

**NAME**

**pthread\_setschedparam**, **pthread\_getschedparam** - thread scheduling parameter manipulation

**LIBRARY**

POSIX Threads Library (libpthread, -lpthread)

**SYNOPSIS**

```
#include <pthread.h>
```

*int*

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

*int*

```
pthread_getschedparam(pthread_t thread, int *restrict policy, struct sched_param *restrict param);
```

**DESCRIPTION**

The **pthread\_setschedparam**() and **pthread\_getschedparam**() functions set and get the scheduling parameters of individual threads. The scheduling policy for a thread can either be SCHED\_FIFO (first in, first out), SCHED\_RR (round-robin), or SCHED\_OTHER (timesharing). Valid thread priorities (accessed via *param->sched\_priority*) must be within the range returned by the `sched_get_priority_min(2)` and `sched_get_priority_max(2)` system calls.

**RETURN VALUES**

If successful, these functions return 0. Otherwise, an error number is returned to indicate the error.

**ERRORS**

The **pthread\_setschedparam**() function will fail if:

- |           |   |
|-----------|---|
| [EINVAL]  | Invalid value for <i>policy</i> .   |
| [ENOTSUP] | Invalid value for scheduling parameters.  |
| [EPERM]   | The calling thread does not have sufficient privilege to perform the operation. |
| [ESRCH]   | Non-existent thread <i>thread</i> .   |

The **pthread\_getschedparam**() function will fail if:

- |         |                                     |
|---------|-------------------------------------|
| [ESRCH] | Non-existent thread <i>thread</i> . |
|---------|-------------------------------------|

**SEE ALSO**

sched\_get\_priority\_max(2), sched\_get\_priority\_min(2)

**STANDARDS**

The **pthread\_setschedparam()** and **pthread\_getschedparam()** functions conform to Version 2 of the Single UNIX Specification ("SUSv2").