



Red Hat Enterprise Linux Release 9.2 Manual Pages on 'pthread_getschedparam.3p' command

\$ man pthread_getschedparam.3p

PTHREAD_GETSCHEDPARAM(3P) POSIX Programmer's Manual PTHREAD_GETSCHEDPARAM(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_getschedparam, pthread_setschedparam ? dynamic thread scheduling parameters access (REALTIME THREADS)

SYNOPSIS

```
#include <pthread.h>

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

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

DESCRIPTION

The `pthread_getschedparam()` and `pthread_setschedparam()` functions shall, respectively, get and set the scheduling policy and parameters of individual threads within a multi-threaded process to be retrieved and set. For `SCHED_FIFO` and `SCHED_RR`, the only required member of the `sched_param` structure is the priority `sched_priority`. For `SCHED_OTHER`, the affected scheduling parameters are implementation-defined.

The `pthread_getschedparam()` function shall retrieve the scheduling pol?

icy and scheduling parameters for the thread whose thread ID is given by thread and shall store those values in policy and param, respectively. The priority value returned from pthread_getschedparam() shall be the value specified by the most recent pthread_setschedparam(), pthread_setschedprio(), or pthread_create() call affecting the target thread. It shall not reflect any temporary adjustments to its priority as a result of any priority inheritance or ceiling functions. The pthread_setschedparam() function shall set the scheduling policy and associated scheduling parameters for the thread whose thread ID is given by thread to the policy and associated parameters provided in policy and param, respectively.

The policy parameter may have the value SCHED_OTHER, SCHED_FIFO, or SCHED_RR. The scheduling parameters for the SCHED_OTHER policy are implementation-defined. The SCHED_FIFO and SCHED_RR policies shall have a single scheduling parameter, priority.

If _POSIX_THREAD_SPORADIC_SERVER is defined, then the policy argument may have the value SCHED_SPORADIC, with the exception for the pthread_setschedparam() function that if the scheduling policy was not SCHED_SPORADIC at the time of the call, it is implementation-defined whether the function is supported; in other words, the implementation need not allow the application to dynamically change the scheduling policy to SCHED_SPORADIC. The sporadic server scheduling policy has the associated parameters sched_ss_low_priority, sched_ss_repl_period, sched_ss_init_budget, sched_priority, and sched_ss_max_repl. The specified sched_ss_repl_period shall be greater than or equal to the specified sched_ss_init_budget for the function to succeed; if it is not, then the function shall fail. The value of sched_ss_max_repl shall be within the inclusive range [1, {SS_REPL_MAX}] for the function to succeed; if not, the function shall fail. It is unspecified whether the sched_ss_repl_period and sched_ss_init_budget values are stored as provided by this function or are rounded to align with the resolution of the clock being used.

If the pthread_setschedparam() function fails, the scheduling parameter

ters shall not be changed for the target thread.

RETURN VALUE

If successful, the `pthread_getschedparam()` and `pthread_setschedparam()` functions shall return zero; otherwise, an error number shall be returned to indicate the error.

ERRORS

The `pthread_setschedparam()` function shall fail if:

ENOTSUP

An attempt was made to set the policy or scheduling parameters to an unsupported value.

ENOTSUP

An attempt was made to dynamically change the scheduling policy to `SCHED_SPORADIC`, and the implementation does not support this change.

The `pthread_setschedparam()` function may fail if:

EINVAL The value specified by policy or one of the scheduling parameters associated with the scheduling policy is invalid.

EPERM The caller does not have appropriate privileges to set either the scheduling parameters or the scheduling policy of the specified thread.

EPERM The implementation does not allow the application to modify one of the parameters to the value specified.

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 use of a thread ID after the end of its lifetime, it is recommended that the function should fail and report an `[ESRCH]` error.

FUTURE DIRECTIONS

None.

SEE ALSO

`pthread_setschedprio()`, `sched_getparam()`, `sched_getscheduler()`

The Base Definitions volume of POSIX.1-2017, `<pthread.h>`, `<sched.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_GETSCHEDPARAM(3P)