



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

\$ man pthread_mutex_consistent.3p

PTHREAD_MUTEX_CONSISTENT(3PPOSIX Programmer's ManuPTHREAD_MUTEX_CONSISTENT(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_mutex_consistent ? mark state protected by robust mutex as consistent

SYNOPSIS

```
#include <pthread.h>

int pthread_mutex_consistent(pthread_mutex_t *mutex);
```

DESCRIPTION

If `mutex` is a robust mutex in an inconsistent state, the `pthread_mutex_consistent()` function can be used to mark the state protected by the mutex referenced by `mutex` as consistent again.

If an owner of a robust mutex terminates while holding the mutex, the mutex becomes inconsistent and the next thread that acquires the mutex lock shall be notified of the state by the return value `[EOWNERDEAD]`. In this case, the mutex does not become normally usable again until the state is marked consistent.

If the thread which acquired the mutex lock with the return value `[EOWNERDEAD]` terminates before calling either `pthread_mutex_consistent()`

tent() or pthread_mutex_unlock(), the next thread that acquires the mutex lock shall be notified about the state of the mutex by the return value [EOWNERDEAD].

The behavior is undefined if the value specified by the mutex argument to pthread_mutex_consistent() does not refer to an initialized mutex.

RETURN VALUE

Upon successful completion, the pthread_mutex_consistent() function shall return zero. Otherwise, an error value shall be returned to indicate the error.

ERRORS

The pthread_mutex_consistent() function shall fail if:

EINVAL The mutex object referenced by mutex is not robust or does not protect an inconsistent state.

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

The following sections are informative.

EXAMPLES

None.

APPLICATION USAGE

The pthread_mutex_consistent() function is only responsible for notifying the implementation that the state protected by the mutex has been recovered and that normal operations with the mutex can be resumed. It is the responsibility of the application to recover the state so it can be reused. If the application is not able to perform the recovery, it can notify the implementation that the situation is unrecoverable by a call to pthread_mutex_unlock() without a prior call to pthread_mutex_consistent(), in which case subsequent threads that attempt to lock the mutex will fail to acquire the lock and be returned [ENOTRECOVERABLE].

RATIONALE

If an implementation detects that the value specified by the mutex argument to pthread_mutex_consistent() does not refer to an initialized mutex, it is recommended that the function should fail and report an [EINVAL] error.

FUTURE DIRECTIONS

None.

SEE ALSO

`pthread_mutex_lock()`, `pthread_mutexattr_getrobust()`

The Base Definitions volume of POSIX.1-2017, `<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_MUTEX_CONSISTENT(3P)