

Dernière mise-à-jour : 2020/08/07 09:33

LDF207 - Gestion des Processus

Contenu du Module

- **LDF207 - Gestion des Processus**
 - Contenu du Module
 - Présentation
 - Les Types de Processus
 - Les Commandes relatives aux Processus
 - La Commande ps
 - La Commande pgrep
 - La Commande pstree
 - La Commande top
 - Les Commandes fg, bg et jobs
 - La Commande wait
 - La Commande nice
 - La Commande renice
 - La Commande nohup
 - La Commande kill
 - La Commande pkill

Présentation

Un processus est un fichier binaire (binary file) qui est chargé en mémoire centrale. Une fois chargé la mémoire exécute le programme en langage machine. Quand le programme est chargé, il a besoin du système d'exploitation qui lui fournit des informations pour qu'il puisse s'exécuter correctement. Ces informations sont appelées des **données d'identification**.

L'ensemble des **données d'identification** est appelé l'**environnement de processus** :

- Un numéro de processus unique (PID),
- Un numéro de processus parent (PPID),
- Un numéro d'utilisateur (UID),
- Un numéro de groupe (GID),
- La durée de traitement,
- La priorité du processus,
- Le répertoire de travail actif,
- Les fichiers ouverts.

Ces informations sont stockés dans le répertoire **/proc**. Le répertoire /proc contient des fichiers et des répertoires virtuels. Le contenu de ces fichiers est créé dynamiquement lors de la consultation. Seul root peut consulter la totalité des informations dans le répertoire /proc.

Saisissez la commande suivante :

```
root@debian8:/# cd /proc; ls -d [0-9]*
1      1024 107   1112 1188 1393 1559 21    31    430 543 71    81     947
10     1026 1080 1116 1197 14    158 219   32    433 544 72    816 951
1000   1030 1081 1117 12    1400 16    2288 397 434 552 73    9    952
1001   1033 1083 1119 1206 143   17    23    408 437 6     74    923 956
1005   1042 1085 1132 1209 15    18    2367 413 443 66    759 924 958
1007   1057 1094 1140 1217 150   1893 2383 415 444 67    763 931 984
1011   1058 1097 1141 1227 1545 19    24    422 475 68    784 932 985
1012   106 11    1142 1255 1548 2     2430 424 479 69    791 934 990
1020   1061 1104 1147 1262 1549 20    3     426 5     7     8     938 992
1023   1069 1108 1159 13    1558 203   30    429 542 70    801 941
```

Chaque répertoire fait référence à un PID d'un processus. Les données de l'**environnement de processus** y sont présentes, par exemple :

```
root@debian8:/proc# cd 1 ; ls -l
total 0
dr-xr-xr-x 2 root root 0 Oct 27 14:12 attr
-rw-r--r-- 1 root root 0 Oct 27 14:12 autogroup
-r----- 1 root root 0 Oct 27 14:12 auxv
-r--r--r-- 1 root root 0 Oct 26 13:51 cgroup
```

```
--w----- 1 root root 0 Oct 27 14:12 clear_refs
-r--r--r-- 1 root root 0 Oct 26 13:51 cmdline
-rw-r--r-- 1 root root 0 Oct 27 14:12 comm
-rw-r--r-- 1 root root 0 Oct 27 14:12 coredump_filter
-r--r--r-- 1 root root 0 Oct 27 14:12 cpuset
lrwxrwxrwx 1 root root 0 Oct 27 14:12 cwd -> /
-r----- 1 root root 0 Oct 27 14:12 environ
lrwxrwxrwx 1 root root 0 Oct 26 13:51 exe -> /lib/systemd/systemd
dr-x----- 2 root root 0 Oct 26 13:51 fd
dr-x----- 2 root root 0 Oct 27 14:12 fdinfo
-rw-r--r-- 1 root root 0 Oct 27 14:12 gid_map
-r----- 1 root root 0 Oct 27 14:12 io
-r--r--r-- 1 root root 0 Oct 26 13:52 limits
-rw-r--r-- 1 root root 0 Oct 27 14:12 loginuid
dr-x----- 2 root root 0 Oct 27 14:12 map_files
-r--r--r-- 1 root root 0 Oct 27 14:12 maps
-rw----- 1 root root 0 Oct 27 14:12 mem
-r--r--r-- 1 root root 0 Oct 26 13:51 mountinfo
-r--r--r-- 1 root root 0 Oct 27 14:12 mounts
-r----- 1 root root 0 Oct 27 14:12 mountstats
dr-xr-xr-x 7 root root 0 Oct 26 13:51 net
dr-x--x--x 2 root root 0 Oct 27 14:12 ns
-rw-r--r-- 1 root root 0 Oct 27 14:12 oom_adj
-r--r--r-- 1 root root 0 Oct 27 14:12 oom_score
-rw-r--r-- 1 root root 0 Oct 26 13:52 oom_score_adj
-r----- 1 root root 0 Oct 27 14:12 pagemap
-r----- 1 root root 0 Oct 27 14:12 personality
-rw-r--r-- 1 root root 0 Oct 27 14:12 projid_map
lrwxrwxrwx 1 root root 0 Oct 26 13:51 root -> /
-rw-r--r-- 1 root root 0 Oct 27 14:12 sched
-r--r--r-- 1 root root 0 Oct 27 14:12 sessionid
-rw-r--r-- 1 root root 0 Oct 27 14:12 setgroups
-r--r--r-- 1 root root 0 Oct 27 14:12 smaps
-r----- 1 root root 0 Oct 27 14:12 stack
```

```
-r--r--r-- 1 root root 0 Oct 26 13:51 stat
-r--r--r-- 1 root root 0 Oct 27 14:12 statm
-r--r--r-- 1 root root 0 Oct 26 13:51 status
-r----- 1 root root 0 Oct 27 14:12 syscall
dr-xr-xr-x 3 root root 0 Oct 26 13:51 task
-r--r--r-- 1 root root 0 Oct 27 14:12 timers
-rw-r--r-- 1 root root 0 Oct 27 14:12 uid_map
-r--r--r-- 1 root root 0 Oct 27 14:12 wchan
```



Important - Vous n'avez pas besoin de consulter le contenu des fichiers et des répertoires. Il convient tout simplement de savoir que ces données existent. Naviguez donc à /root en ligne de commande.

Les Types de Processus

Il existe trois types de processus :

- **interactif** qui est lancé par le shell dans une console en premier plan ou en tâche de fond
- **batch** qui est lancé par le système au moment propice
- **daemon** qui est lancé au démarrage par le système (lpd, dns etc)

Un processus peut être dans un de neuf états ou *process states* :

- *user mode* - le processus s'exécute en mode utilisateur,
- *kernel mode* - le processus s'exécute en mode noyau,
- *waiting* - le processus est en attente pour une ressource autre que le processeur,
- *sleeping* - le processus est endormi,
- *runnable* - le processus dispose de toutes les ressources nécessaires à son exécution sauf le processeur,
- *swap* - le processus est endormi dans la mémoire virtuelle,
- *new* - le processus est nouveau,
- *elected* - le processus a le contrôle du processeur,

- *zombie* – le processus a terminé son exécution et est prêt à mourir.

Les Commandes relatives aux Processus

La Commande ps

Cette commande affiche les processus de l'utilisateur attaché au terminal :

```
root@debian8:/proc/1# cd ~
root@debian8:~# ps
  PID TTY          TIME CMD
 1558 pts/0    00:00:00 su
 1559 pts/0    00:00:00 bash
 2450 pts/0    00:00:00 ps
```

Pour plus de détails, il convient d'utiliser l'option **-l** :

```
root@debian8:~# ps -l
 F S      UID      PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
 4 S        0    1558   1549  0  80   0 -   1449 wait  pts/0    00:00:00 su
 4 S        0    1559   1558  0  80   0 -   1648 wait  pts/0    00:00:00 bash
 0 R        0    2452   1559  0  80   0 -   1131 -      pts/0    00:00:00 ps
```

On note dans cette sortie :

F	Drapeaux du processus. La valeur 4 indique que le processus utilise les privilèges de root
S	État du processus S (sleeping), R (In run queue), Z (zombie), N (low priority), D (uninterruptible sleep), T (Traced)
UID	Numéro de l'Utilisateur
PID	Numéro Unique de Processus
PPID	PID du processus parent
C	Facteur de priorité du processus

PRI	Priorité du processus
NI	La valeur de nice
ADDR	Adresse mémoire du processus
SZ	Utilisation de la mémoire virtuelle
WCHAN	Nom de la fonction du noyau dans laquelle le processus est endormi
TTY	Nom du terminal depuis lequel le processus a été lancé
TIME	Durée d'exécution du processus
CMD	Commande exécutée

Pour visualiser la table des processus, utilisez la commande ps avec les options l et x - la commande affiche tous les processus avec un affichage long :

```
root@debian8:~# ps -lx | more
```

F	UID	PID	PPID	PRI	NI	VSZ	RSS	WCHAN	STAT	TTY	TIME	COMMAND
4	0	1	0	20	0	22992	4316	ep_pol	Ss	?	0:01	/sbin/init
1	0	2	0	20	0	0	0	kthrea	S	?	0:00	[kthreadd]
1	0	3	2	20	0	0	0	smplib	S	?	0:00	[ksoftirqd/0]
1	0	5	2	0	-20	0	0	worker	S<	?	0:00	[kworker/0:0H]
1	0	6	2	20	0	0	0	worker	S	?	0:00	[kworker/u2:0]
1	0	7	2	20	0	0	0	rcu_gp	S	?	0:02	[rcu_sched]
1	0	8	2	20	0	0	0	rcu_gp	S	?	0:00	[rcu_bh]
1	0	9	2	-100	-	0	0	smplib	S	?	0:00	[migration/0]
5	0	10	2	-100	-	0	0	smplib	S	?	0:00	[watchdog/0]
1	0	11	2	0	-20	0	0	rescue	S<	?	0:00	[khelper]
5	0	12	2	20	0	0	0	devtmp	S	?	0:00	[kdevtmpfs]
1	0	13	2	0	-20	0	0	rescue	S<	?	0:00	[netns]
1	0	14	2	20	0	0	0	watchd	S	?	0:00	[khungtaskd]
1	0	15	2	0	-20	0	0	rescue	S<	?	0:00	[writeback]
1	0	16	2	25	5	0	0	ksm_sc	SN	?	0:00	[ksmd]
1	0	17	2	39	19	0	0	khugep	SN	?	0:00	[khugepaged]
1	0	18	2	0	-20	0	0	rescue	S<	?	0:00	[crypto]
1	0	19	2	0	-20	0	0	rescue	S<	?	0:00	[kintegrityd]
1	0	20	2	0	-20	0	0	rescue	S<	?	0:00	[bioset]
1	0	21	2	0	-20	0	0	rescue	S<	?	0:00	[kblockd]

```

1      0      23      2  20      0      0      0 kswapd S   ?           0:00 [kswapd0]
1      0      24      2  20      0      0      0 fsnoti S   ?           0:00 [fsnotify_mark]
--More--

```

On note dans cette sortie certaines informations supplémentaires :

VSZ	La même chose que SZ dans l'exemple ci-dessus
RSS	La mémoire utilisée en kilobytes par le processus
STAT	La même chose que S dans l'exemple ci-dessus

Avec des options a,u et x la commande affiche le résultat suivant :

```

root@debian8:~# ps aux | more
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1   0.0   0.4  22992  4316 ?        Ss   11:27   0:01 /sbin/init
root         2   0.0   0.0      0     0 ?        S    11:27   0:00 [kthreadd]
root         3   0.0   0.0      0     0 ?        S    11:27   0:00 [ksoftirqd/0]
root         5   0.0   0.0      0     0 ?        S<   11:27   0:00 [kworker/0:0H]
root         6   0.0   0.0      0     0 ?        S    11:27   0:00 [kworker/u2:0]
root         7   0.0   0.0      0     0 ?        S    11:27   0:02 [rcu_sched]
root         8   0.0   0.0      0     0 ?        S    11:27   0:00 [rcu_bh]
root         9   0.0   0.0      0     0 ?        S    11:27   0:00 [migration/0]
root        10   0.0   0.0      0     0 ?        S    11:27   0:00 [watchdog/0]
root        11   0.0   0.0      0     0 ?        S<   11:27   0:00 [khelper]
root        12   0.0   0.0      0     0 ?        S    11:27   0:00 [kdevtmpfs]
root        13   0.0   0.0      0     0 ?        S<   11:27   0:00 [netns]
root        14   0.0   0.0      0     0 ?        S    11:27   0:00 [khungtaskd]
root        15   0.0   0.0      0     0 ?        S<   11:27   0:00 [writeback]
root        16   0.0   0.0      0     0 ?        SN   11:27   0:00 [ksmd]
root        17   0.0   0.0      0     0 ?        SN   11:27   0:00 [khugepaged]
root        18   0.0   0.0      0     0 ?        S<   11:27   0:00 [crypto]
root        19   0.0   0.0      0     0 ?        S<   11:27   0:00 [kintegrityd]
root        20   0.0   0.0      0     0 ?        S<   11:27   0:00 [bioset]
root        21   0.0   0.0      0     0 ?        S<   11:27   0:00 [kblockd]

```

```
root      23  0.0  0.0      0      0 ?        S    11:27   0:00 [kswapd0]
root      24  0.0  0.0      0      0 ?        S    11:27   0:00 [fsnotify_mark]
--More--
```

On note dans cette sortie certaines informations supplémentaires :

USER	L'utilisateur du processus
%CPU	Ressources du microprocesseur utilisées par le processus
%MEM	Ressources en mémoire vive utilisées par le processus

Options de la commande ps

Les options de cette commande sont :

```
root@debian8:~# ps --help
```

Usage:

ps [options]

Try 'ps --help <simple|list|output|threads|misc|all>'
or 'ps --help <s|l|o|t|m|a>'
for additional help text.

For more details see ps(1).

La Commande pgrep

La commande **pgrep** permet de rechercher un processus en fonction de son nom et d'autres propriétés puis d'afficher son PID sur la sortie standard.

Par exemple, la commande suivante affiche le PID du processus sshd appartenant à root :

```
root@debian8:~# pgrep -u root sshd
```


429

Tandis que la commande suivante affiche tous les PID des processus appartenant à root ou à trainee :

```
root@debian8:~# pgrep -u root,trainee | more
1
2
3
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
23
24
30
--More--
```

Option de la commande pgrep

```
root@debian8:~# pgrep --help
```

Usage:

pgrep [options] <pattern>

Options:

-d, --delimiter <string>	specify output delimiter
-l, --list-name	list PID and process name
-v, --inverse	negates the matching
-w, --lightweight	list all TID
-c, --count	count of matching processes
-f, --full	use full process name to match
-g, --pgroup <id,...>	match listed process group IDs
-G, --group <gid,...>	match real group IDs
-n, --newest	select most recently started
-o, --oldest	select least recently started
-P, --parent <ppid,...>	match only child processes of the given parent
-s, --session <sid,...>	match session IDs
-t, --terminal <tty,...>	match by controlling terminal
-u, --euid <id,...>	match by effective IDs
-U, --uid <id,...>	match by real IDs
-x, --exact	match exactly with the command name
-F, --pidfile <file>	read PIDs from file
-L, --logpidfile	fail if PID file is not locked
--ns <pid>	match the processes that belong to the same namespace as <pid>
--nslist <ns,...>	list which namespaces will be considered for the --ns option. Available namespaces: ipc, mnt, net, pid, user, uts
-h, --help	display this help and exit
-V, --version	output version information and exit

For more details see `pgrep(1)`.

La Commande pstree

Cette commande affiche les processus en forme d'arborescence, démontrant ainsi les processus parents en enfants :

```

root@debian8:~# pstree
systemd--ModemManager--{gdbus}
                        --{gmain}
--NetworkManager--dhclient
                    --{NetworkManager}
                    --{gdbus}
                    --{gmain}
--VBoxClient--VBoxClient--{SHCLIP}
--VBoxClient--VBoxClient--{VT_MONITOR}
--VBoxClient--VBoxClient--{X11 events}
--VBoxClient--VBoxClient--{dndHGCM}
                        --{dndX11}
--VBoxService--{automount}
                --{control}
                --{cpuhotplug}
                --{memballoon}
                --{timesync}
                --{vminfo}
                --{vmstats}
--accounts-daemon--{gdbus}
                  --{gmain}
--agetty
--2*[at-spi-bus-laun--dbus-daemon]
                    --{dconf worker}}
                    --{gdbus}}
                    --{gmain}}
--2*[at-spi2-registr--{gdbus}}
--atd
--avahi-daemon--avahi-daemon
  
```

```

|colord--{gdbus}
|      |--{gmain}
|cron
|csd-printer--{gdbus}
|cups-browsed
|cupsd
|4*[dbus-daemon]
|3*[dbus-launch]
|dconf-service--{gdbus}
|              |--{gmain}
|exim4
|gconfd-2
|gedit--{dconf worker}
|      |--{gdbus}
|      |--{gmain}
|gnome-keyring-d--{dconf worker}
|                |--{gdbus}
|                |--{gmain}
|                |--{timer}
|gnome-terminal--bash--su--bash--pstree
|                |--gnome-pty-helpe
|                |--{dconf worker}
|                |--{gdbus}
|                |--{gmain}

```

...

Options de la commande pstree

```

root@debian8:~# pstree --help
pstree: unrecognized option '--help'
Usage: pstree [ -a ] [ -c ] [ -h | -H PID ] [ -l ] [ -n ] [ -p ] [ -g ] [ -u ]
           [ -A | -G | -U ] [ PID | USER ]
pstree -V

```

Display a tree of processes.

```
-a, --arguments      show command line arguments
-A, --ascii          use ASCII line drawing characters
-c, --compact        don't compact identical subtrees
-h, --highlight-all highlight current process and its ancestors
-H PID,
--highlight-pid=PID highlight this process and its ancestors
-g, --show-pgids      show process group ids; implies -c
-G, --vt100          use VT100 line drawing characters
-l, --long            don't truncate long lines
-n, --numeric-sort   sort output by PID
-N type,
--ns-sort=type        sort by namespace type (ipc, mnt, net, pid, user, uts)
-p, --show-pids       show PIDs; implies -c
-s, --show-parents    show parents of the selected process
-S, --ns-changes      show namespace transitions
-u, --uid-changes     show uid transitions
-U, --unicode         use UTF-8 (Unicode) line drawing characters
-V, --version         display version information
PID    start at this PID; default is 1 (init)
USER   show only trees rooted at processes of this user
```

La Commande top

Cette commande indique les processus en mémoire :

```
root@debian8:~# top
```

```
top - 14:30:21 up 3:03, 2 users, load average: 0.78, 0.65, 0.52
Tasks: 139 total, 2 running, 137 sleeping, 0 stopped, 0 zombie
%Cpu(s): 26.7 us, 7.6 sy, 0.0 ni, 65.7 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem: 1031864 total, 927092 used, 104772 free, 72984 buffers
```

KiB Swap: 1951740 total, 560 used, 1951180 free. 378380 cached Mem

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1132	trainee	20	0	595944	236540	60584	R	22.7	22.9	24:40.86	cinnamon
784	root	20	0	191492	68876	25684	S	8.2	6.7	6:41.86	Xorg
1545	trainee	20	0	68632	27704	21516	S	3.7	2.7	0:32.08	gnome-terminal-
1012	trainee	20	0	17604	2940	2624	S	0.9	0.3	1:54.42	VBoxClient
1141	trainee	20	0	104552	31260	23824	S	0.1	3.0	0:04.01	nemo
1393	trainee	20	0	106924	41648	25856	S	0.1	4.0	3:07.13	gedit
1400	trainee	20	0	617932	147156	68072	S	0.1	14.3	1:52.13	iceweasel
2514	root	20	0	0	0	0	S	0.1	0.0	0:00.32	kworker/0:1
2569	root	20	0	5120	2756	2388	R	0.1	0.3	0:00.05	top
1	root	20	0	22992	4316	3028	S	0.0	0.4	0:01.92	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	20	0	0	0	0	S	0.0	0.0	0:00.96	ksoftirqd/0
5	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/0:0H
6	root	20	0	0	0	0	S	0.0	0.0	0:00.69	kworker/u2:0
7	root	20	0	0	0	0	S	0.0	0.0	0:02.64	rcu_sched
8	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_bh
9	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	migration/0

...

Pour afficher l'aide de la commande **top**, appuyez sur la touche **h** :

Help for Interactive Commands - procps-ng version 3.3.9

Window 1:Def: Cumulative mode Off. System: Delay 10.0 secs; Secure mode Off.

Z,B,E,e Global: 'Z' colors; 'B' bold; 'E'/'e' summary/task memory scale
 l,t,m Toggle Summary: 'l' load avg; 't' task/cpu stats; 'm' memory info
 0,1,2,3,I Toggle: '0' zeros; '1/2/3' cpus or numa node views; 'I' Irix mode
 f,F,X Fields: 'f'/'F' add/remove/order/sort; 'X' increase fixed-width

L,&,<,> . Locate: 'L'/'&' find/again; Move sort column: '<'/'>' left/right
 R,H,V,J . Toggle: 'R' Sort; 'H' Threads; 'V' Forest view; 'J' Num justify

```

c,i,S,j . Toggle: 'c' Cmd name/line; 'i' Idle; 'S' Time; 'j' Str justify
x,y      . Toggle highlights: 'x' sort field; 'y' running tasks
z,b      . Toggle: 'z' color/mono; 'b' bold/reverse (only if 'x' or 'y')
u,U,o,0 . Filter by: 'u'/'U' effective/any user; 'o'/'O' other criteria
n,#,^0 . Set: 'n'/'#' max tasks displayed; Show: Ctrl+'O' other filter(s)
C,...    . Toggle scroll coordinates msg for: up,down,left,right,home,end

k,r      Manipulate tasks: 'k' kill; 'r' renice
d or s    Set update interval
W,Y      Write configuration file 'W'; Inspect other output 'Y'
q        Quit
          ( commands shown with '.' require a visible task display window )
Press 'h' or '?' for help with Windows,
Type 'q' or <Esc> to continue

```



Important - Pour revenir à l'affichage précédent, appuyez sur la touche **q** ou **echap**.

Au lancement, le temps de rafraîchissement de la liste est de 3 secondes. Pour modifier ce temps à 1 seconde, appuyez sur la touche **s** puis la touche **1** et validez :

```

root@debian8:~# top
...
top - 14:32:07 up 3:04, 2 users, load average: 0.54, 0.61, 0.52
Tasks: 139 total, 1 running, 138 sleeping, 0 stopped, 0 zombie
%Cpu(s): 5.5 us, 2.0 sy, 0.0 ni, 92.5 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem: 1031864 total, 931660 used, 100204 free, 73012 buffers
KiB Swap: 1951740 total, 556 used, 1951184 free. 378404 cached Mem
Change delay from 3.0 to 1
...

```

Pour trier la liste selon l'utilisation de la mémoire, appuyez sur la touche **M** :

```

root@debian8:~# top
...
top - 14:34:12 up 3:07, 2 users, load average: 0.50, 0.59, 0.52
Tasks: 139 total, 1 running, 138 sleeping, 0 stopped, 0 zombie
%Cpu(s): 4.5 us, 2.2 sy, 0.0 ni, 93.3 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem: 1031864 total, 931780 used, 100084 free, 73108 buffers
KiB Swap: 1951740 total, 552 used, 1951188 free. 378432 cached Mem

  PID USER      PR  NI   VIRT   RES   SHR S  %CPU  %MEM    TIME+  COMMAND
 1132 trainee   20   0 595948 236524 60584 S   5.1  22.9 25:42.37 cinnamon
 1400 trainee   20   0 617932 147156 68072 S   0.4  14.3  1:52.32 iceweasel
   784 root       20   0 196036 73420 25684 S   1.5   7.1  6:53.60 Xorg
 1393 trainee   20   0 106924 41648 25856 S   0.0   4.0  3:12.60 gedit
 1141 trainee   20   0 104552 31260 23824 S   0.0   3.0  0:04.07 nemo
 1042 trainee   20   0 156052 29844 24736 S   0.0   2.9  0:06.88 cinnamon-settin
 1545 trainee   20   0  68632 27704 21516 S   0.4   2.7  0:33.31 gnome-terminal-
 1142 trainee   20   0  76592 26152 20584 S   0.0   2.5  0:01.22 nm-applet
 1255 trainee   20   0  93608 25804 22796 S   0.0   2.5  0:14.84 cinnamon-screen
 1116 trainee   20   0  49484 25064 18712 S   0.0   2.4  0:00.48 cinnamon-launch
 1097 trainee   20   0 141840 20524 17588 S   0.0   2.0  0:00.50 goa-daemon
 1262 trainee   20   0  39656 19332 12204 S   0.0   1.9  0:01.11 applet.py
   958 trainee   20   0  59616 18264 15380 S   0.0   1.8  0:01.17 x-session-manag
 1140 trainee   20   0  40636 16788 14384 S   0.0   1.6  0:00.73 polkit-gnome-au
 1197 trainee   20   0  70244 14620 12972 S   0.0   1.4  0:00.30 zeitgeist-fts
   426 root       20   0  58400 13128 11520 S   0.0   1.3  0:00.47 NetworkManager
 1117 colord     20   0  46244 12608  8664 S   0.0   1.2  0:00.34 colord
...

```

Pour ne pas visualiser les processus zombies ou les processus en attente, appuyez sur la touche i :

```

root@debian8:~# top
...
top - 14:37:23 up 3:10, 2 users, load average: 0.72, 0.59, 0.53
Tasks: 139 total, 1 running, 138 sleeping, 0 stopped, 0 zombie

```



```
%Cpu(s): 59.5 us,  9.6 sy,  0.0 ni, 30.8 id,  0.0 wa,  0.0 hi,  0.2 si,  0.0 st
KiB Mem:  1031864 total,  931900 used,   99964 free,   73148 buffers
KiB Swap: 1951740 total,    544 used, 1951196 free.  378440 cached Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1132	trainee	20	0	595948	236484	60584	S	53.1	22.9	26:22.67	cinnamon
1400	trainee	20	0	617932	147156	68072	S	0.2	14.3	1:52.45	iceweasel
784	root	20	0	196036	73420	25684	S	9.0	7.1	7:00.38	Xorg
1393	trainee	20	0	106924	41648	25856	S	5.0	4.0	3:16.03	gedit
1545	trainee	20	0	68776	27704	21516	S	0.6	2.7	0:33.96	gnome-terminal-
1255	trainee	20	0	93608	25804	22796	S	0.2	2.5	0:14.87	cinnamon-screen
1012	trainee	20	0	17604	2940	2624	S	3.4	0.3	2:00.88	VBoxClient
2569	root	20	0	5120	2756	2388	R	0.3	0.3	0:00.40	top
1081	trainee	20	0	6556	2740	2372	S	0.2	0.3	0:00.15	xprop
2553	root	20	0	0	0	0	S	0.2	0.0	0:00.32	kworker/0:0

Pour quitter top, appuyez sur la touche **q**.

Options de la commande top

```
root@debian8:~# top --help
top: inappropriate '-help'
Usage:
  top -hv | -bcHiOSs -d secs -n max -u|U user -p pid(s) -o field -w [cols]
```

Les Commandes fg, bg et jobs

Normalement les commandes s'exécutent en avant plan. Vous pouvez également lancer des processus en arrière plan (en tâche de fond). Si vous lancez une commande en tâche de fond, il faut rajouter **(espace)&** à la fin de la commande :

```
# sleep 9999 &
```



Notez qu'un processus en arrière plan est dit **asynchrone** car il se poursuit indépendamment de son parent qui est le shell. En avant plan le processus est dit **synchrone**.

Linux numérote tous les processus qui sont placés en tâches de fond. On parle donc d'un **numéro de tâche**.

La commande **jobs** permet de se renseigner sur les processus en arrière plan :

```
root@debian8:~# sleep 9999 &
[1] 2621
root@debian8:~# jobs -l
[1]+  2621 Running                  sleep 9999 &
```



Important - Notez que le numéro de tâche est indiqué entre [crochets] tandis que le PID ne l'est pas. Le signe **+** qui suit le numéro de tâche [1] indique que la tâche est la dernière à avoir été manipulée.

Si on souhaite envoyer un processus en arrière plan de façon à libérer le shell pour d'autres commandes, il faut d'abord suspendre le processus en question. Normalement on suspend un processus en utilisant la combinaison de touches CtrlZ.

Par exemple :

```
root@debian8:~# sleep 1234
^Z
[2]+  Stopped                  sleep 1234
```

Un fois suspendu, on utilise la commande **bg** (background) suivi par **%** et le numéro de tâche pour envoyer le processus en arrière plan :

```
root@debian8:~# bg %2
[2]+  sleep 1234 &
```

```
root@debian8:~# jobs -l
[1]-  2621 Running                sleep 9999 &
[2]+  2626 Running                sleep 1234 &
```



Important - Notez que lors du passage en arrière plan, le processus reprend son exécution normalement. Le caractère - qui suit le numéro de tâche [1] indique que la tâche est l'avant-dernière à avoir été manipulée.

Pour ramener le processus en avant plan, il faut de nouveau interrompre le processus concerné. Or cette fois-ci, nous ne pouvons pas utiliser la commande `CtrlZ`. Il faut donc envoyer un **signal** au processus en utilisant la commande **kill** avec l'opérateur **-stop**.

```
root@debian8:~# kill -stop %2
root@debian8:~# jobs -l
[1]-  2621 Running                sleep 9999 &
[2]+  2626 Stopped (signal)       sleep 1234
```

Pour reprendre le processus en arrière plan, sans le ramener en avant plan, on utilise la commande kill avec l'option **-cont** :

```
root@debian8:~# kill -cont %2
root@debian8:~# jobs -l
[1]-  2621 Running                sleep 9999 &
[2]+  2626 Running                sleep 1234 &
```

Pour ramener le processus en avant plan, on utilise la commande fg :

```
root@debian8:~# kill -stop %2
root@debian8:~# jobs -l
[1]-  2621 Running                sleep 9999 &
[2]+  2626 Stopped (signal)       sleep 1234
root@debian8:~# fg %2
sleep 1234
^C
```

```
root@debian8:~#
```



Important - Notez l'utilisation des touches `CtrlC` pour tuer le processus en avant plan.

Options de la commande jobs

Les options de la commande jobs sont :

```
root@debian8:~# help jobs
jobs: jobs [-lnprs] [jobspec ...] or jobs -x command [args]
  Display status of jobs.
  Lists the active jobs. JOBSPEC restricts output to that job.
  Without options, the status of all active jobs is displayed.
  Options:
    -l    lists process IDs in addition to the normal information
    -n    lists only processes that have changed status since the last
          notification
    -p    lists process IDs only
    -r    restrict output to running jobs
    -s    restrict output to stopped jobs
  If -x is supplied, COMMAND is run after all job specifications that
  appear in ARGS have been replaced with the process ID of that job's
  process group leader.
  Exit Status:
  Returns success unless an invalid option is given or an error occurs.
  If -x is used, returns the exit status of COMMAND.
```

La Commande wait

Cette commande permet de doter un processus asynchrone du comportement d'un processus synchrone. Elle est utilisée pour attendre jusqu'à ce qu'un processus en tâche de fond soit terminé :

```
root@debian8:~# jobs -l
[1]+  2621 Running                  sleep 9999 &
root@debian8:~# wait %1
^C
root@debian8:~# jobs -l
[1]+  2621 Running                  sleep 9999 &
```



Important - Notez que l'utilisation des touches `CtrlC` tue le processus généré par la commande **wait** et non le processus généré par la commande **sleep**.

La Commande nice

Cette commande affiche ou modifie la priorité d'un processus. La priorité par défaut de nice est 10. La valeur de nice la plus prioritaire est -20. La valeur la moins prioritaire est 19 :

```
root@debian8:~# nice -n -20 sleep 1234
^Z
[2]+  Stopped                  nice -n -20 sleep 1234

root@debian8:~# ps lx | grep sleep
0      0  2621  1559  20   0   3732    568 hrtime S    pts/0        0:00 sleep 9999
4      0  2944  1559   0 -20   3732    512 signal T<   pts/0        0:00 sleep 1234
0      0  2947  1559  20   0   4536   2172 pipe_w S+   pts/0        0:00 grep sleep
```

```
root@debian8:~# nice -n 19 sleep 5678
^Z
[3]+  Stopped                  nice -n 19 sleep 5678

root@debian8:~# ps lx | grep sleep
0      0  2621  1559  20   0   3732    568 hrtime S    pts/0      0:00 sleep 9999
4      0  2944  1559   0 -20   3732    512 signal T<   pts/0      0:00 sleep 1234
0      0  2949  1559  39  19   3732    520 signal TN   pts/0      0:00 sleep 5678
0      0  2951  1559  20   0   4536   2168 pipe_w S+   pts/0      0:00 grep sleep
```

Comme vous pouvez constater la 6ième colonne contient la valeur de nice qui s'applique à la priorité dans la colonne 5.



Important - Notez que seul root peut lancer des processus avec une valeur négative.

Options de la commande

Les options de cette commande sont :

```
root@debian8:~# nice --help
Usage: nice [OPTION] [COMMAND [ARG]...]
Run COMMAND with an adjusted niceness, which affects process scheduling.
With no COMMAND, print the current niceness.  Niceness values range from
-20 (most favorable to the process) to 19 (least favorable to the process).

Mandatory arguments to long options are mandatory for short options too.
  -n, --adjustment=N  add integer N to the niceness (default 10)
  --help              display this help and exit
  --version           output version information and exit

NOTE: your shell may have its own version of nice, which usually supersedes
```

the version described here. Please refer to your shell's documentation for details about the options it supports.

GNU coreutils online help: <<http://www.gnu.org/software/coreutils/>>
Full documentation at: <<http://www.gnu.org/software/coreutils/nice>>
or available locally via: info '(coreutils) nice invocation'

La Commande renice

Cette commande modifie la priorité d'un processus déjà en cours. La valeur de la priorité ne peut être modifiée que par le propriétaire du processus ou par root.

```
root@debian8:~# jobs -l
[1] 2621 Running          sleep 9999 &
[2]- 2944 Stopped         nice -n -20 sleep 1234
[3]+ 2949 Stopped         nice -n 19 sleep 5678
root@debian8:~# bg %2
[2]- nice -n -20 sleep 1234 &
root@debian8:~# bg %3
[3]+ nice -n 19 sleep 5678 &
root@debian8:~# jobs -l
[1] 2621 Running          sleep 9999 &
[2]- 2944 Running         nice -n -20 sleep 1234 &
[3]+ 2949 Running         nice -n 19 sleep 5678 &
root@debian8:~# renice +5 2944
2944 (process ID) old priority -20, new priority 5
root@debian8:~# renice -5 2949
2949 (process ID) old priority 19, new priority -5
root@debian8:~# ps lx | grep sleep
0      0  2621  1559  20   0  3732   568 hrtime S    pts/0    0:00 sleep 9999
4      0  2944  1559  25   5  3732   512 restar SN   pts/0    0:00 sleep 1234
0      0  2949  1559  15  -5  3732   520 restar S<  pts/0    0:00 sleep 5678
```

```
0      0  2972  1559  20    0  4536  2312 pipe_w S+   pts/0    0:00 grep sleep
```



Important -Notez que seul root peut décrémenter la valeur de priorité avec la commande renice.

Options de la commande

Les options de cette commande sont :

```
root@debian8:~# renice --help
```

Usage:

```
renice [-n] <priority> [-p|--pid] <pid>...
renice [-n] <priority>  -g|--pgrp <pgid>...
renice [-n] <priority>  -u|--user <user>...
```

Options:

```
-g, --pgrp <id>          interpret argument as process group ID
-n, --priority <num>     specify the nice increment value
-p, --pid <id>           interpret argument as process ID (default)
-u, --user <name|id>     interpret argument as username or user ID
-h, --help               display help text and exit
-V, --version             display version information and exit
```

For more information see renice(1).

La Commande nohup

Cette commande permet à un processus de poursuivre son exécution après la déconnexion. Un processus enfant meurt quand le processus parent

meure ou se termine. Comme une connexion et un processus, quand vous vous déconnectez, vos processus se terminent. Pour éviter de rester connecté après avoir lancé un processus long, vous utiliserez la commande `nohup` :

```
nohup lp ventes.txt &
```

Options de la commande

Les options de cette commande sont :

```
root@debian8:~# nohup --help
Usage: nohup COMMAND [ARG]...
      or: nohup OPTION
Run COMMAND, ignoring hangup signals.
```

```
--help      display this help and exit
--version   output version information and exit
```

If standard input is a terminal, redirect it from `/dev/null`.
If standard output is a terminal, append output to `'nohup.out'` if possible, `'$HOME/nohup.out'` otherwise.
If standard error is a terminal, redirect it to standard output.
To save output to FILE, use `'nohup COMMAND > FILE'`.

NOTE: your shell may have its own version of `nohup`, which usually supersedes the version described here. Please refer to your shell's documentation for details about the options it supports.

GNU coreutils online help: [<http://www.gnu.org/software/coreutils/>](http://www.gnu.org/software/coreutils/)
Full documentation at: <http://www.gnu.org/software/coreutils/nohup>
or available locally via: `info '(coreutils) nohup invocation'`

La Commande kill

La commande kill envoie des signaux aux processus. La liste des signaux possibles peut être afficher avec l'option **-l** :

```
root@debian8:~# kill -l
 1) SIGHUP   2) SIGINT   3) SIGQUIT   4) SIGILL   5) SIGTRAP
 6) SIGABRT  7) SIGBUS   8) SIGFPE   9) SIGKILL  10) SIGUSR1
11) SIGSEGV 12) SIGUSR2  13) SIGPIPE 14) SIGALRM 15) SIGTERM
16) SIGSTKFLT 17) SIGCHLD 18) SIGCONT 19) SIGSTOP 20) SIGTSTP
21) SIGTTIN 22) SIGTTOU 23) SIGURG  24) SIGXCPU 25) SIGXFSZ
26) SIGVTALRM 27) SIGPROF 28) SIGWINCH 29) SIGIO   30) SIGPWR
31) SIGSYS  34) SIGRTMIN 35) SIGRTMIN+1 36) SIGRTMIN+2 37) SIGRTMIN+3
38) SIGRTMIN+4 39) SIGRTMIN+5 40) SIGRTMIN+6 41) SIGRTMIN+7 42) SIGRTMIN+8
43) SIGRTMIN+9 44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9  56) SIGRTMAX-8  57) SIGRTMAX-7
58) SIGRTMAX-6 59) SIGRTMAX-5  60) SIGRTMAX-4  61) SIGRTMAX-3  62) SIGRTMAX-2
63) SIGRTMAX-1 64) SIGRTMAX
```



Important - Vous constaterez que chaque signal possède un numéro. Ces numéros de signaux sont utilisés à la place des options. Par exemple, **-19** à la place de l'option **-stop**.

Parmi les numéros de signaux les plus utiles on trouve :

Numéro	Description
-1	Le signal Hang Up est envoyé à tous les enfants d'un processus quand il se termine
-2	Interruption du processus - équivalent à <code>CtrlC</code>
-3	La même chose que -2 mais avec la génération d'un fichier de débogage
-9	Le signal qui tue un processus brutalement
-15	Le signal envoyé par défaut par la commande kill . Le processus se termine normalement

Options de la commande kill

```
root@debian8:~# help kill
kill: kill [-s sigspec | -n signum | -sigspec] pid | jobspec ... or kill -l [sigspec]
  Send a signal to a job.
  Send the processes identified by PID or JOBSPEC the signal named by
  SIGSPEC or SIGNUM.  If neither SIGSPEC nor SIGNUM is present, then
  SIGTERM is assumed.
  Options:
    -s sig      SIG is a signal name
    -n sig      SIG is a signal number
    -l          list the signal names; if arguments follow '-l' they are
                assumed to be signal numbers for which names should be listed
  Kill is a shell builtin for two reasons: it allows job IDs to be used
  instead of process IDs, and allows processes to be killed if the limit
  on processes that you can create is reached.
  Exit Status:
  Returns success unless an invalid option is given or an error occurs.
```

La Commande pkill

La commande pkill permet d'envoyer des signaux aux processus identifiés par leur nom. Par exemple la commande suivante force syslog de relire son fichier de configuration :

```
root@debian8:~# pkill -HUP rsyslogd
```

Options de la commande pkill

```
root@debian8:~# pkill --help
```

Usage:

```
pkill [options] <pattern>
```

Options:

```
-<sig>, --signal <sig>    signal to send (either number or name)
-e, --echo                display what is killed
-c, --count               count of matching processes
-f, --full                use full process name to match
-g, --pgroup <id,...>    match listed process group IDs
-G, --group <gid,...>    match real group IDs
-n, --newest              select most recently started
-o, --oldest              select least recently started
-P, --parent <ppid,...>  match only child processes of the given parent
-s, --session <sid,...>  match session IDs
-t, --terminal <tty,...> match by controlling terminal
-u, --euid <id,...>       match by effective IDs
-U, --uid <id,...>        match by real IDs
-x, --exact               match exactly with the command name
-F, --pidfile <file>      read PIDs from file
-L, --logpidfile          fail if PID file is not locked
--ns <pid>               match the processes that belong to the same
                          namespace as <pid>
--nslist <ns,...>        list which namespaces will be considered for
                          the --ns option.
                          Available namespaces: ipc, mnt, net, pid, user, uts

-h, --help                display this help and exit
-V, --version              output version information and exit
```

For more details see `pgrep(1)`.

<html>

Copyright © 2020 Hugh Norris.

</html>
