

Modify Process Execution Priorities

Information

These notes were originally written in the year 2000 as part of a set of LPI Exam 101 training materials. The LPI training course at Bromley College was subsequently discontinued and some of the sections of the notes modified and incorporated into our one-day System Administration Courses. The remainder of the notes have now been made publicly available on the linuxtraining.org.uk website.

If you are a beginner please do not be put off of training courses by these notes, as they are rather technical. On the other hand if you are a more experienced Linux user we hope you find the coverage of this topic refreshingly clear.

For full details of our current Linux training please visit the site:

<http://ce.bromley.ac.uk/linux>

If you have reached this page from a search engine and wish to see the full contents list for the published notes please visit the site:

<http://www.linuxtraining.org.uk>

We hope you find these notes useful, but please remember that they apply to the 2.2 kernel. I will update them when I have the time.

Clive Gould - 21st December 2004

Modify Process Execution Priorities

Objective 6

Modify Process Execution Priorities: Run a program with higher or lower priority, determine the priority of a process, change the priority of a running process. Includes the command nice and its relatives.

Scheduling Priorities

Process scheduling priorities vary from -20, the highest priority (very fast), to 20, the lowest priority (process will only run if nothing else in the system wants to), with a base scheduling priority of 0.

Scheduling priorities are displayed for running processes in the PRI and NI columns of the top display as illustrated below:

```
10:09am up 15 days, 1:16, 3 users, load average: 0.08, 0.06, 0.08
87 processes: 85 sleeping, 2 running, 0 zombie, 0 stopped
CPU states: 11.2% user, 3.2% system, 0.0% nice, 85.4% idle
Mem: 128092K av, 118264K used, 9828K free, 84920K shrd, 2728K buff
Swap: 72256K av, 0K used, 72256K free 70560K cached
```

PID	USER	PRI	NI	SIZE	RSS	SHARE	STAT	LIB	%CPU	%MEM	TIME	COMMAND
3120	root	16	0	14028	13M	1676	R	0	9.8	10.9	2:30	X
6039	clive	10	0	1028	1028	824	R	0	2.5	0.8	0:00	top
5957	clive	4	0	3012	3012	2436	S	0	2.1	2.3	0:01	gnome-termin
1	root	0	0	476	476	408	S	0	0.0	0.3	0:03	init
2	root	0	0	0	0	0	SW	0	0.0	0.0	0:00	kflushd
3	root	0	0	0	0	0	SW	0	0.0	0.0	0:00	kpiod
4	root	0	0	0	0	0	SW	0	0.0	0.0	0:00	kswapd
5	root	-20	-20	0	0	0	SW<	0	0.0	0.0	0:00	mdrecoveryd
135	root	0	0	460	460	400	S	0	0.0	0.3	0:00	apmd
291	bin	0	0	376	376	308	S	0	0.0	0.2	0:00	portmap
338	root	0	0	608	608	492	S	0	0.0	0.4	0:03	syslogd
349	root	0	0	752	752	388	S	0	0.0	0.5	0:00	klogd
363	daemon	0	0	472	472	400	S	0	0.0	0.3	0:00	atd
377	root	0	0	592	592	504	S	0	0.0	0.4	0:00	cron
395	root	0	0	560	560	476	S	0	0.0	0.4	0:00	inetd
409	root	0	0	1512	1512	776	S	0	0.0	1.1	0:00	named
423	root	0	0	560	560	472	S	0	0.0	0.4	0:00	lpd
441	root	0	0	528	528	452	S	0	0.0	0.4	0:00	rpc.statd

In the above example you can see that the process with the highest priority is **mdrecoveryd** (-20), whereas the process with the lowest priority in the above display is **X Windows** (+16).

Modify Process Execution Priorities

Run a Command with Modified Scheduling Priority `-nice`

The `nice` command prints or modifies the scheduling priority of a job.

The syntax for the `nice` command is shown below:

```
nice adjustment command command_option(s)
```

If no arguments are given, `nice` prints the current scheduling priority, which it inherited. Otherwise, `nice` runs the given command with its scheduling priority adjusted. If no adjustment is given, the priority of the command is incremented by 10. According to the appropriate info page, the priority can be adjusted by `nice` over the range from -20 (high) to 19 (low).

Only the root can run processes with negative priority. Running a process with an increased priority can have a significant impact on the system and any other processes, including the operating system itself.

An example of using `nice` is illustrated below:

```
[clive@redhat clive]$ nice
0
[clive@redhat clive]$ sleep 200 &
[1] 5979
[clive@redhat clive]$ nice sleep 200 &
[2] 5980
```

When `ps -l` is run at the same time as the two sleep processes the following screen display is obtained:

```
[clive@redhat clive]$ ps -l
 F S  UID  PID  PPID  C PRI  NI ADDR  SZ WCHAN  TTY  TIME  CMD
000 S  500  5959  5957  0  66   0  -   435 wait4 pts/3 00:00:00 bash
000 S  500  5979  5959  0  61   0  -   262 nanosl pts/3 00:00:00 sleep
000 S  500  5980  5959  0  71  10  -   262 nanosl pts/3 00:00:00 sleep
000 R  500  5982  5959  0  72   0  -   621 -      pts/3 00:00:00 ps
```

In the above example you can see that the second sleep process has a higher nice (**NI**) value and therefore a lower priority than the first sleep process.

Modify Process Execution Priorities

Alter Priority of Running Processes - renice

Renice alters the scheduling priority of one or more running processes.

The syntax for renice is shown below:

renice *adjustment arguments*

Common renice arguments are:

<i>Argument</i>	<i>Explanation</i>
<i>PID</i>	Change priority for process with <i>PID</i>
-u <i>USER</i>	Change priority of all processes owned by <i>USER</i>
-g <i>ID</i>	Change priority for all processes in process group <i>ID</i> .

According to the appropriate info page, the priority can be incrementally adjusted by renice over the range from PRIO_MAX (-20) to PRIO_MIN (20). Only the root can adjust process priority so that it becomes more negative.

An example of using renice is illustrated below:

```
[clive@redhat clive]$ sleep 300 &
[1] 6184
[clive@redhat clive]$ renice +5 6184
6184: old priority 0, new priority 5
[clive@redhat clive]$ ps -l
  F S  UID    PID  PPID  C PRI  NI ADDR  SZ WCHAN  TTY    TIME    CMD
000 S  500   6152   6150  0  66   0   -   435 wait4 pts/3   00:00:00 bash
000 S  500   6184   6152  0  66   5   -   262 nanosl pts/3   00:00:00 sleep
000 R  500   6186   6152  0  72   0   -   621 -       pts/3   00:00:00 ps
```

In the above example you can see that the priority of the running process sleep has been reduced by 5. (NI)