



Processos em execução, modificando prioridades

Sumário

Capítulo 1

Processos em execução, modificando as prioridades.....	3
1.1. Mãos a obra.....	4

Capítulo 2

Gerenciando	8
2.1. Objetivos.....	8
2.1. Troubleshooting.....	8

Índice de tabelas

Índice de Figuras

Capítulo 1

Processos em execução, modificando as prioridades

- Prioridade padrão de um trabalho que é criado;
- Executar um programa com prioridade maior ou menor do que o padrão;
- Alterar a prioridade de um processo em execução;
- nice ;
- renice;
- htop.

1.1. Mãos a obra

Prioridade dos programas

Os programas quando estão sendo executados possuem uma prioridade de execução, e o kernel como sendo um escalonador de processos, decide o que irá ser executado e por quanto tempo.

O usuário pode interferir nas prioridades durante a execução dos programas e até antes de inicia-los. Quando o usuário interage com as prioridades, é liberado mais ou menos tempo de CPU.



Mas como eu faço isso?

Através do comando `ps lax`, que exibe uma lista de todos os processos em execução. Na coluna `PRI` é exibido o valor padrão de prioridade de um processo quando executado. Veja um exemplo:

```
debian:~# ps lax
F  UID  PID  PPID PRI  NI   VSZ  RSS  WCHAN  STAT TTY        TIME COMMAND
4    0    1    0  20   0  2100  684  -      Ss   ?         0:01 init [2]
```

Em nosso exemplo o `init` esta com a prioridade 20 na coluna `PRI`, e quando você altera a prioridade de um programa, é exibido um novo valor na coluna `NI`.

Renice

Para mudar a prioridade de um programa em execução, você pode usar alguns comandos ou programas. A prioridades vão de -20 a 19. Vamos a prática.



```
# renice -20 -p 1
```

Em nosso exemplo foi usado o comando `renice` + o valor da nova prioridade + PID do processo. A prioridade do programa foi aumentando, isso porque quanto maior

for o numero da prioridade, mais tempo de CPU ele tem e assim mais importante ele é. Quanto menor for o numero da prioridade, menos tempo de CPU ele terá, e assim menos importância.

Veja agora a nova prioridade com o comando ps lax:

```
debian:~# ps lax
  UID    PID  PPID  PRI  NI   VSZ  RSS  WCHAN  STAT TTY          TIME COMMAND
  0      1      0   0  -20  2100  684  -      S<s  ?           0:01 init [2]
```

Vamos agora diminuir a prioridade do init com o comando renice:



```
# renice +20 -p 1
```

```
debian:~# ps lax
  UID    PID  PPID  PRI  NI   VSZ  RSS  WCHAN  STAT TTY          TIME COMMAND
  0      1      0  39  19  2100  684  -      SNs  ?           0:01 init [2]
```

Veja em nosso exemplo os valores 39 e 19, onde 39 é soma de 20 que é o padrão + 19 máximo de prioridade. NI é a coluna com a prioridade adicionada.

Nice

Quando for necessário iniciar um programa já com uma prioridade diferente, use o comando nice + nova prioridade + programa. Vamos a prática:



```
# nice -n -19 updatedb
```

Em nosso exemplo foi feito uma atualização da base usada pelo comando locate. O tempo de atualização é mais rápida por esta rodando com uma prioridade diferente.

Htop

O htop é uma ferramenta interativa usado para monitorar o seu sistema, além de gerenciar processos, visualizar informações de memória, usuários, usar renice, entre outros.

Para instalar o htop no Debian:



```
# aptitude install htop
```

No RedHat



```
# yum install htop
```

Para iniciar o htop:



```
# htop
```

CPU[] 1.3%										Tasks: 34 total, 1 running	
Mem[] 30/377MB										Load average: 0.03 0.01 0.03	
Swap[] 0/972MB										Uptime: 06:15:16	
PID	USER	PRI	NI	UIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
3693	root	20	0	2416	1192	936	R	0.0	0.3	0:00.28	htop
1927	root	20	0	3388	1044	912	S	0.0	0.3	0:24.50	hald-addon-storage: polling /dev/hdc (every 2 sec)
1	root	39	19	2100	684	588	S	0.0	0.2	0:01.38	init [2]
838	root	16	-4	2292	800	488	S	0.0	0.2	0:00.53	udevd --daemon
2165	root	20	0	28428	1440	952	S	0.0	0.4	0:00.00	/usr/sbin/rsyslogd -c3
2166	root	20	0	28428	1440	952	S	0.0	0.4	0:00.00	/usr/sbin/rsyslogd -c3
1798	root	20	0	28428	1440	952	S	0.0	0.4	0:00.06	/usr/sbin/rsyslogd -c3
1809	root	20	0	1764	644	548	S	0.0	0.2	0:00.00	/usr/sbin/acpid
1819	messageb	20	0	2620	896	700	S	0.0	0.2	0:00.00	/usr/bin/dbus-daemon --system
1836	bind	20	0	42384	9660	2452	S	0.0	2.5	0:00.12	/usr/sbin/named -u bind
1837	bind	20	0	42384	9660	2452	S	0.0	2.5	0:00.00	/usr/sbin/named -u bind
1838	bind	20	0	42384	9660	2452	S	0.0	2.5	0:00.00	/usr/sbin/named -u bind
1835	bind	20	0	42384	9660	2452	S	0.0	2.5	0:00.14	/usr/sbin/named -u bind
1849	root	20	0	5416	1024	668	S	0.0	0.3	0:00.00	/usr/sbin/sshd
1874	root	20	0	6344	2300	1740	S	0.0	0.6	0:00.14	/usr/sbin/cupsd
1885	root	20	0	7080	1640	1208	S	0.0	0.4	0:02.09	/usr/sbin/nmbd -D
1887	root	20	0	12540	2692	2024	S	0.0	0.7	0:00.02	/usr/sbin/smbd -D
1898	root	20	0	12540	1016	348	S	0.0	0.3	0:00.00	/usr/sbin/smbd -D
1899	haldacmo	20	0	5372	3376	2764	S	0.0	0.9	0:00.36	/usr/sbin/hald
1900	root	20	0	3324	1068	912	S	0.0	0.3	0:00.06	hald-runner
1918	root	20	0	3388	1036	908	S	0.0	0.3	0:00.18	hald-addon-input: Listening on /dev/input/event3 /dev/input/event2
1926	haldacmo	20	0	2272	880	752	S	0.0	0.2	0:00.00	hald-addon-acpi: listening on acpid socket /var/run/acpid.socket
1950	root	20	0	3852	1024	856	S	0.0	0.3	0:00.00	/usr/bin/system-tools-backends
1970	daemon	20	0	2048	436	316	S	0.0	0.1	0:00.00	/usr/sbin/atd
1990	root	20	0	3456	984	804	S	0.0	0.3	0:00.00	/usr/sbin/cron
2009	root	20	0	1764	508	436	S	0.0	0.1	0:00.00	/sbin/getty 38400 tty2
2011	root	20	0	2628	1204	948	S	0.0	0.3	0:00.04	/bin/login --
2013	root	20	0	1764	500	436	S	0.0	0.1	0:00.00	/sbin/getty 38400 tty4
2015	root	20	0	1764	500	436	S	0.0	0.1	0:00.00	/sbin/getty 38400 tty5
2017	root	20	0	1764	504	436	S	0.0	0.1	0:00.00	/sbin/getty 38400 tty6
3076	aluno	20	0	5608	2888	1400	S	0.0	0.7	0:00.19	-bash
3131	root	20	0	1764	504	436	S	0.0	0.1	0:00.00	/sbin/getty 38400 tty1
3154	root	20	0	3812	1172	956	S	0.0	0.3	0:00.00	su
3156	root	20	0	4268	1756	1356	S	0.0	0.5	0:00.90	bash

F1Help F2Setup F3Search F4Invert F5Tree F6SortBy F7Nice -F8Nice +F9Kill F10Quit

As teclas na parte inferior (F1 a F10) pode ser usadas conforme seus nomes já explicam a função. Para ter acesso a mais teclas use F1

```
htop 0.7 - (C) 2004-2006 Hisham Muhammad.  
Released under the GNU GPL. See 'man' page for more info.  
  
CPU usage bar: [low-priority/normal/kernel          used%]  
Memory bar:    [used/buffers/cache                used/total]  
Swap bar:      [used                              used/total]  
Type and layout of header meters are configurable in the setup screen.  
Status: R: running; S: sleeping; T: traced/stopped; Z: zombie; D: disk sleep  
  
Arrows: scroll process list          F5 t: tree view  
Digits: incremental PID search      u: show processes of a single user  
F3 /: incremental name search        H: hide/show user threads  
                                       K: hide/show kernel threads  
Space: tag processes                F: cursor follows process  
U: untag all processes  
F9 k: kill process/tagged processes P: sort by CPU%  
+ [ F7: lower priority (+ nice)      M: sort by MEM%  
- ] F8: higher priority (root only)  T: sort by TIME  
                                       F4 I: invert sort order  
F2 S: setup                          F6 >: select sort column  
F1 h: show this help screen  
F10 q: quit                          s: trace syscalls with strace  
  
Press any key to return.
```

Algumas teclas são bem interessantes como P, T, M, t entre outras.



Faça alguns testes com o htop, monitorando os processo de sua maquina.

Capítulo 2

Gerenciando

2.1. Objetivos

- Troubleshooting de prioridades.

2.1. Troubleshooting



Como posso compactar de meus arquivos de forma mais rapida?

O uso de prioridades para executar programas demorados, pode ser de grande ajuda no dia a dia. Como exemplo vamos empacotar e compactar todos os dados do diretório /var com o comando tar. A diferença aqui é o uso do nice aumentando a prioridade do processo. Vamos a prática.



```
# nice -n -19 tar cjvf var.tar.bz2 /var
```