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_getprotocol.3p' command

$ man pthread_mutexattr_getprotocol.3p

PTHREAD_MUTEXATTR_GETPROTOCPOSIX)Programmer'sPTHREAD_MUTEXATTR_GETPROTOCOL(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_getprotocol, pthread_mutexattr_setprotocol ? get and set the protocol attribute of the mutex attributes object (REALTIME THREADS)

SYNOPSIS

```
#include <pthread.h>
int pthread_mutexattr_getprotocol(const pthread_mutexattr_t
    *restrict attr, int *restrict protocol);
int pthread_mutexattr_setprotocol(pthread_mutexattr_t *attr,
    int protocol);
```

DESCRIPTION

The pthread_mutexattr_getprotocol() and pthread_mutexattr_setprotocol() functions, respectively, shall get and set the protocol attribute of a mutex attributes object pointed to by attr which was previously created by the function pthread_mutexattr_init().

The protocol attribute defines the protocol to be followed in utilizing mutexes. The value of protocol may be one of:

PTHREAD_PRIO_INHERIT

PTHREAD_PRIO_NONE

PTHREAD_PRIO_PROTECT

which are defined in the <pthread.h> header. The default value of the attribute shall be PTHREAD_PRIO_NONE.

When a thread owns a mutex with the PTHREAD_PRIO_NONE protocol attribute, its priority and scheduling shall not be affected by its mutex ownership.

When a thread is blocking higher priority threads because of owning one or more robust mutexes with the PTHREAD_PRIO_INHERIT protocol attribute, it shall execute at the higher of its priority or the priority of the highest priority thread waiting on any of the robust mutexes owned by this thread and initialized with this protocol.

When a thread is blocking higher priority threads because of owning one or more non-robust mutexes with the PTHREAD_PRIO_INHERIT protocol attribute, it shall execute at the higher of its priority or the priority of the highest priority thread waiting on any of the non-robust mutexes owned by this thread and initialized with this protocol.

When a thread owns one or more robust mutexes initialized with the PTHREAD_PRIO_PROTECT protocol, it shall execute at the higher of its priority or the highest of the priority ceilings of all the robust mutexes owned by this thread and initialized with this attribute, regardless of whether other threads are blocked on any of these robust mutexes or not.

When a thread owns one or more non-robust mutexes initialized with the PTHREAD_PRIO_PROTECT protocol, it shall execute at the higher of its priority or the highest of the priority ceilings of all the non-robust mutexes owned by this thread and initialized with this attribute, regardless of whether other threads are blocked on any of these non-robust mutexes or not.

While a thread is holding a mutex which has been initialized with the PTHREAD_PRIO_INHERIT or PTHREAD_PRIO_PROTECT protocol attributes, it shall not be subject to being moved to the tail of the scheduling queue

at its priority in the event that its original priority is changed, such as by a call to sched_setparam(). Likewise, when a thread unlocks a mutex that has been initialized with the PTHREAD_PRIO_INHERIT or PTHREAD_PRIO_PROTECT protocol attributes, it shall not be subject to being moved to the tail of the scheduling queue at its priority in the event that its original priority is changed.

If a thread simultaneously owns several mutexes initialized with dif? ferent protocols, it shall execute at the highest of the priorities that it would have obtained by each of these protocols.

When a thread makes a call to pthread_mutex_lock(), the mutex was ini? tialized with the protocol attribute having the value PTHREAD_PRIO_IN? HERIT, when the calling thread is blocked because the mutex is owned by another thread, that owner thread shall inherit the priority level of the calling thread as long as it continues to own the mutex. The imple? mentation shall update its execution priority to the maximum of its as? signed priority and all its inherited priorities. Furthermore, if this owner thread itself becomes blocked on another mutex with the protocol attribute having the value PTHREAD_PRIO_INHERIT, the same priority in? heritance effect shall be propagated to this other owner thread, in a recursive manner.

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

RETURN VALUE

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

ERRORS

The pthread_mutexattr_setprotocol() function shall fail if:

ENOTSUP

The value specified by protocol is an unsupported value.

The pthread_mutexattr_getprotocol() and pthread_mutexattr_setprotocol() functions may fail if:

EINVAL The value specified by protocol 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 argu?
ment to pthread_mutexattr_getprotocol() or pthread_mutexattr_setproto?
col() 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