

NAME

time – get time in seconds

LIBRARY

Standard C library (*libc*, *-lc*)

SYNOPSIS

```
#include <time.h>
```

```
time_t time(time_t * _Nullable tloc);
```

DESCRIPTION

time() returns the time as the number of seconds since the Epoch, 1970-01-01 00:00:00 +0000 (UTC).

If *tloc* is non-NULL, the return value is also stored in the memory pointed to by *tloc*.

RETURN VALUE

On success, the value of time in seconds since the Epoch is returned. On error, $((time_t) - 1)$ is returned, and *errno* is set to indicate the error.

ERRORS**EFAULT**

tloc points outside your accessible address space (but see BUGS).

On systems where the C library **time()** wrapper function invokes an implementation provided by the **vdso(7)** (so that there is no trap into the kernel), an invalid address may instead trigger a **SIGSEGV** signal.

STANDARDS

SVr4, 4.3BSD, C99, POSIX.1-2001. POSIX does not specify any error conditions.

NOTES

POSIX.1 defines *seconds since the Epoch* using a formula that approximates the number of seconds between a specified time and the Epoch. This formula takes account of the facts that all years that are evenly divisible by 4 are leap years, but years that are evenly divisible by 100 are not leap years unless they are also evenly divisible by 400, in which case they are leap years. This value is not the same as the actual number of seconds between the time and the Epoch, because of leap seconds and because system clocks are not required to be synchronized to a standard reference. The intention is that the interpretation of seconds since the Epoch values be consistent; see POSIX.1-2008 Rationale A.4.15 for further rationale.

On Linux, a call to **time()** with *tloc* specified as NULL cannot fail with the error **EOverflow**, even on ABIs where *time_t* is a signed 32-bit integer and the clock reaches or exceeds 2³¹ seconds (2038-01-19 03:14:08 UTC, ignoring leap seconds). (POSIX.1 permits, but does not require, the **EOverflow** error in the case where the seconds since the Epoch will not fit in *time_t*.) Instead, the behavior on Linux is undefined when the system time is out of the *time_t* range. Applications intended to run after 2038 should use ABIs with *time_t* wider than 32 bits.

BUGS

Error returns from this system call are indistinguishable from successful reports that the time is a few seconds *before* the Epoch, so the C library wrapper function never sets *errno* as a result of this call.

The *tloc* argument is obsolescent and should always be NULL in new code. When *tloc* is NULL, the call cannot fail.

C library/kernel differences

On some architectures, an implementation of **time()** is provided in the **vdso(7)**.

SEE ALSO

date(1), **gettimeofday(2)**, **ctime(3)**, **ftime(3)**, **time(7)**, **vdso(7)**