



Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'getpriority.3p' command

\$ man getpriority.3p

GETPRIORITY(3P) POSIX Programmer's Manual GETPRIORITY(3P)

PROLOG

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

NAME

getpriority, setpriority ? get and set the nice value

SYNOPSIS

```
#include <sys/resource.h>

int getpriority(int which, id_t who);

int setpriority(int which, id_t who, int value);
```

DESCRIPTION

The `getpriority()` function shall obtain the nice value of a process, process group, or user. The `setpriority()` function shall set the nice value of a process, process group, or user to `value+{NZERO}`.

Target processes are specified by the values of the `which` and `who` arguments. The `which` argument may be one of the following values: `PRIO_PROCESS`, `PRIO_PGRP`, or `PRIO_USER`, indicating that the `who` argument is to be interpreted as a process ID, a process group ID, or an effective user ID, respectively. A 0 value for the `who` argument specifies the current process, process group, or user.

The nice value set with `setpriority()` shall be applied to the process.

If the process is multi-threaded, the nice value shall affect all system scope threads in the process.

If more than one process is specified, `getpriority()` shall return value `{NZERO}` less than the lowest nice value pertaining to any of the specified processes, and `setpriority()` shall set the nice values of all of the specified processes to `value+{NZERO}`.

The default nice value is `{NZERO}`; lower nice values shall cause more favorable scheduling. While the range of valid nice values is `[0,{NZERO}*2-1]`, implementations may enforce more restrictive limits. If `value+{NZERO}` is less than the system's lowest supported nice value, `setpriority()` shall set the nice value to the lowest supported value; if `value+{NZERO}` is greater than the system's highest supported nice value, `setpriority()` shall set the nice value to the highest supported value.

Only a process with appropriate privileges can lower its nice value.

Any processes or threads using `SCHED_FIFO` or `SCHED_RR` shall be unaffected by a call to `setpriority()`. This is not considered an error. A process which subsequently reverts to `SCHED_OTHER` need not have its priority affected by such a `setpriority()` call.

The effect of changing the nice value may vary depending on the process-scheduling algorithm in effect.

Since `getpriority()` can return the value `-1` upon successful completion, it is necessary to set `errno` to `0` prior to a call to `getpriority()`. If `getpriority()` returns the value `-1`, then `errno` can be checked to see if an error occurred or if the value is a legitimate nice value.

RETURN VALUE

Upon successful completion, `getpriority()` shall return an integer in the range `-{NZERO}` to `{NZERO}-1`. Otherwise, `-1` shall be returned and `errno` set to indicate the error.

Upon successful completion, `setpriority()` shall return `0`; otherwise, `-1` shall be returned and `errno` set to indicate the error.

ERRORS

The `getpriority()` and `setpriority()` functions shall fail if:

ESRCH No process could be located using the which and who argument values specified.

EINVAL The value of the which argument was not recognized, or the value of the who argument is not a valid process ID, process group ID, or user ID.

In addition, setpriority() may fail if:

EPERM A process was located, but neither the real nor effective user ID of the executing process match the effective user ID of the process whose nice value is being changed.

EACCES A request was made to change the nice value to a lower numeric value and the current process does not have appropriate privileges.

The following sections are informative.

EXAMPLES

Using getpriority()

The following example returns the current scheduling priority for the process ID returned by the call to getpid().

```
#include <sys/resource.h>
...
int which = PRIO_PROCESS;
id_t pid;
int ret;
pid = getpid();
ret = getpriority(which, pid);
```

Using setpriority()

The following example sets the priority for the current process ID to -20.

```
#include <sys/resource.h>
...
int which = PRIO_PROCESS;
id_t pid;
int priority = -20;
int ret;
```

```
pid = getpid();  
ret = setpriority(which, pid, priority);
```

APPLICATION USAGE

The `getpriority()` and `setpriority()` functions work with an offset nice value (nice value `-{NZERO}`). The nice value is in the range `[0,2*{NZERO} -1]`, while the return value for `getpriority()` and the third parameter for `setpriority()` are in the range `[-{NZERO},{NZERO} -1]`.

RATIONALE

None.

FUTURE DIRECTIONS

None.

SEE ALSO

`nice()`, `sched_get_priority_max()`, `sched_setscheduler()`

The Base Definitions volume of POSIX.1-2017, `<sys_resource.h>`

COPYRIGHT

Portions of this text are reprinted and reproduced in electronic form from IEEE Std 1003.1-2017, Standard for Information Technology -- Portable Operating System Interface (POSIX), The Open Group Base Specifications Issue 7, 2018 Edition, Copyright (C) 2018 by the Institute of Electrical and Electronics Engineers, Inc and The Open Group. In the event of any discrepancy between this version and the original IEEE and The Open Group Standard, the original IEEE and The Open Group Standard is the referee document. The original Standard can be obtained online at <http://www.opengroup.org/unix/online.html>.

Any typographical or formatting errors that appear in this page are most likely to have been introduced during the conversion of the source files to man page format. To report such errors, see https://www.kernel.org/doc/man-pages/reporting_bugs.html.