NAME

time - get time of day

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

#include <time.h>

time_t
time(*time_t *tloc*);

DESCRIPTION

The **time**() function returns the value of time in seconds since 0 hours, 0 minutes, 0 seconds, January 1, 1970, Coordinated Universal Time (UTC). If an error occurs, **time**() returns the value (*time_t*)-1.

The return value is also stored in **tloc*, provided that *tloc* is non-null.

ERRORS

The **time**() function may fail for any of the reasons described in clock_gettime(2).

SEE ALSO

clock_gettime(2), gettimeofday(2), ctime(3)

STANDARDS

The **time** function conforms to IEEE Std 1003.1-2008 ("POSIX.1").

HISTORY

The **time**() system call first appeared in Version 1 AT&T UNIX. Through the Version 3 AT&T UNIX, it returned 60 Hz ticks since an epoch that changed occasionally, because it was a 32-bit value that overflowed in a little over 2 years.

In Version 4 AT&T UNIX the granularity of the return value was reduced to whole seconds, delaying the aforementioned overflow until 2038.

Version 7 AT&T UNIX introduced the **ftime**() system call, which returned time at a millisecond level, though retained the **gtime**() system call (exposed as **time**() in userland). **time**() could have been implemented as a wrapper around **ftime**(), but that wasn’t done.

4.1cBSD implemented a higher-precision time function **gettimeofday**() to replace **ftime**() and

reimplemented **time**() in terms of that.

Since FreeBSD 9 the implementation of **time**() uses **clock_gettime**(*CLOCK_SECOND*) instead of **gettimeofday**() for performance reasons.

BUGS

Neither ISO/IEC 9899:1999 ("ISO C99") nor IEEE Std 1003.1-2001 ("POSIX.1") requires **time**() to set *errno* on failure; thus, it is impossible for an application to distinguish the valid time value -1 (representing the last UTC second of 1969) from the error return value.

Systems conforming to earlier versions of the C and POSIX standards (including older versions of FreeBSD) did not set **tloc* in the error case.