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_timedrdlock.3p' command

$ man pthread_rwlock_timedrdlock.3p

PTHREAD_RWLOCK_TIMEDRDLOCK(POSIX Programmer's MaPTHREAD_RWLOCK_TIMEDRDLOCK(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_timedrdlock ? lock a read-write lock for reading

SYNOPSIS

```
#include <pthread.h>
#include <time.h>
int pthread_rwlock_timedrdlock(pthread_rwlock_t *restrict rwlock,
    const struct timespec *restrict abstime);
```

DESCRIPTION

The pthread_rwlock_timedrdlock() function shall apply a read lock to the read-write lock referenced by rwlock as in the pthread_rwlock_rd? lock() function. However, if the lock cannot be acquired without wait? ing for other threads to unlock the lock, this wait shall be terminated when the specified timeout expires. The timeout shall expire when the absolute time specified by abstime passes, as measured by the clock on which timeouts are based (that is, when the value of that clock equals or exceeds abstime), or if the absolute time specified by abstime has already been passed at the time of the call.

The timeout shall be based on the CLOCK_REALTIME clock. The resolution of the timeout shall be the resolution of the CLOCK_REALTIME clock. The timespec data type is defined in the <time.h> header. Under no circum? stances shall the function fail with a timeout if the lock can be ac? quired immediately. The validity of the abstime parameter need not be checked if the lock can be immediately acquired.

If a signal that causes a signal handler to be executed is delivered to a thread blocked on a read-write lock via a call to pthread_rwlock_timedrdlock(), upon return from the signal handler the thread shall resume waiting for the lock as if it was not interrupted.

The calling thread may deadlock if at the time the call is made it holds a write lock on rwlock. The results are undefined if this func? tion is called with an uninitialized read-write lock.

RETURN VALUE

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

ERRORS

The pthread_rwlock_timedrdlock() function shall fail if:

ETIMEDOUT

The lock could not be acquired before the specified timeout ex? pired.

The pthread_rwlock_timedrdlock() function may fail if:

EAGAIN The read lock could not be acquired because the maximum number of read locks for lock would be exceeded.

EDEADLK

A deadlock condition was detected or the calling thread already holds a write lock on rwlock.

EINVAL The abstime nanosecond value is less than zero or greater than or equal to 1000 million.

This function shall not return an error code of [EINTR].

The following sections are informative.

EXAMPLES

None.

APPLICATION USAGE

Applications using this function may be subject to priority inversion, as discussed in the Base Definitions volume of POSIX.1?2017, Section 3.291, Priority Inversion.

RATIONALE

If an implementation detects that the value specified by the rwlock ar? gument to pthread_rwlock_timedrdlock() 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_timedwrlock(), pthread_rwlock_trywrlock(), pthread_rwlock_unlock()

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

COPYRIGHT