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_mutex_getprioceiling.3p' command

$ man pthread_mutex_getprioceiling.3p

PTHREAD_MUTEX_GETPRIOCEILINPOSIX Programmer's PTHREAD_MUTEX_GETPRIOCEILING(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_getprioceiling, pthread_mutex_setprioceiling ? get and set the priority ceiling of a mutex (REALTIME THREADS)

SYNOPSIS

```
#include <pthread.h>
int pthread_mutex_getprioceiling(const pthread_mutex_t *restrict mutex,
    int *restrict prioceiling);
int pthread_mutex_setprioceiling(pthread_mutex_t *restrict mutex,
    int prioceiling, int *restrict old_ceiling);
```

DESCRIPTION

The pthread_mutex_getprioceiling() function shall return the current priority ceiling of the mutex.

The pthread_mutex_setprioceiling() function shall attempt to lock the mutex as if by a call to pthread_mutex_lock(), except that the process of locking the mutex need not adhere to the priority protect protocol. On acquiring the mutex it shall change the mutex's priority ceiling and then release the mutex as if by a call to pthread_mutex_unlock(). When

the change is successful, the previous value of the priority ceiling shall be returned in old_ceiling.

If the pthread_mutex_setprioceiling() function fails, the mutex prior? ity ceiling shall not be changed.

RETURN VALUE

If successful, the pthread_mutex_getprioceiling() and pthread_mu? tex_setprioceiling() functions shall return zero; otherwise, an error number shall be returned to indicate the error.

ERRORS

These functions shall fail if:

EINVAL The protocol attribute of mutex is PTHREAD_PRIO_NONE.

EPERM The implementation requires appropriate privileges to perform the operation and the caller does not have appropriate privi? leges.

The pthread_mutex_setprioceiling() function shall fail if:

EAGAIN The mutex could not be acquired because the maximum number of recursive locks for mutex has been exceeded.

EDEADLK

The mutex type is PTHREAD_MUTEX_ERRORCHECK and the current thread already owns the mutex.

EINVAL The mutex was created with the protocol attribute having the value PTHREAD_PRIO_PROTECT and the calling thread's priority is higher than the mutex's current priority ceiling, and the imple? mentation adheres to the priority protect protocol in the process of locking the mutex.

ENOTRECOVERABLE

The mutex is a robust mutex and the state protected by the mutex is not recoverable.

EOWNERDEAD

The mutex is a robust mutex and the process containing the pre? vious owning thread terminated while holding the mutex lock. The mutex lock shall be acquired by the calling thread and it is up to the new owner to make the state consistent (see pthread_mu?

tex_lock()).

The pthread_mutex_setprioceiling() function may fail if:

EDEADLK

A deadlock condition was detected.

EINVAL The priority requested by prioceiling is out of range.

EOWNERDEAD

The mutex is a robust mutex and the previous owning thread ter? minated while holding the mutex lock. The mutex lock shall be acquired by the calling thread and it is up to the new owner to make the state consistent (see pthread_mutex_lock()).

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

The following sections are informative.

EXAMPLES

None.

APPLICATION USAGE

None.

RATIONALE

None.

FUTURE DIRECTIONS

None.

SEE ALSO

pthread_mutex_destroy(), pthread_mutex_lock(), pthread_mutex_timed? lock()

The Base Definitions volume of POSIX.1?2017, <pthread.h>

COPYRIGHT

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.ker?nel.org/doc/man-pages/reporting_bugs.html .