NAME

pthread_rwlock_wrlock, **pthread_rwlock_trywrlock** - acquire a read/write lock for writing

LIBRARY

POSIX Threads Library (libpthread, -lpthread)

SYNOPSIS

#include <pthread.h>

int
pthread_rwlock_wrlock(*pthread_rwlock_t *lock*);

int
pthread_rwlock_trywrlock(*pthread_rwlock_t *lock*);

DESCRIPTION

The **pthread_rwlock_wrlock**() function blocks until a write lock can be acquired against *lock*. The **pthread_rwlock_trywrlock**() function performs the same action, but does not block if the lock cannot be immediately obtained.

The results are undefined if the calling thread already holds the lock at the time the call is made.

IMPLEMENTATION NOTES

To prevent writer starvation, writers are favored over readers.

RETURN VALUES

If successful, the **pthread_rwlock_wrlock**() and **pthread_rwlock_trywrlock**() functions will return zero. Otherwise an error number will be returned to indicate the error.

ERRORS

The **pthread_rwlock_trywrlock**() function will fail if:

[EBUSY] The calling thread is not able to acquire the lock without blocking.

The **pthread_rwlock_wrlock**() and **pthread_rwlock_trywrlock**() functions may fail if:

[EDEADLK] The calling thread already owns the read/write lock (for reading or writing).

[EINVAL] The value specified by *lock* is invalid.

[ENOMEM] Insufficient memory exists to initialize the lock (applies to statically initialized locks only).

SEE ALSO

pthread_rwlock_init(3), pthread_rwlock_rdlock(3), pthread_rwlock_tryrdlock(3), pthread_rwlock_unlock(3)

STANDARDS

The **pthread_rwlock_wrlock**() and **pthread_rwlock_trywrlock**() functions are expected to conform to Version 2 of the Single UNIX Specification ("SUSv2").

HISTORY

The **pthread_rwlock_wrlock**() function first appeared in FreeBSD 3.0.