



Linux Ubuntu 26.04 Manual Pages on command 'sched_rr_get_interval.2'

\$ man sched_rr_get_interval.2

sched_rr_get_interval(2) System Calls Manual sched_rr_get_interval(2)

NAME

sched_rr_get_interval - get the SCHED_RR interval for the named process

LIBRARY

Standard C library (libc, -lc)

SYNOPSIS

```
#include <sched.h>

int sched_rr_get_interval(pid_t pid, struct timespec *tp);
```

DESCRIPTION

sched_rr_get_interval() writes into the timespec(3) structure pointed to by tp the round-robin time quantum for the process identified by pid. The specified process should be running under the SCHED_RR scheduling policy.

If pid is zero, the time quantum for the calling process is written into *tp.

RETURN VALUE

On success, sched_rr_get_interval() returns 0. On error, -1 is returned, and errno is set to indicate the error.

ERRORS

- EFAULT Problem with copying information to user space.
- EINVAL Invalid pid.
- ENOSYS The system call is not yet implemented (only on rather old kernels).
- ESRCH Could not find a process with the ID pid.

VERSIONS

Linux

Linux 3.9 added a new mechanism for adjusting (and viewing) the SCHED_RR quantum: the `/proc/sys/kernel/sched_rr_timeslice_ms` file exposes the quantum as a millisecond value, whose default is 100. Writing 0 to this file resets the quantum to the default value.

STANDARDS

POSIX.1-2024.

HISTORY

POSIX.1-2001.

Linux

POSIX does not specify any mechanism for controlling the size of the round-robin time quantum. Older Linux kernels provide a (nonportable) method of doing this. The quantum can be controlled by adjusting the process's nice value (see `setpriority(2)`). Assigning a negative (i.e., high) nice value results in a longer quantum; assigning a positive (i.e., low) nice value results in a shorter quantum. The default quantum is 0.1 seconds; the degree to which changing the nice value affects the quantum has varied somewhat across kernel versions. This method of adjusting the quantum was removed starting with Linux 2.6.24.

NOTES

POSIX systems on which `sched_rr_get_interval()` is available define `_POSIX_PRIORITY_SCHEDULING` in `<unistd.h>`.

SEE ALSO

`timespec(3)`, `sched(7)`

Linux man-pages 6.17

2026-02-08

`sched_rr_get_interval(2)`