NAME

sched_setscheduler - set scheduling policy and scheduling parameters

SYNOPSIS

```
cc [ flag... ] file... -lrt [ library... ]
#include <sched.h>

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

DESCRIPTION

The sched_setscheduler() function 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 sched_param structure is any integer within the inclusive priority range for the scheduling policy specified by policy. The sched_setscheduler() function ignores the other members of the sched_param structure. If the value of pid is negative, the behavior of the sched_setscheduler() function is unspecified.

The possible values for the policy parameter are defined in the header <sched.h> (see sched(3HEAD)):

If a process specified by pid exists and if the calling process has permission, the scheduling policy and scheduling parameters are set 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).

If pid is 0, the scheduling policy and scheduling parameters are set for the calling process.

To change the policy of any process to either of the real time policies SCHED_FIFO or SCHED_RR, the calling process must either have the SCHED_FIFO or SCHED_RR policy or have an effective user ID of 0.

The sched_setscheduler() function is considered successful if it succeeds in setting the scheduling policy and scheduling parameters of the process specified by pid to the values specified by policy and the structure pointed to by param, respectively.

The effect of this function on individual threads is dependent on the scheduling contention scope of the threads:

- For threads with system scheduling contention scope, these functions have no effect on their scheduling.
- For threads with process scheduling contention scope, the threads' scheduling policy and associated parameters will not be affected. However, the scheduling of these threads with respect to threads in other processes may be dependent on the scheduling parameters of their process, which are governed using these functions.

The system supports a two-level scheduling model in which library threads are multiplexed on top of several kernel scheduled entities. The underlying kernel scheduled entities for the system contention scope threads will not be affected by these functions.

The underlying kernel scheduled entities for the process contention scope threads will have their scheduling policy and associated scheduling parameters changed to the values specified in policy and param, respectively. Kernel scheduled entities for use by process contention scope threads that are created after this call completes inherit their scheduling policy and associated scheduling parameters from the process.

This function is not atomic with respect to other threads in the process. Threads are allowed to continue to execute while this function call is in the process of changing the scheduling policy and associated scheduling parameters for the underlying kernel scheduled entities used by the process contention scope threads.

RETURN VALUES

Upon successful completion, the function returns the former scheduling policy of the specified process. If the sched_setscheduler() function fails to complete successfully, the policy and scheduling paramenters remain unchanged, and the function returns -1 and sets errno to indicate the error.

ERRORS

The sched_setscheduler() function will fail if:

EINVAL

The value of policy is invalid, or one or more of the parameters contained in param is outside the valid range for the specified scheduling policy.

ENOSYS

The sched_setscheduler() function is not supported by the system.

EPERM

The requesting process does not have permission to set either or both of the scheduling parameters or the scheduling policy of the specified process.

ESRCH

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

ATTRIBUTES

See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	Standard
MT-Level	MT-Safe

SEE ALSO

priocntl(1), intro(2), exec(2), priocntl(2), librt(3LIB), sched(3HEAD), sched_get_priority_max(3RT), sched_getparam(3RT), sched_getscheduler(3RT), sched_setparam(3RT), attributes(5)

NOTES

Solaris 2.6 was the first release to support libposix4/librt. Prior to this release, this function always returned -1 and set errno to ENOSYS.

SunOS 5.9 Last change: 5 Oct 2001