NAME

sched_setparam, sched_getparam – set/get scheduling parameters

SYNOPSIS

cc [*flag* ...] *file* ... **–lposix4** [*library* ...]

#include <sched.h>

int sched_setparam(pid_t *pid***, const struct sched_param** **param***);**

int sched_getparam(pid_t *pid***, struct sched_param** **param***);**

```
struct sched_param {
      int           sched_priority;      /* process execution scheduling priority */
 ...
}
```

MT-LEVEL

MT-Safe

DESCRIPTION

sched_setparam( ) sets the scheduling parameters of the process specified by *pid* to the values specified by the **sched_param** structure referenced by *param.*

sched_getparam( ) stores the scheduling parameters of a process, specified by *pid,* in the **sched_param** structure pointed to by *param.*

If the target process has as its scheduling policy, **SCHED_FIFO** or **SCHED_RR**:

If *pid* is zero, the scheduling parameters are set/stored for the calling process. Otherwise, if a process specified by *pid* exists and if the calling process has permission, the scheduling parameters are set/stored for the process whose process ID is equal to *pid.* The real or effective user ID of the calling process must match the real or saved (from **exec**(2)) user ID of the target process unless the effective user ID of the calling process is **0**. See **intro**(2).

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

If the priority of the process, *pid,* is set higher than that of the lowest priority running process, and if process *pid* is ready to run, then process *pid* preempts a lowest priority running process. Similarly, if the process calling **sched_setparam( )** sets its own priority lower than that of one or more other non-empty process lists, then the process that is the head of the highest priority list preempts 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.

The value of *param*-**>sched_priority** must be an 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.

RETURN VALUES

If successful, **sched_setparam( )** and **sched_getparam( )** returns **0;** otherwise, the priority remains unchanged, the function returns **-1,** and sets **errno** to indicate the error condition.

ERRORS

EINVAL One or more of **sched_setparam( )**'s requested scheduling parameters is outside the range defined for the specified *pid*'s scheduling policy.

ENOSYS **sched_setparam( )** and **sched_getparam( )** are not supported by this implementation.

EPERM The requesting process does not have permission to set/get the scheduling parameters for the specified process, or does not have the appropriate privilege to invoke **sched_setparam( ).**

ESRCH No process can be found corresponding to that specified by *pid*.

SEE ALSO

intro(2), **exec**(2), **sched_setscheduler**(3R)

BUGS

In Solaris 2.5, these functions always return –**1** and set **errno** to **ENOSYS,** because this release does not support the Priority Scheduling option. It is our intention to provide support for these interfaces in future releases.