NAME

renice – alter priority of running processes

SYNOPSIS

renice [**–n** *increment*] [**–g** | **–p** | **–u**] *ID* . . .
renice *priority* [**–p**] *pid* . . . [**–g** *gid* . . .] [**–p** *pid* . . .] [**–u** *user* . . .]
renice *priority* **–g** *gid* . . . [**–g** *gid* . . .] [**–p** *pid* . . .] [**–u** *user* . . .]
renice *priority* **–u** *user* . . . [**–g** *gid* . . .] [**–p** *pid* . . .] [**–u** *user* . . .]

AVAILABILITY

SUNWcsu

DESCRIPTION

The **renice** command alters the scheduling priority of one or more running processes. By default, the processes to be affected are specified by their process IDs.

If the first operand is a number within the valid range of priorities, renice will treat it as a *priority* (as in all but the first synopsis form); otherwise, renice will treat it as an *ID* (as in the first synopsis form).

Altering Process Priority

Users other than the privileged user may only alter the priority of processes they own, and can only monotonically increase their "nice value" within the range **0** to **19**. This prevents overriding administrative fiats. The privileged user may alter the priority of any process and set the priority to any value in the range –**20** to **19**. Useful priorities are: **19** (the affected processes will run only when nothing else in the system wants to), **0** (the "base" scheduling priority) and any negative value (to make things go very fast).

OPTIONS

renice supports the following option features:

- The first operand, *priority*, must precede the options and can have the appearance of a multi-digit option.
- The **–g**, **–p** and **–u** options can each take multiple option-arguments.
- The *pid* option-argument can be used without its **–p** option.

The following options are supported:

–g Interpret all operands or just the *gid* arguments as unsigned decimal integer process group IDs.

–n *increment* Specify how the system scheduling priority of the specified process or processes is to be adjusted. The *increment* option-argument is a positive or negative decimal integer that will be used to modify the system scheduling priority of the specified process or processes.

Positive *increment* values cause a lower system scheduling priority. Negative *increment* values may require appropriate privileges and will cause a higher system scheduling priority.

–p Interpret all operands or just the *pid* arguments as unsigned decimal integer process IDs. The **–p** option is the default if no options are specified.

–u Interpret all operands or just the *user* argument as users. If a user exists with a user name equal to the operand, then the user ID of that user will be used in further processing. Otherwise, if the operand represents an unsigned decimal integer, it will be used as the numeric user ID of the user.

OPERANDS

The following operands are supported:

ID A process ID, process group ID or user name/user ID, depending on the option selected.

priority The value specified is taken as the actual system scheduling priority, rather than as an increment to the existing system scheduling priority. Specifying a scheduling priority higher than that of the existing process may require appropriate privileges.

EXAMPLES

Adjust the system scheduling priority so that process IDs **987** and **32** would have a lower scheduling priority:

example% renice -n 5 -p 987 32

Adjust the system scheduling priority so that group IDs **324** and **76** would have a higher scheduling priority, if the user has the appropriate privileges to do so:

example% renice -n -4 -g 324 76

Adjust the system scheduling priority so that numeric user ID 8 and user **sas** would have a lower scheduling priority:

example% renice -n 4 -u 8 sas

ENVIRONMENT

See **environ**(5) for descriptions of the following environment variables that affect the execution of **renice**: **LC_CTYPE**, **LC_MESSAGES**, and **NLSPATH**.

EXIT STATUS

The following exit values are returned:

0 Successful completion.

>0 An error occurred.

FILES

/etc/passwd map user names to user ID's

SEE ALSO

nice(1), **priocntl**(1), **environ**(5)

NOTES

If you make the priority very negative, then the process cannot be interrupted.

To regain control you must make the priority greater than **0**.

Users other than the privileged user cannot increase scheduling priorities of their own processes, even if they were the ones that decreased the priorities in the first place.

The **priocntl** command subsumes the function of **renice**.