Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'pthread_rwlock_trywrlock.3p' command

$ man pthread_rwlock_trywrlock.3p

PTHREAD_RWLOCK_TRYWRLOCK(3PPOSIX Programmer's ManuPTHREAD_RWLOCK_TRYWRLOCK(3P)

PROLOG

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

NAME

pthread_rwlock_trywrlock, pthread_rwlock_wrlock ? lock a read-write lock object for writing

SYNOPSIS

```
#include <pthread.h>
int pthread_rwlock_trywrlock(pthread_rwlock_t *rwlock);
int pthread_rwlock_wrlock(pthread_rwlock_t *rwlock);
```

DESCRIPTION

The pthread_rwlock_trywrlock() function shall apply a write lock like the pthread_rwlock_wrlock() function, with the exception that the func? tion shall fail if any thread currently holds rwlock (for reading or writing).

The pthread_rwlock_wrlock() function shall apply a write lock to the read-write lock referenced by rwlock. The calling thread shall acquire the write lock if no thread (reader or writer) holds the read-write lock rwlock. Otherwise, if another thread holds the read-write lock rwlock, the calling thread shall block until it can acquire the lock.

If a deadlock condition occurs or the calling thread already owns the read-write lock for writing or reading, the call shall either deadlock or return [EDEADLK].

Results are undefined if any of these functions are called with an uninitialized read-write lock.

If a signal is delivered to a thread waiting for a read-write lock for writing, upon return from the signal handler the thread resumes waiting for the read-write lock for writing as if it was not interrupted.

RETURN VALUE

The pthread_rwlock_trywrlock() function shall return zero if the lock for writing on the read-write lock object referenced by rwlock is ac? quired. Otherwise, an error number shall be returned to indicate the error.

If successful, the pthread_rwlock_wrlock() function shall return zero; otherwise, an error number shall be returned to indicate the error.

ERRORS

The pthread_rwlock_trywrlock() function shall fail if:

EBUSY The read-write lock could not be acquired for writing because it was already locked for reading or writing.

The pthread_rwlock_wrlock() function may fail if:

EDEADLK

A deadlock condition was detected or the current thread already owns the read-write lock for writing or reading.

These functions shall not return an error code of [EINTR].

The following sections are informative.

EXAMPLES

None.

APPLICATION USAGE

Applications using these functions may be subject to priority inver? sion, as discussed in the Base Definitions volume of POSIX.1?2017, Sec? tion 3.291, Priority Inversion.

RATIONALE

If an implementation detects that the value specified by the rwlock ar?

gument to pthread_rwlock_trywrlock() or pthread_rwlock_wrlock() does not refer to an initialized read-write lock object, it is recommended that the function should fail and report an [EINVAL] error.

FUTURE DIRECTIONS

None.

SEE ALSO

pthread_rwlock_destroy(), pthread_rwlock_rdlock(), pthread_rwlock_time? drdlock(), pthread_rwlock_timedwrlock(), pthread_rwlock_unlock()

The Base Definitions volume of POSIX.1?2017, Section 3.291, Priority Inversion, Section 4.12, Memory Synchronization, <pthread.h>

COPYRIGHT