



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

\$ man pthread_rwlock_unlock.3p

PTHREAD_RWLOCK_UNLOCK(3P) POSIX Programmer's Manual PTHREAD_RWLOCK_UNLOCK(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_unlock ? unlock a read-write lock object

SYNOPSIS

```
#include <pthread.h>

int pthread_rwlock_unlock(pthread_rwlock_t *rwlock);
```

DESCRIPTION

The pthread_rwlock_unlock() function shall release a lock held on the read-write lock object referenced by rwlock. Results are undefined if the read-write lock rwlock is not held by the calling thread.

If this function is called to release a read lock from the read-write lock object and there are other read locks currently held on this read-write lock object, the read-write lock object remains in the read locked state. If this function releases the last read lock for this read-write lock object, the read-write lock object shall be put in the unlocked state with no owners.

If this function is called to release a write lock for this read-write lock object, the read-write lock object shall be put in the unlocked

state.

If there are threads blocked on the lock when it becomes available, the scheduling policy shall determine which thread(s) shall acquire the lock. If the Thread Execution Scheduling option is supported, when threads executing with the scheduling policies `SCHED_FIFO`, `SCHED_RR`, or `SCHED_SPORADIC` are waiting on the lock, they shall acquire the lock in priority order when the lock becomes available. For equal priority threads, write locks shall take precedence over read locks. If the Thread Execution Scheduling option is not supported, it is implementation-defined whether write locks take precedence over read locks. Results are undefined if this function is called with an uninitialized read-write lock.

RETURN VALUE

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

ERRORS

The `pthread_rwlock_unlock()` function shall not return an error code of `[EINTR]`.

The following sections are informative.

EXAMPLES

None.

APPLICATION USAGE

None.

RATIONALE

If an implementation detects that the value specified by the `rwlock_argument` to `pthread_rwlock_unlock()` does not refer to an initialized read-write lock object, it is recommended that the function should fail and report an `[EINVAL]` error.

If an implementation detects that the value specified by the `rwlock_argument` to `pthread_rwlock_unlock()` refers to a read-write lock object for which the current thread does not hold a lock, it is recommended that the function should fail and report an `[EPERM]` error.

FUTURE DIRECTIONS

None.

SEE ALSO

`pthread_rwlock_destroy()`, `pthread_rwlock_rdlock()`, `pthread_rwlock_time?`
`rdlock()`, `pthread_rwlock_timedwrlock()`, `pthread_rwlock_trywrlock()`

The Base Definitions volume of POSIX.1-2017, Section 4.12, Memory Synchronization, `<pthread.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.

IEEE/The Open Group 2017 PTHREAD_RWLOCK_UNLOCK(3P)