

Realtime Library Functions sched_setparam(3RT)

NAME

sched_setparam - set scheduling parameters

SYNOPSIS

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

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

DESCRIPTION

The sched_setparam() function 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 sched_param structure is 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. If the value of pid is negative, the behavior of the sched_setparam() function is unspecified.

If a process specified by pid exists and if the calling process has permission, the scheduling parameters will be 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 zero, the scheduling parameters will be set for the calling process.

The target process, 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 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 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 also 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.

If the current scheduling policy for the process specified by pid is not SCHED_FIFO or SCHED_RR, including SCHED_OTHER, the result is equal to priocntl(P_PID, pid, PC_SETPARMS, &pparam), where pparam is an image of *param.

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

- o For threads with system scheduling contention scope, these functions have no effect on their scheduling.
- o For threads with process scheduling contention scope, the threads' scheduling 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.

If an implementation supports a two-level scheduling model in which library threads are multiplexed on top of several kernel scheduled entities, then 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 parameters changed to the value specified in param. 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 for the underlying kernel scheduled entities used by the process contention scope threads.

RETURN VALUES

If successful, the `sched_setparam()` function returns 0.

If the call to `sched_setparam()` is unsuccessful, the priority remains unchanged, and the function returns -1 and sets `errno` to indicate the error.

ERRORS

The `sched_setparam()` function will fail if:

EINVAL

One or more of the requested scheduling parameters is outside the range defined for the scheduling policy of the specified pid.

ENOSYS

The `sched_setparam()` function is not supported by the system.

EPERM The requesting process does not have permission to set 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.

ATTRIBUTES

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

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

SEE ALSO

`intro(2)`, `exec(2)`, `librt(3LIB)`, `sched(3HEAD)`,
`sched_getparam(3RT)`, `sched_getscheduler(3RT)`,
`sched_setscheduler(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