

**NAME**

**renice** - alter priority of running processes

**SYNOPSIS**

**renice** *priority* [[-**gpu**] *target*]

**renice -n** *increment* [[-**gpu**] *target*]

**DESCRIPTION**

The **renice** utility alters the scheduling priority of one or more running processes. The following *target* parameters are interpreted as process ID's (the default), process group ID's, user ID's or user names. The **renice**'ing of a process group causes all processes in the process group to have their scheduling priority altered. The **renice**'ing of a user causes all processes owned by the user to have their scheduling priority altered.

The following options are available:

- n** Instead of changing the specified processes to the given priority, interpret the following argument as an increment to be applied to the current priority of each process.
- g** Interpret *target* parameters as process group ID's.
- p** Interpret *target* parameters as process ID's (the default).
- u** Interpret *target* parameters as user names or user ID's.

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

**FILES**

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

**EXAMPLES**

Change the priority of process ID's 987 and 32, and all processes owned by users daemon and root.

```
renice +1 987 -u daemon root -p 32
```

**SEE ALSO**

nice(1), rtprio(1), getpriority(2), setpriority(2)

**STANDARDS**

The **renice** utility conforms to IEEE Std 1003.1-2001 ("POSIX.1").

**HISTORY**

The **renice** utility appeared in 4.0BSD.

**BUGS**

Non super-users cannot increase scheduling priorities of their own processes, even if they were the ones that decreased the priorities in the first place.