

NAME

mtx_trylock - attempt to acquire a mutex

SYNOPSIS

library "threads"

#include <threads.h>

int

mtx_trylock(*mtx_t* **mtx*);

DESCRIPTION

The function **mtx_trylock**() will attempt to acquire a mutex *mtx* if it has not been acquired already by an other thread.

If the mutex is already held by the current thread and locking recursion was specified in **mtx_init**(3) an additional instance of the mutex will be acquired in the current thread. In this case to unlock the mutex **mtx_unlock**(3) must be called an equal number of times as **mtx_trylock**(), **mtx_lock**(3) or **mtx_timedlock**(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_trylock**() will return *thrd_success* if the mutex has been acquired or *thrd_busy* if the mutex is being held by an other thread. In case of error *thrd_error* will be returned.

SEE ALSO

mtx_init(3) **mtx_destroy**(3) **mtx_unlock**(3) **mtx_lock**(3) **mtx_timedlock**(3)

HISTORY

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

AUTHORS

Jan Adelsbach <jan@jadelsbach.de>