NAME

pthread_setschedparam, pthread_getschedparam − control thread scheduling parameters

SYNOPSIS

#include <pthread.h>

int pthread_setschedparam(pthread_t *target_thread***, int** *policy***, const struct sched_param ***param***);**

int pthread_getschedparam(pthread_t *target_thread***, int ***policy***, struct sched_param ***param***);**

DESCRIPTION

pthread_setschedparam sets the scheduling parameters for the thread *target_thread* as indicated by *policy* and *param*. *policy* can be either **SCHED_OTHER** (regular, non-realtime scheduling), **SCHED_RR** (realtime, round-robin) or **SCHED_FIFO** (realtime, first-in first-out). *param* specifies the scheduling priority for the two realtime policies. See **sched_setpolicy**(2) for more information on scheduling policies.

The realtime scheduling policies **SCHED_RR** and **SCHED_FIFO** are available only to processes with superuser privileges.

pthread_getschedparam retrieves the scheduling policy and scheduling parameters for the thread *target_thread* and store them in the locations pointed to by *policy* and *param*, respectively.

RETURN VALUE

pthread_setschedparam and **pthread_getschedparam** return 0 on success and a non-zero error code on error.

ERRORS

On error, **pthread_setschedparam** returns the following error codes:

EINVAL
policy is not one of **SCHED_OTHER**, **SCHED_RR**, **SCHED_FIFO**

EINVAL
the priority value specified by *param* is not valid for the specified policy

EPERM
the calling process does not have superuser permissions

ESRCH
the *target_thread* is invalid or has already terminated

EFAULT
param points outside the process memory space

On error, **pthread_getschedparam** returns the following error codes:

ESRCH
the *target_thread* is invalid or has already terminated

EFAULT
policy or *param* point outside the process memory space

AUTHOR

Xavier Leroy <Xavier.Leroy@inria.fr>

SEE ALSO

sched_setscheduler(2), **sched_getscheduler**(2), **sched_getparam**(2), **pthread_attr_setschedpolicy**(3), **pthread_attr_setschedparam**(3).