NAME

rtprio, **rtprio_thread** - examine or modify realtime or idle priority

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

#include <sys/types.h>
#include <sys/rtprio.h>

int
rtprio(*int function, pid_t pid, struct rtprio *rtp*);

int
rtprio_thread(*int function, lwpid_t lwpid, struct rtprio *rtp*);

DESCRIPTION

The **rtprio**() system call is used to lookup or change the realtime or idle priority of a process, or the calling thread. The **rtprio_thread**() system call is used to lookup or change the realtime or idle priority of a thread.

The *function* argument specifies the operation to be performed. RTP_LOOKUP to lookup the current priority, and RTP_SET to set the priority.

For the **rtprio**() system call, the *pid* argument specifies the process to operate on, 0 for the calling thread. When *pid* is non-zero, the system call reports the highest priority in the process, or sets all threads' priority in the process, depending on value of the *function* argument.

For the **rtprio_thread**() system call, the *lwpid* specifies the thread to operate on, 0 for the calling thread.

The **rtp* argument is a pointer to a struct rtprio which is used to specify the priority and priority type. This structure has the following form:

```
struct rtprio {
        u_short    type;
        u_short prio;
};
```

The value of the *type* field may be RTP_PRIO_REALTIME for realtime priorities, RTP_PRIO_NORMAL for normal priorities, and RTP_PRIO_IDLE for idle priorities. The priority

specified by the *prio* field ranges between 0 and RTP_PRIO_MAX (usually 31). 0 is the highest possible priority.

Realtime and idle priority is inherited through **fork**() and **exec**().

A realtime thread can only be preempted by a thread of equal or higher priority, or by an interrupt; idle priority threads will run only when no other real/normal priority thread is runnable. Higher real/idle priority threads preempt lower real/idle priority threads. Threads of equal real/idle priority are run round-robin.

RETURN VALUES

The **rtprio**() and **rtprio_thread**() functions return the value 0 if successful; otherwise the value -1 is returned and the global variable *errno* is set to indicate the error.

ERRORS

The **rtprio**() and **rtprio_thread**() system calls will fail if:

[EFAULT] The rtp pointer passed to **rtprio**() or **rtprio_thread**() was invalid.

[EINVAL] The specified *prio* was out of range.

[EPERM] The calling thread is not allowed to set the priority. Only root is allowed to change the realtime or idle priority of any thread. Exceptional privileges can be granted through the mac_priority(4) policy and the realtime and idletime user groups. The sysctl(8) variable *security.bsd.unprivileged_idprio* is deprecated. If set to non-zero, it lets any user change the idle priority of threads they own.

[ESRCH] The specified process or thread was not found or visible.

SEE ALSO

nice(1), ps(1), rtprio(1), setpriority(2), nice(3), mac_priority(4), renice(8), p_cansee(9)

AUTHORS

The original author was Henrik Vestergaard Draboel <*hvd@terry.ping.dk*>. This implementation in FreeBSD was substantially rewritten by David Greenman. The **rtprio_thread**() system call was implemented by David Xu.