

NAME

getpriority, setpriority - get/set program scheduling priority

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

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

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

int

```
getpriority(int which, int who);
```

int

```
setpriority(int which, int who, int prio);
```

DESCRIPTION

The scheduling priority of the process, process group, or user, as indicated by *which* and *who* is obtained with the **getpriority()** system call and set with the **setpriority()** system call. The *which* argument is one of `PRIO_PROCESS`, `PRIO_PGRP`, or `PRIO_USER`, and *who* is interpreted relative to *which* (a process identifier for `PRIO_PROCESS`, process group identifier for `PRIO_PGRP`, and a user ID for `PRIO_USER`). A zero value of *who* denotes the current process, process group, or user. The *prio* argument is a value in the range -20 to 20. The default priority is 0; lower priorities cause more favorable scheduling.

The **getpriority()** system call returns the highest priority (lowest numerical value) enjoyed by any of the specified processes. The **setpriority()** system call sets the priorities of all of the specified processes to the specified value. Only the super-user may lower priorities.

RETURN VALUES

Since **getpriority()** can legitimately return the value -1, it is necessary to clear the external variable *errno* prior to the call, then check it afterward to determine if a -1 is an error or a legitimate value.

The **setpriority()** function returns the value 0 if successful; otherwise the value -1 is returned and the global variable *errno* is set to indicate the error.

ERRORS

The **getpriority()** and **setpriority()** system calls will fail if:

[ESRCH] No process was located using the *which* and *who* values specified.

[EINVAL] The *which* argument was not one of PRIO_PROCESS, PRIO_PGRP, or PRIO_USER.

In addition to the errors indicated above, **setpriority()** will fail if:

[EPERM] A process was located, but neither its effective nor real user ID matched the effective user ID of the caller.

[EACCES] A non super-user attempted to lower a process priority.

SEE ALSO

nice(1), fork(2), renice(8)

HISTORY

The **getpriority()** system call appeared in 4.2BSD.