NAME

pthread_spin_lock, **pthread_spin_trylock**, **pthread_spin_unlock** **-** lock or unlock a spin lock

LIBRARY

POSIX Threads Library (libpthread, -lpthread)

SYNOPSIS

#include <pthread.h>

int
pthread_spin_lock(*pthread_spinlock_t *lock*);

int
pthread_spin_trylock(*pthread_spinlock_t *lock*);

int
pthread_spin_unlock(*pthread_spinlock_t *lock*);

DESCRIPTION

The **pthread_spin_lock**() function will acquire *lock* if it is not currently owned by another thread. If the lock cannot be acquired immediately, it will spin attempting to acquire the lock (it will not sleep) until it becomes available.

The **pthread_spin_trylock**() function is the same as **pthread_spin_lock**() except that if it cannot acquire *lock* immediately it will return with an error.

The **pthread_spin_unlock**() function will release *lock*, which must have been previously locked by a call to **pthread_spin_lock**() or **pthread_spin_trylock**().

RETURN VALUES

If successful, all these functions will return zero. Otherwise, an error number will be returned to indicate the error.

None of these functions will return EINTR.

ERRORS

The **pthread_spin_lock**(), **pthread_spin_trylock**() and **pthread_spin_unlock**() functions will fail if:

[EINVAL] The value specified by *lock* is invalid or is not initialized.

The **pthread_spin_lock**() function may fail if:

[EDEADLK]	The calling thread already owns the lock.

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

[EBUSY]	Another thread currently holds *lock*.

The **pthread_spin_unlock**() function may fail if:

[EPERM]	The calling thread does not own *lock*.

SEE ALSO

pthread_spin_destroy(3), pthread_spin_init(3)

HISTORY

The **pthread_spin_lock**(), **pthread_spin_trylock**() and **pthread_spin_unlock**() functions first appeared in N:M Threading Library (libkse, -lkse) in FreeBSD 5.2, and in 1:1 Threading Library (libthr, -lthr) in FreeBSD 5.3.

BUGS

The implementation of **pthread_spin_lock**(), **pthread_spin_trylock**() and **pthread_spin_unlock**() is expected to conform to IEEE Std 1003.2 ("POSIX.2").