



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

`$ man pthread_rwlock_timedwrlock.3p`

PTHREAD_RWLOCK_TIMEDWRLOCK(POSIX Programmer's Manual: PTHREAD_RWLOCK_TIMEDWRLOCK(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_timedwrlock` ? lock a read-write lock for writing

SYNOPSIS

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

DESCRIPTION

The `pthread_rwlock_timedwrlock()` function shall apply a write lock to the read-write lock referenced by `rwlock` as in the `pthread_rwlock_wrlock()` function. However, if the lock cannot be acquired without waiting 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 circumstances shall the function fail with a timeout if the lock can be acquired 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_timedwrlock(), 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 the read-write lock. The results are undefined if this function is called with an uninitialized read-write lock.

RETURN VALUE

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

ERRORS

The pthread_rwlock_timedwrlock() function shall fail if:

ETIMEDOUT

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

The pthread_rwlock_timedwrlock() function may fail if:

EDEADLK

A deadlock condition was detected or the calling thread already holds the 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` argument to `pthread_rwlock_timedwrlock()` 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_timedrdlock()`, `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

Portions of this text are reprinted and reproduced in electronic form from IEEE Std 1003.1-2017, Standard for Information Technology -- Portable Operating System Interface (POSIX), The Open Group Base Specifications Issue 7, 2018 Edition, Copyright (C) 2018 by the Institute of Electrical and Electronics Engineers, Inc and The Open Group. In the event of any discrepancy between this version and the original IEEE and The Open Group Standard, the original IEEE and The Open Group Standard is the referee document. The original Standard can be obtained online at <http://www.opengroup.org/unix/online.html>.

Any typographical or formatting errors that appear in this page are most likely to have been introduced during the conversion of the source files to man page format. To report such errors, see https://www.kernel.org/doc/man-pages/reporting_bugs.html.