NAME

nanosleep - high resolution sleep

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

#include <time.h>

int
clock_nanosleep(*clockid_t clock_id, int flags, const struct timespec *rqtp, struct timespec *rmtp*);

int
nanosleep(*const struct timespec *rqtp, struct timespec *rmtp*);

DESCRIPTION

If the TIMER_ABSTIME flag is not set in the *flags* argument, then **clock_nanosleep**() suspends execution of the calling thread until either the time interval specified by the *rqtp* argument has elapsed, or a signal is delivered to the calling process and its action is to invoke a signal-catching function or to terminate the process. The clock used to measure the time is specified by the *clock_id* argument.

If the TIMER_ABSTIME flag is set in the *flags* argument, then **clock_nanosleep**() suspends execution of the calling thread until either the value of the clock specified by the *clock_id* argument reaches the absolute time specified by the *rqtp* argument, or a signal is delivered to the calling process and its action is to invoke a signal-catching function or to terminate the process. If, at the time of the call, the time value specified by *rqtp* is less than or equal to the time value of the specified clock, then **clock_nanosleep**() returns immediately and the calling thread is not suspended.

The suspension time may be longer than requested due to the scheduling of other activity by the system. It is also subject to the allowed time interval deviation specified by the *kern.timecounter.alloweddeviation* sysctl(8) variable. An unmasked signal will terminate the sleep early, regardless of the SA_RESTART value on the interrupting signal. The *rqtp* and *rmtp* arguments can point to the same object.

The following *clock_id* values are supported:

CLOCK_MONOTONIC
CLOCK_MONOTONIC_FAST
CLOCK_MONOTONIC_PRECISE
CLOCK_REALTIME

CLOCK_REALTIME_FAST
CLOCK_REALTIME_PRECISE
CLOCK_SECOND
CLOCK_UPTIME
CLOCK_UPTIME_FAST
CLOCK_UPTIME_PRECISE

The **nanosleep**() function behaves like **clock_nanosleep**() with a *clock_id* argument of CLOCK_REALTIME and without the TIMER_ABSTIME flag in the *flags* argument.

RETURN VALUES

These functions return zero when the requested time has elapsed.

If these functions return for any other reason, then **clock_nanosleep**() will directly return the error number, and **nanosleep**() will return -1 with the global variable *errno* set to indicate the error. If a relative sleep is interrupted by a signal and *rmtp* is non-NULL, the timespec structure it references is updated to contain the unslept amount (the request time minus the time actually slept).

ERRORS

These functions can fail with the following errors.

[EFAULT] Either *rqtp* or *rmtp* points to memory that is not a valid part of the process address space.

[EINTR] The function was interrupted by the delivery of a signal.

[EINVAL] The *rqtp* argument specified a nanosecond value less than zero or greater than or equal to 1000 million.

[EINVAL] The *flags* argument contained an invalid flag.

[EINVAL] The *clock_id* argument was CLOCK_THREAD_CPUTIME_ID or an unrecognized value.

[ENOTSUP] The *clock_id* argument was valid but not supported by this implementation of **clock_nanosleep**().

SEE ALSO

clock_gettime(2), sigaction(2), sleep(3)

STANDARDS

These functions conform to IEEE Std 1003.1-2008 ("POSIX.1").

HISTORY

The predecessor of this system call, **sleep**(), appeared in Version 3 AT&T UNIX, but was removed when alarm(3) was introduced into Version 7 AT&T UNIX. The **nanosleep**() system call has been available since NetBSD 1.3 and was ported to OpenBSD 2.1 and FreeBSD 3.0.