time_t value 536457599 to stand for 1986-12-31 23:59:59 UTC. This effectively implies that POSIX *time_t* values cannot include leap seconds and, therefore, that the system time must be adjusted as each leap occurs.

If the time package is configured with leap-second support enabled, however, no such adjustment is needed and *time_t* values continue to increase over leap events (as a true "seconds since..." value). This means that these values will differ from those required by POSIX by the net number of leap seconds inserted since the Epoch.

Typically this is not a problem as the type *time_t* is intended to be (mostly) opaque -- *time_t* values should only be obtained-from and passed-to functions such as time(3), localtime(3), mktime(3) and difftime(3). However, IEEE Std 1003.1-1988 ("POSIX.1") gives an arithmetic expression for directly computing a *time_t* value from a given date/time, and the same relationship is assumed by some (usually older) applications. Any programs creating/dissecting *time_t* values using such a relationship will typically not handle intervals over leap seconds correctly.

The **time2posix**() and **posix2time**() functions are provided to address this *time_t* mismatch by converting between local *time_t* values and their POSIX equivalents. This is done by accounting for the number of time-base changes that would have taken place on a POSIX system as leap seconds were inserted or deleted. These converted values can then be used in lieu of correcting the older applications, or when communicating with POSIX-compliant systems.

The **time2posix**() function is single-valued. That is, every local *time_t* corresponds to a single POSIX *time_t*. The **posix2time**() function is less well-behaved: for a positive leap second hit the result is not

unique, and for a negative leap second hit the corresponding POSIX *time_t* does not exist so an adjacent value is returned. Both of these are good indicators of the inferiority of the POSIX representation.

The following table summarizes the relationship between a time T and its conversion to, and back from, the POSIX representation over the leap second inserted at the end of June, 1993.

DATE	TIME	T	X=time2posix(T)	posix2time(X)
93/06/30	23:59:59	A+0	B+0	A+0
93/06/30	23:59:60	A+1	B+1	A+1 or A+2
93/07/01	00:00:00	A+2	B+1	A+1 or A+2
93/07/01	00:00:01	A+3	B+2	A+3

A leap second deletion would look like...

DATE	TIME	T	X=time2posix(T)	posix2time(X)
??/06/30	23:59:58	A+0	B+0	A+0
??/07/01	00:00:00	A+1	B+2	A+1
??/07/01	00:00:01	A+2	B+3	A+2

[Note: posix2time(B+1) => A+0 or A+1]

If leap-second support is not enabled, local *time_t* and POSIX *time_t* values are equivalent, and both **time2posix**() and **posix2time**() degenerate to the identity function.

SEE ALSO

difftime(3), localtime(3), mktime(3), time(3)