

**NAME**

**pthread\_rwlock\_timedrdlock** - acquire a read-write lock for reading or give up after a specified period

**LIBRARY**

POSIX Threads Library (libpthread, -lpthread)

**SYNOPSIS**

```
#include <pthread.h>
```

*int*

```
pthread_rwlock_timedrdlock(pthread_rwlock_t *restrict rwlock,  
    const struct timespec *restrict abs_timeout);
```

**DESCRIPTION**

This function acquires a read lock on the read-write lock *rwlock*. However, if the lock cannot be acquired without waiting for another thread to unlock the lock, this wait shall be terminated when *abs\_timeout* expires.

A thread may hold multiple concurrent read locks. The `pthread_rwlock_unlock(3)` function must be called once for each lock acquired.

If the thread should be interrupted by a signal, the `pthread_rwlock_timedrdlock()` function will be automatically restarted after the thread returns from the signal handler.

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 function is called with an uninitialized read-write lock.

**IMPLEMENTATION NOTES**

To prevent writer starvation, writers are favored over readers.

**RETURN VALUES**

If successful, the `pthread_rwlock_timedrdlock()` function will return zero. Otherwise, an error number will be returned to indicate the error.

This function shall not return an error code of EINTR.

**ERRORS**

The `pthread_rwlock_timedrdlock()` function will fail if:

[ETIMEDOUT]      The lock could not be acquired before the specified timeout expired.

The **pthread\_rwlock\_timedrdlock()** function may fail if:

- |           |   |
|-----------|---|
| [EAGAIN]  | The read lock could not be acquired because the maximum number of read locks for <i>rwlock</i> would be exceeded.   |
| [EDEADLK] | The calling thread already holds a write lock on <i>rwlock</i> .  |
| [EINVAL]  | The value specified by <i>rwlock</i> does not refer to an initialized read-write lock object, or the <i>abs_timeout</i> nanosecond value is less than zero or greater than or equal to 1 billion. |

### SEE ALSO

pthread\_rwlock\_init(3), pthread\_rwlock\_timedwrlock(3), pthread\_rwlock\_unlock(3)

### STANDARDS

The **pthread\_rwlock\_timedrdlock()** function is expected to conform to ISO/IEC 9945-1:1996 ("POSIX.1").

### HISTORY

The **pthread\_rwlock\_timedrdlock()** function first appeared in FreeBSD 5.2.