

**NAME**

**sched\_setparam, sched\_getparam** - set/get scheduling parameters

**LIBRARY**

Standard C Library (libc, -lc)

**SYNOPSIS**

**#include <sched.h>**

*int*

**sched\_setparam**(*pid\_t pid*, *const struct sched\_param \*param*);

*int*

**sched\_getparam**(*pid\_t pid*, *struct sched\_param \*param*);

**DESCRIPTION**

The **sched\_setparam()** system call sets the scheduling parameters of the process specified by *pid* to the values specified by the *sched\_param* structure pointed to by *param*. The value of the *sched\_priority* member in the *param* structure must be any integer within the inclusive priority range for the current scheduling policy of the process specified by *pid*. Higher numerical values for the priority represent higher priorities.

In this implementation, if the value of *pid* is negative the system call will fail.

If a process specified by *pid* exists and if the calling process has permission, the scheduling parameters are set for the process whose process ID is equal to *pid*.

If *pid* is zero, the scheduling parameters are set for the calling process.

In this implementation, the policy of when a process can affect the scheduling parameters of another process is specified in IEEE Std 1003.1b-1993 ("POSIX.1b") as a write-style operation.

The target process, whether it is running or not running, will resume execution after all other runnable processes of equal or greater priority have been scheduled to run.

If the priority of the process specified by the *pid* argument is set higher than that of the lowest priority running process and if the specified process is ready to run, the process specified by the *pid* argument will preempt a lowest priority running process. Similarly, if the process calling **sched\_setparam()** sets its own priority lower than that of one or more other nonempty process lists, then the process that is the head of the highest priority list will also preempt the calling process. Thus, in either case, the

originating process might not receive notification of the completion of the requested priority change until the higher priority process has executed.

In this implementation, when the current scheduling policy for the process specified by *pid* is normal timesharing (SCHED\_OTHER, aka SCHED\_NORMAL when not POSIX-source) or the idle policy (SCHED\_IDLE when not POSIX-source) then the behavior is as if the process had been running under SCHED\_RR with a priority lower than any actual realtime priority.

The **sched\_getparam()** system call will return the scheduling parameters of a process specified by *pid* in the *sched\_param* structure pointed to by *param*.

If a process specified by *pid* exists and if the calling process has permission, the scheduling parameters for the process whose process ID is equal to *pid* are returned.

In this implementation, the policy of when a process can obtain the scheduling parameters of another process are detailed in IEEE Std 1003.1b-1993 ("POSIX.1b") as a read-style operation.

If *pid* is zero, the scheduling parameters for the calling process will be returned. In this implementation, the *sched\_getparam* system call will fail if *pid* is negative.

## RETURN VALUES

Upon successful completion, the value 0 is returned; otherwise the value -1 is returned and the global variable *errno* is set to indicate the error.

## ERRORS

On failure *errno* will be set to the corresponding value:

[ENOSYS]	The system is not configured to support this functionality.
[EPERM]	The requesting process doesn not have permission as detailed in IEEE Std 1003.1b-1993 ("POSIX.1b").
[ESRCH]	No process can be found corresponding to that specified by <i>pid</i> .
[EINVAL]	For <b>sched_setparam()</b> : one or more of the requested scheduling parameters is outside the range defined for the scheduling policy of the specified <i>pid</i> .

## SEE ALSO

sched\_get\_priority\_max(2), sched\_get\_priority\_min(2), sched\_getscheduler(2),  
sched\_rr\_get\_interval(2), sched\_setscheduler(2), sched\_yield(2)

**STANDARDS**

The **sched\_setparam()** and **sched\_getparam()** system calls conform to IEEE Std 1003.1b-1993 ("POSIX.1b").