NAME

sched_setscheduler, **sched_getscheduler** - set/get scheduling policy and scheduler parameters

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

#include <sched.h>

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

int
sched_getscheduler(*pid_t pid*);

DESCRIPTION

The **sched_setscheduler**() system call sets the scheduling policy and scheduling parameters of the process specified by *pid* to *policy* and the parameters specified in the *sched_param* structure pointed to by *param*, respectively. The value of the *sched_priority* member in the *param* structure must be any integer within the inclusive priority range for the scheduling policy specified by *policy*.

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 policy and scheduling parameters will be set for the process whose process ID is equal to *pid*.

If *pid* is zero, the scheduling policy and 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 scheduling policies are in <*sched.h*>:

[SCHED_FIFO] First-in-first-out fixed priority scheduling with no round robin scheduling;

[SCHED_OTHER] The standard time sharing scheduler;

[SCHED_RR] Round-robin scheduling across same priority processes.

The *sched_param* structure is defined in <*sched.h*>:

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

The **sched_getscheduler**() system call returns the scheduling policy of the process specified by *pid*.

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_getscheduler* 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 *pid*.
[EINVAL]	The value of the *policy* argument is invalid, or one or more of the parameters contained in *param* is outside the valid range for the specified scheduling policy.

SEE ALSO

sched_get_priority_max(2), sched_get_priority_min(2), sched_getparam(2), sched_rr_get_interval(2), sched_setparam(2), sched_yield(2)

STANDARDS

The **sched_setscheduler**() and **sched_getscheduler**() system calls conform to IEEE Std 1003.1b-1993 ("POSIX.1b").