NAME

nice − change process priority

SYNOPSIS

#include <unistd.h>

int nice(int *inc***);**

Feature Test Macro Requirements for glibc (see **feature_test_macros**(7)):

nice(): _BSD_SOURCE || _SVID_SOURCE || _XOPEN_SOURCE

DESCRIPTION

nice() adds *inc* to the nice value for the calling process. (A higher nice value means a low priority.) Only the superuser may specify a negative increment, or priority increase. The range for nice values is described in **getpriority**(2).

RETURN VALUE

On success, the new nice value is returned (but see NOTES below). On error, −1 is returned, and *errno* is set appropriately.

ERRORS

EPERM

The calling process attempted to increase its priority by supplying a negative *inc* but has insufficient privileges. Under Linux the **CAP_SYS_NICE** capability is required. (But see the discussion of the **RLIMIT_NICE** resource limit in **setrlimit**(2).)

CONFORMING TO

SVr4, 4.3BSD, POSIX.1-2001. However, the Linux and (g)libc (earlier than glibc 2.2.4) return value is non-standard, see below. SVr4 documents an additional **EINVAL** error code.

NOTES

SUSv2 and POSIX.1-2001 specify that **nice**() should return the new nice value. However, the Linux syscall and the **nice**() library function provided in older versions of (g)libc (earlier than glibc 2.2.4) return 0 on success. The new nice value can be found using **getpriority**(2).

Since glibc 2.2.4, **nice**() is implemented as a library function that calls **getpriority**(2) to obtain the new nice value to be returned to the caller. With this implementation, a successful call can legitimately return −1. To reliably detect an error, set *errno* to 0 before the call, and check its value when **nice**() returns −1.

SEE ALSO

nice(1), **fork**(2), **getpriority**(2), **setpriority**(2), **capabilities**(7), **renice**(8)

COLOPHON

This page is part of release 3.22 of the Linux *man-pages* project. A description of the project, and information about reporting bugs, can be found at http://www.kernel.org/doc/man-pages/.