



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_mutexattr_getprioceiling.3p' command

\$ man pthread_mutexattr_getprioceiling.3p

PTHREAD_MUTEXATTR_GETPRIOCEPOSIX(ProgrammePTHREAD_MUTEXATTR_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_mutexattr_getprioceiling, pthread_mutexattr_setprioceiling ?
get and set the prioceiling attribute of the mutex attributes object
(REALTIME THREADS)

SYNOPSIS

```
#include <pthread.h>

int pthread_mutexattr_getprioceiling(const pthread_mutexattr_t
    *restrict attr, int *restrict prioceiling);

int pthread_mutexattr_setprioceiling(pthread_mutexattr_t *attr,
    int prioceiling);
```

DESCRIPTION

The pthread_mutexattr_getprioceiling() and pthread_mutexattr_setprioceiling() functions, respectively, shall get and set the priority ceiling attribute of a mutex attributes object pointed to by attr which was previously created by the function pthread_mutexattr_init(). The prioceiling attribute contains the priority ceiling of initialized mutexes. The values of prioceiling are within the maximum range of pri?

ities defined by SCHED_FIFO.

The prioceiling attribute defines the priority ceiling of initialized mutexes, which is the minimum priority level at which the critical section guarded by the mutex is executed. In order to avoid priority inversion, the priority ceiling of the mutex shall be set to a priority higher than or equal to the highest priority of all the threads that may lock that mutex. The values of prioceiling are within the maximum range of priorities defined under the SCHED_FIFO scheduling policy.

The behavior is undefined if the value specified by the attr argument to pthread_mutexattr_getprioceiling() or pthread_mutexattr_setprioceiling() does not refer to an initialized mutex attributes object.

RETURN VALUE

Upon successful completion, the pthread_mutexattr_getprioceiling() and pthread_mutexattr_setprioceiling() functions shall return zero; otherwise, an error number shall be returned to indicate the error.

ERRORS

These functions may fail if:

EINVAL The value specified by prioceiling is invalid.

EPERM The caller does not have the privilege to perform the operation.

These functions 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 attr argument to pthread_mutexattr_getprioceiling() or pthread_mutexattr_setprioceiling() does not refer to an initialized mutex attributes object, it is recommended that the function should fail and report an [EINVAL] error.

FUTURE DIRECTIONS

None.

SEE ALSO

`pthread_cond_destroy()`, `pthread_create()`, `pthread_mutex_destroy()`

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_MUTEXATTR_GETPRIOCEILING(3P)