NAME

timer_create - create a per-process timer (REALTIME)

LIBRARY

POSIX Real-time Library (librt, -lrt)

SYNOPSIS

#include <time.h>
#include <signal.h>

int
timer_create(*clockid_t clockid*, *struct sigevent *restrict evp*, *timer_t *restrict timerid*);

DESCRIPTION

The **timer_create**() system call creates a per-process timer using the specified clock, *clock_id*, as the timing base. The **timer_create**() system call returns, in the location referenced by *timerid*, a timer ID of type *timer_t* used to identify the timer in timer requests. This timer ID is unique within the calling process until the timer is deleted. The particular clock, *clock_id*, is defined in <*time.h*>. The timer whose ID is returned is in a disarmed state upon return from **timer_create**().

The *evp* argument, if non-NULL, points to a *sigevent* structure. This structure, allocated by the application, defines the asynchronous notification to occur when the timer expires.

If *evp->sigev_notify* is SIGEV_SIGNO or SIGEV_THREAD_ID, the signal specified in *evp->sigev_signo* will be sent to the calling process (SIGEV_SIGNO) or to the thread whose LWP ID is *evp->sigev_notify_thread_id* (SIGEV_THREAD_ID). The information for the queued signal will include:

Member	**Value**
si_code	SI_TIMER
si_value	the value stored in *evp->sigev_value*
si_timerid	timer ID
si_overrun	timer overrun count
si_errno	If timer overrun is {DELAYTIMER_MAX}, an error code defined in <*errno.h*>

If the *evp* argument is NULL, the effect is as if the *evp* argument pointed to a *sigevent* structure with the *sigev_notify* member having the value SIGEV_SIGNAL, the *sigev_signo* having a default signal number (SIGALRM), and the *sigev_value* member having the value of the timer ID.

This implementation supports a *clock_id* of CLOCK_REALTIME or CLOCK_MONOTONIC.

If *evp->sigev_notify* is SIGEV_THREAD and *sev->sigev_notify_attributes* is not NULL, if the attribute pointed to by *sev->sigev_notify_attributes* has a thread stack address specified by a call to **pthread_attr_setstack**() or **pthread_attr_setstackaddr**(), the results are unspecified if the signal is generated more than once.

RETURN VALUES

If the call succeeds, **timer_create**() returns zero and updates the location referenced by *timerid* to a *timer_t*, which can be passed to the per-process timer calls. If an error occurs, the system call returns a value of -1 and the global variable *errno* is set to indicate the error. The value of *timerid* is undefined if an error occurs.

ERRORS

The **timer_create**() system call will fail if:

[EAGAIN] The calling process has already created all of the timers it is allowed by this implementation.

[EINVAL] The specified clock ID is not supported.

[EINVAL] The specified asynchronous notification method is not supported.

[EFAULT] Any arguments point outside the allocated address space or there is a memory protection fault.

SEE ALSO

clock_getres(2), timer_delete(2), timer_getoverrun(2), sigevent(3), siginfo(3)

STANDARDS

The **timer_create**() system call conforms to IEEE Std 1003.1-2004 ("POSIX.1").

HISTORY

Support for POSIX per-process timer first appeared in FreeBSD 7.0.