NAME

mtx_timedlock - attempt acquire a mutex with timeout

SYNOPSIS

library "threads"
#include <threads.h>

int
mtx_timedlock(*mtx_t *mtx, const struct timespec *ts*);

DESCRIPTION

The function **mtx_timedlock**() will attempt to acquire a mutex *mtx*, if the mutex is already acquired by an other thread the function will wait for the amount specified by *ts*. The timeout *ts* is absolute and derived from CLOCK_REALTIME.

If the mutex is already being held by the current thread and locking recursion was not specified to mtx_init(3) the thread will be in a deadlock. If recursion is enabled the thread will acquire an additional instance of the mutex. In this case to unlock the mutex mtx_unlock(3) must be called an equal number of times as **mtx_timedlock**(), mtx_trylock(3) or mtx_lock(3) have been called from the same thread on the same mutex, in order to unlock the mutex for other threads.

RETURN VALUES

Upon success **mtx_timedlock**() will return *thrd_success* if the mutex has been acauired, if the mutex could not be acquired within the timeout *thrd_timedout* will be returned. In case of error *thrd_error* will be returned.

SEE ALSO

mtx_init(3) mtx_destroy(3) mtx_unlock(3) mtx_trylock(3) mtx_lock(3)

HISTORY

The **mtx_timedlock**() function first appeared in the C11 standard ISO/IEC 9899:2011.

AUTHORS

Jan Adelsbach <jan@jadelsbach.de>