Niveau : Admin Junior	Numéro de la Leçon	Dernière Modification
2/4	<progrecss 7/12 style=inline />	2020/01/30 03:28

Gestion des Processus

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@debian:~# cd /proc; ls -d [0-9]*
1     1114  14    1595    1625  1777  1810  1842    1902  26    3781  4181    8
10    1143  141   1596  1695  1778  1812  1847    1904  2994  3782  4187    9
1002  12    1413  1597  1696  1788  1813  1870    2     2996  3784  4300    919
1003  13    1431  1598  17    1792  1815  1872    2095  3     3807  4354    934
1004  131   15    1599    170   1793  1818  1873    21    335   3823  4364    952
1014  132   1508  16    1714  1798  1819  1879    22    3585  3932  5    963
```

```
1021  133   1534  1600  1759  18    1821  1884    23    3588  4      6    974
1027  134   1547  1619  1767  1804  1830  1892    24    3589  402   659    992
1029  136   1549  1622  1770  1806  1833  19    240   3633  4167  671
11    139   1550  1623  1774  1809  1838  190    25    3642  4168  7
```

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

```
root@debian:/proc# cd 1 ; ls -l
total 0
dr-xr-xr-x 2 root root 0  5 mai   10:30 attr
-r-------- 1 root root 0  5 mai   10:30 auxv
-r--r--r-- 1 root root 0  5 mai   10:30 cgroup
--w------- 1 root root 0  5 mai   10:30 clear_refs
-r--r--r-- 1 root root 0  5 mai   10:30 cmdline
-rw-r--r-- 1 root root 0  5 mai   10:30 coredump_filter
-r--r--r-- 1 root root 0  5 mai   10:30 cpuset
lrwxrwxrwx 1 root root 0  5 mai   10:30 cwd -> /
-r-------- 1 root root 0  5 mai   10:30 environ
lrwxrwxrwx 1 root root 0  5 mai   10:30 exe -> /sbin/init
dr-x------ 2 root root 0  5 mai   10:30 fd
dr-x------ 2 root root 0  5 mai   10:30 fdinfo
-r--r--r-- 1 root root 0  5 mai   10:30 io
-r-------- 1 root root 0  5 mai   10:30 limits
-rw-r--r-- 1 root root 0  5 mai   10:30 loginuid
-r--r--r-- 1 root root 0  5 mai   10:30 maps
-rw------- 1 root root 0  5 mai   10:30 mem
-r--r--r-- 1 root root 0  5 mai   10:30 mountinfo
-r--r--r-- 1 root root 0  5 mai   10:30 mounts
-r-------- 1 root root 0  5 mai   10:30 mountstats
dr-xr-xr-x 5 root root 0  5 mai   10:30 net
-rw-r--r-- 1 root root 0  5 mai   10:30 oom_adj
-r--r--r-- 1 root root 0  5 mai   10:30 oom_score
-r-------- 1 root root 0  5 mai   10:30 pagemap
-r-------- 1 root root 0  5 mai   10:30 personality
```

```
lrwxrwxrwx 1 root root 0  5 mai   10:30 root -> /
-rw-r--r-- 1 root root 0  5 mai   10:30 sched
-r--r--r-- 1 root root 0  5 mai   10:30 sessionid
-r--r--r-- 1 root root 0  5 mai   10:30 smaps
-r-------- 1 root root 0  5 mai   10:30 stack
-r--r--r-- 1 root root 0 17 avril 18:44 stat
-r--r--r-- 1 root root 0  5 mai   10:30 statm
-r--r--r-- 1 root root 0 17 avril 18:44 status
-r-------- 1 root root 0  5 mai   10:30 syscall
dr-xr-xr-x 3 root root 0  5 mai   10:30 task
-r--r--r-- 1 root root 0  5 mai   10:30 wchan
```

<note 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. </note>

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* :

<note important>

- *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 le ressources nécessaire 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.

</note>

Les Commandes relatives aux Processus

La commande ps

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

```
root@debian:/proc/1# cd ~
root@debian:~# ps
  PID TTY          TIME CMD
 3633 pts/1    00:00:00 su
 3642 pts/1    00:00:00 bash
 4388 pts/1    00:00:00 ps
```

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

```
root@debian:~# ps -l
F S   UID   PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S     0  3633  3589  0  80   0 -  1104 -      pts/1    00:00:00 su
0 S     0  3642  3633  0  80   0 -  1507 -      pts/1    00:00:00 bash
4 R     0  4402  3642  0  80   0 -   914 -      pts/1    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@debian:~# ps lx
F   UID   PID  PPID PRI  NI    VSZ   RSS WCHAN  STAT TTY        TIME COMMAND
4     0     1     0  20   0   2028   676 -      Ss   ?          0:00 init [2]
1     0     2     0  20   0      0     0 -      S    ?          0:00 [kthreadd]
1     0     3     2 -100  -      0     0 -      S    ?          0:00 [migration]
1     0     4     2  20   0      0     0 -      S    ?          0:00 [ksoftirqd]
5     0     5     2 -100  -      0     0 -      S    ?          0:00 [watchdog/]
1     0     6     2  20   0      0     0 -      S    ?          0:00 [events/0]
1     0     7     2  20   0      0     0 -      S    ?          0:00 [cpuset]
1     0     8     2  20   0      0     0 -      S    ?          0:00 [khelper]
1     0     9     2  20   0      0     0 -      S    ?          0:00 [netns]
1     0    10     2  20   0      0     0 -      S    ?          0:00 [async/mgr]
1     0    11     2  20   0      0     0 -      S    ?          0:00 [pm]
1     0    12     2  20   0      0     0 -      S    ?          0:00 [sync_supe]
1     0    13     2  20   0      0     0 -      S    ?          0:00 [bdi-defau]
1     0    14     2  20   0      0     0 -      S    ?          0:00 [kintegrit]
1     0    15     2  20   0      0     0 -      S    ?          0:00 [kblockd/0]
1     0    16     2  20   0      0     0 -      S    ?          0:00 [kacpid]
```

```
1     0    17     2  20   0      0     0 -      S    ?          0:00 [kacpi_not]
1     0    18     2  20   0      0     0 -      S    ?          0:00 [kacpi_hot]
5     0    19     2  20   0      0     0 -      S    ?          0:00 [kseriod]
1     0    21     2  20   0      0     0 -      S    ?          0:00 [kondemand]
1     0    22     2  20   0      0     0 -      S    ?          0:00 [khungtask]
1     0    23     2  20   0      0     0 -      S    ?          0:00 [kswapd0]
1     0    24     2  25   5      0     0 -      SN   ?          0:00 [ksmd]
1     0    25     2  20   0      0     0 -      S    ?          0:00 [aio/0]
1     0    26     2  20   0      0     0 -      S    ?          0:00 [crypto/0]
1     0   131     2  20   0      0     0 -      S    ?          0:00 [ksuspend_]
5     0   132     2  20   0      0     0 -      S    ?          0:00 [khubd]
1     0   133     2  20   0      0     0 -      S    ?          0:08 [ata/0]
1     0   134     2  20   0      0     0 -      S    ?          0:00 [ata_aux]
1     0   136     2  20   0      0     0 -      S    ?          0:00 [scsi_eh_0]
1     0   139     2  20   0      0     0 -      S    ?          0:00 [scsi_eh_1]
1     0   141     2  20   0      0     0 -      S    ?          0:04 [scsi_eh_2]
1     0   170     2  20   0      0     0 -      S    ?          0:00 [usbhid_re]
1     0   190     2  20   0      0     0 -      S    ?          0:01 [kjournald]
5     0   240     1  16  -4   2528  1076 -      S<s  ?          0:00 udevd --dae
1     0   335     2  20   0      0     0 -      S    ?          0:00 [kpsmoused]
1     0   402     2  20   0      0     0 -      S    ?          0:02 [flush-8:0]
1     0   934     1  20   0   1700   604 -      Ss   ?          0:00 /usr/sbin/a
4     0   963     1  20   0   4176  2252 -      S    ?          0:00 /usr/sbin/m
4     0   974     1  20   0   4752  1700 -      S    ?          0:00 /sbin/wpa_s
4     0   992     1  20   0  16228  2684 -      Sl   ?          0:00 /usr/sbin/g
4     0  1004   992  20   0  18628  3804 -      Sl   ?          0:00 /usr/lib/gd
5     0  1014     1  20   0   4012  1776 -      Ss   ?          0:00 /usr/sbin/b
1     0  1021     2  20   0      0     0 -      S    ?          0:00 [bluetooth]
4     0  1027  1004  20   0  30096 23048 -      Ss+  tty7       2:48 /usr/bin/Xo
5     0  1029     2  10 -10      0     0 -      S<   ?          0:00 [krfcommd]
5     0  1114     1  20   0   3808   932 -      Ss   ?          0:00 /usr/sbin/c
4     0  1143     1  20   0   6584  2440 -      Ss   ?          0:00 /usr/sbin/c
4     0  1431     1  20   0  17144  3024 -      Sl   ?          0:00 /usr/sbin/c
1     0  1508     1  20   0   6820  1064 -      Ss   ?          0:01 /usr/sbin/k
```

```
1     0  1534     1  20   0   6480   768 -      Sl   ?          0:01 /usr/sbin/V
0     0  1547     1  20   0   2636  1136 -      S    ?          0:00 /bin/bash /
0     0  1549     1  20   0   1548   340 -      Ss   ?          0:00 startpar -f
1     0  1550     2  20   0      0     0 -      S    ?          0:00 [kconserva]
0     0  1595     1  20   0   1704   528 -      Ss+  tty1       0:00 /sbin/getty
0     0  1596     1  20   0   1704   532 -      Ss+  tty2       0:00 /sbin/getty
0     0  1597     1  20   0   1704   528 -      Ss+  tty3       0:00 /sbin/getty
0     0  1598     1  20   0   1704   528 -      Ss+  tty4       0:00 /sbin/getty
0     0  1599     1  20   0   1704   532 -      Ss+  tty5       0:00 /sbin/getty
0     0  1600     1  20   0   1704   528 -      Ss+  tty6       0:00 /sbin/getty
4     0  1622     1  20   0   5924  3552 -      S    ?          0:00 /usr/lib/po
4     0  1623  1004  20   0  16832  2988 -      Sl   ?          0:00 /usr/lib/gd
4     0  1625     1  20   0   7876  3188 -      S    ?          0:00 /usr/lib/up
1     0  1696     2  20   0      0     0 -      S    ?          0:00 [kauditd]
4     0  1812     1  20   0   5328  2796 -      S    ?          0:00 /usr/lib/ud
1     0  1813  1812  20   0   5080   764 -      S    ?          0:04 udisks-daem
5     0  2095     1  20   0  15836  3744 -      Ssl  ?          0:00 /usr/sbin/N
5     0  2994     1  16  -4  11720   816 -      S<sl ?          0:00 /sbin/audit
4     0  2996  2994  12  -8  10044   676 -      S<sl ?          0:00 /sbin/audis
4     0  3633  3589  20   0   4416  1560 -      S    pts/1      0:00 su -
0     0  3642  3633  20   0   6028  3332 -      S    pts/1      0:00 -su
5     0  3932     1  20   0  51956  1612 -      Sl   ?          0:00 /usr/sbin/r
5     0  4167   240  18  -2   2524   992 -      S<   ?          0:00 udevd --dae
5     0  4168   240  18  -2   2524  1036 -      S<   ?          0:00 udevd --dae
4     0  4187  2095  20   0   2328  1112 -      S    ?          0:00 /sbin/dhcli
0     0  4398  1547  20   0   1680   412 -      S    ?          0:00 sleep 60
4     0  4407  3642  20   0   3656   828 -      R+   pts/1      0:00 ps lx
```

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@debian:~# ps aux
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0   2028   676 ?        Ss   07:28   0:00 init [2]
root         2  0.0  0.0      0     0 ?        S    07:28   0:00 [kthreadd]
root         3  0.0  0.0      0     0 ?        S    07:28   0:00 [migration/0]
root         4  0.0  0.0      0     0 ?        S    07:28   0:00 [ksoftirqd/0]
root         5  0.0  0.0      0     0 ?        S    07:28   0:00 [watchdog/0]
root         6  0.0  0.0      0     0 ?        S    07:28   0:00 [events/0]
root         7  0.0  0.0      0     0 ?        S    07:28   0:00 [cpuset]
root         8  0.0  0.0      0     0 ?        S    07:28   0:00 [khelper]
root         9  0.0  0.0      0     0 ?        S    07:28   0:00 [netns]
root        10  0.0  0.0      0     0 ?        S    07:28   0:00 [async/mgr]
root        11  0.0  0.0      0     0 ?        S    07:28   0:00 [pm]
root        12  0.0  0.0      0     0 ?        S    07:28   0:00 [sync_supers]
root        13  0.0  0.0      0     0 ?        S    07:28   0:00 [bdi-default]
root        14  0.0  0.0      0     0 ?        S    07:28   0:00 [kintegrityd/0]
root        15  0.0  0.0      0     0 ?        S    07:28   0:00 [kblockd/0]
root        16  0.0  0.0      0     0 ?        S    07:28   0:00 [kacpid]
root        17  0.0  0.0      0     0 ?        S    07:28   0:00 [kacpi_notify]
root        18  0.0  0.0      0     0 ?        S    07:28   0:00 [kacpi_hotplug]
root        19  0.0  0.0      0     0 ?        S    07:28   0:00 [kseriod]
root        21  0.0  0.0      0     0 ?        S    07:28   0:00 [kondemand/0]
root        22  0.0  0.0      0     0 ?        S    07:28   0:00 [khungtaskd]
root        23  0.0  0.0      0     0 ?        S    07:28   0:00 [kswapd0]
root        24  0.0  0.0      0     0 ?        SN   07:28   0:00 [ksmd]
root        25  0.0  0.0      0     0 ?        S    07:28   0:00 [aio/0]
root        26  0.0  0.0      0     0 ?        S    07:28   0:00 [crypto/0]
root       131  0.0  0.0      0     0 ?        S    07:28   0:00 [ksuspend_usbd]
root       132  0.0  0.0      0     0 ?        S    07:28   0:00 [khubd]
root       133  0.0  0.0      0     0 ?        S    07:28   0:08 [ata/0]
root       134  0.0  0.0      0     0 ?        S    07:28   0:00 [ata_aux]
root       136  0.0  0.0      0     0 ?        S    07:28   0:00 [scsi_eh_0]
```

```
root       139  0.0  0.0      0     0 ?        S    07:28   0:00 [scsi_eh_1]
root       141  0.0  0.0      0     0 ?        S    07:28   0:04 [scsi_eh_2]
root       170  0.0  0.0      0     0 ?        S    07:28   0:00 [usbhid_resume]
root       190  0.0  0.0      0     0 ?        S    07:28   0:01 [kjournald]
root       240  0.0  0.1   2528  1076 ?        S<s  07:28   0:00 udevd --daemon
root       335  0.0  0.0      0     0 ?        S    07:28   0:00 [kpsmoused]
root       402  0.0  0.0      0     0 ?        S    07:28   0:02 [flush-8:0]
daemon     659  0.0  0.0   1804   492 ?        Ss   07:28   0:00 /sbin/portmap
statd      671  0.0  0.0   1932   784 ?        Ss   07:28   0:00 /sbin/rpc.statd
101        919  0.0  0.1   3124  1512 ?        Ss   07:28   0:00 /usr/bin/dbus-d
root       934  0.0  0.0   1700   604 ?        Ss   07:28   0:00 /usr/sbin/acpid
daemon     952  0.0  0.0   2156   440 ?        Ss   07:28   0:00 /usr/sbin/atd
root       963  0.0  0.2   4176  2252 ?        S    07:28   0:00 /usr/sbin/modem
root       974  0.0  0.1   4752  1700 ?        S    07:28   0:00 /sbin/wpa_suppl
root       992  0.0  0.2  16228  2684 ?        Sl   07:28   0:00 /usr/sbin/gdm3
avahi     1002  0.0  0.1   2836  1476 ?        S    07:28   0:00 avahi-daemon: r
avahi     1003  0.0  0.0   2836   492 ?        S    07:28   0:00 avahi-daemon: c
root      1004  0.0  0.3  18628  3804 ?        Sl   07:28   0:00 /usr/lib/gdm3/g
root      1014  0.0  0.1   4012  1776 ?        Ss   07:28   0:00 /usr/sbin/bluet
root      1021  0.0  0.0      0     0 ?        S    07:28   0:00 [bluetooth]
root      1027  1.5  2.2  30096 23128 tty7     Ss+  07:28   2:50 /usr/bin/Xorg :
root      1029  0.0  0.0      0     0 ?        S<   07:28   0:00 [krfcommd]
root      1114  0.0  0.0   3808   932 ?        Ss   07:28   0:00 /usr/sbin/cron
root      1143  0.0  0.2   6584  2440 ?        Ss   07:28   0:00 /usr/sbin/cupsd
102       1413  0.0  0.0   6536   940 ?        Ss   07:28   0:00 /usr/sbin/exim4
root      1431  0.0  0.2  17144  3024 ?        Sl   07:28   0:00 /usr/sbin/conso
root      1508  0.0  0.1   6820  1064 ?        Ss   07:28   0:01 /usr/sbin/kerne
root      1534  0.0  0.0   6480   768 ?        Sl   07:28   0:01 /usr/sbin/VBoxS
root      1547  0.0  0.1   2636  1136 ?        S    07:28   0:00 /bin/bash /usr/
root      1549  0.0  0.0   1548   340 ?        Ss   07:28   0:00 startpar -f --
root      1550  0.0  0.0      0     0 ?        S    07:28   0:00 [kconservative]
root      1595  0.0  0.0   1704   528 tty1     Ss+  07:28   0:00 /sbin/getty 384
root      1596  0.0  0.0   1704   532 tty2     Ss+  07:28   0:00 /sbin/getty 384
root      1597  0.0  0.0   1704   528 tty3     Ss+  07:28   0:00 /sbin/getty 384
```

```
root      1598  0.0  0.0   1704   528 tty4     Ss+  07:28   0:00 /sbin/getty 384
root      1599  0.0  0.0   1704   532 tty5     Ss+  07:28   0:00 /sbin/getty 384
root      1600  0.0  0.0   1704   528 tty6     Ss+  07:28   0:00 /sbin/getty 384
106       1619  0.0  0.6  17776  6892 ?        S    07:28   0:00 /usr/lib/policy
root      1622  0.0  0.3   5924  3552 ?        S    07:28   0:00 /usr/lib/policy
root      1623  0.0  0.2  16832  2988 ?        Sl   07:28   0:00 /usr/lib/gdm3/g
root      1625  0.0  0.3   7876  3188 ?        S    07:28   0:00 /usr/lib/upower
trainee   1695  0.0  0.2  24120  2688 ?        Sl   07:30   0:00 /usr/bin/gnome-
root      1696  0.0  0.0      0     0 ?        S    07:30   0:00 [kauditd]
trainee   1714  0.0  0.6  25648  6636 ?        Ssl  07:30   0:00 x-session-manag
trainee   1759  0.0  0.1   6872  2000 ?        Sl   07:31   0:00 /usr/bin/VBoxCl
trainee   1767  0.0  0.1   6728  1548 ?        Sl   07:31   0:00 /usr/bin/VBoxCl
trainee   1770  0.0  0.1   6672  1124 ?        Sl   07:31   0:00 /usr/bin/VBoxCl
trainee   1774  0.0  0.0   3232   340 ?        Ss   07:31   0:00 /usr/bin/ssh-ag
trainee   1777  0.0  0.0   3284   716 ?        S    07:31   0:00 /usr/bin/dbus-l
trainee   1778  0.0  0.1   2972  1164 ?        Ss   07:31   0:00 /usr/bin/dbus-d
trainee   1788  0.0  0.5  18728  5824 ?        Ss   07:31   0:00 /usr/bin/seahor
trainee   1792  0.0  0.2   7180  2392 ?        S    07:31   0:00 /usr/lib/gvfs/g
trainee   1793  0.0  0.4   8644  5132 ?        S    07:31   0:00 /usr/lib/libgco
trainee   1798  0.0  0.9  73616  9808 ?        S    07:31   0:00 gnome-power-man
trainee   1804  0.0  0.9  22716  9948 ?        Ss   07:31   0:02 /usr/lib/gnome-
trainee   1806  0.0  1.0  20368 10428 ?        S    07:31   0:05 /usr/bin/metaci
trainee   1809  0.0  0.3   8008  3372 ?        S    07:31   0:00 /usr/lib/gvfs/g
trainee   1810  0.1  1.8  88792 18712 ?        S    07:31   0:12 gnome-panel
root      1812  0.0  0.2   5328  2796 ?        S    07:31   0:00 /usr/lib/udisks
root      1813  0.0  0.0   5080   764 ?        S    07:31   0:04 udisks-daemon:
trainee   1815  0.0  0.2  16848  2140 ?        Sl   07:31   0:00 /usr/lib/gvfs/g
trainee   1818  0.0  0.2   7068  2220 ?        S    07:31   0:00 /usr/lib/gvfs/g
trainee   1819  0.0  1.7 103704 17888 ?        S    07:31   0:03 nautilus
trainee   1821  0.0  0.3  49204  3424 ?        Ssl  07:31   0:00 /usr/lib/bonobo
trainee   1830  0.0  0.5  16700  5708 ?        S    07:31   0:00 /usr/lib/policy
trainee   1833  0.0  0.5  16340  5968 ?        S    07:31   0:00 kerneloops-appl
trainee   1838  0.0  1.4  31068 15280 ?        S    07:31   0:00 python /usr/bin
trainee   1842  0.0  0.6  18164  6816 ?        S    07:31   0:00 /usr/lib/gnome-
```

```
trainee   1847  0.0  0.6  17204  6948 ?        S    07:31   0:00 bluetooth-apple
trainee   1870  0.0  0.8  29332  8720 ?        S    07:31   0:00 /usr/lib/evolut
trainee   1872  0.0  0.9  74016 10336 ?        S    07:31   0:00 update-notifier
trainee   1873  0.0  1.1 130220 12144 ?        S    07:31   0:00 nm-applet --sm-
trainee   1879  0.0  1.3  89612 13504 ?        Sl   07:31   0:00 /usr/lib/gnome-
trainee   1884  0.0  0.6  18028  6356 ?        Ss   07:31   0:00 gnome-screensav
trainee   1892  0.0  0.2   7540  2972 ?        S    07:31   0:00 /usr/lib/gvfs/g
trainee   1902  0.0  0.2   7184  2472 ?        S    07:31   0:00 /usr/lib/gvfs/g
trainee   1904  0.0  0.1   6180  1896 ?        S    07:31   0:00 /usr/lib/gvfs/g
root      2095  0.0  0.3  15836  3744 ?        Ssl  07:33   0:00 /usr/sbin/Netwo
root      2994  0.0  0.0  11720   816 ?        S<sl 08:23   0:00 /sbin/auditd
root      2996  0.0  0.0  10044   676 ?        S<sl 08:23   0:00 /sbin/audispd
trainee   3585  0.2  1.1  83904 12316 ?        Sl   09:40   0:07 gnome-terminal
trainee   3588  0.0  0.0   1896   684 ?        S    09:40   0:00 gnome-pty-helpe
trainee   3589  0.0  0.3   5856  3164 pts/1    Ss   09:40   0:00 bash
root      3633  0.0  0.1   4416  1560 pts/1    S    09:40   0:00 su -
root      3642  0.0  0.3   6028  3332 pts/1    S    09:40   0:00 -su
root      3932  0.0  0.1  51956  1612 ?        Sl   09:58   0:00 /usr/sbin/rsysl
root      4167  0.0  0.0   2524   992 ?        S<   10:19   0:00 udevd --daemon
root      4168  0.0  0.1   2524  1036 ?        S<   10:19   0:00 udevd --daemon
trainee   4181  0.0  1.0  74592 10480 ?        S    10:19   0:00 /usr/lib/notifi
root      4187  0.0  0.1   2328  1112 ?        S    10:20   0:00 /sbin/dhclient
trainee   4354  2.2  2.2 114796 23636 ?        Sl   10:29   0:05 gedit
root      4417  0.0  0.0   1680   412 ?        S    10:32   0:00 sleep 60
root      4426  0.0  0.0   3868  1012 pts/1    R+   10:33   0:00 ps aux
```

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@debian:~# ps --help
********* simple selection *********  ********* selection by list *********
-A all processes                      -C by command name
-N negate selection                   -G by real group ID (supports names)
-a all w/ tty except session leaders  -U by real user ID (supports names)
-d all except session leaders         -g by session OR by effective group name
-e all processes                      -p by process ID
T  all processes on this terminal     -s processes in the sessions given
a  all w/ tty, including other users  -t by tty
g  OBSOLETE -- DO NOT USE             -u by effective user ID (supports names)
r  only running processes             U  processes for specified users
x  processes w/o controlling ttys     t  by tty
*********** output format **********  *********** long options ***********
-o,o user-defined  -f full            --Group --User --pid --cols --ppid
-j,j job control   s  signal          --group --user --sid --rows --info
-O,O preloaded -o  v  virtual memory  --cumulative --format --deselect
-l,l long          u  user-oriented   --sort --tty --forest --version
-F   extra full    X  registers       --heading --no-heading --context
                    ********* misc options *********
-V,V  show version      L  list format codes  f  ASCII art forest
-m,m,-L,-T,H  threads   S  children in sum    -y change -l format
-M,Z  security data     c  true command name  -c scheduling class
-w,w  wide output       n  numeric WCHAN,UID  -H process hierarchy
```

La commande pstree

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

```
root@debian:~# pstree
init─┬─NetworkManager─┬─dhclient
     │                └─{NetworkManager}
     ├─3*[VBoxClient───{VBoxClient}]
     ├─VBoxService───7*[{VBoxService}]
     ├─acpid
     ├─atd
     ├─auditd─┬─audispd───{audispd}
     │        └─{auditd}
     ├─avahi-daemon───avahi-daemon
     ├─bluetoothd
     ├─bonobo-activati───{bonobo-activat}
     ├─console-kit-dae───63*[{console-kit-da}]
     ├─cron
     ├─cupsd
     ├─2*[dbus-daemon]
     ├─dbus-launch
     ├─exim4
     ├─gconfd-2
     ├─gdm3─┬─gdm-simple-slav─┬─Xorg
     │      │                 ├─gdm-session-wor─┬─x-session-manag─┬─bluetooth-a+
     │      │                 │                 │                 ├─evolution-a+
     │      │                 │                 │                 ├─gdu-notific+
     │      │                 │                 │                 ├─gnome-panel
     │      │                 │                 │                 ├─gnome-power+
     │      │                 │                 │                 ├─kerneloops-+
     │      │                 │                 │                 ├─metacity
     │      │                 │                 │                 ├─nautilus
     │      │                 │                 │                 ├─nm-applet
     │      │                 │                 │                 ├─polkit-gnom+
     │      │                 │                 │                 ├─python
     │      │                 │                 │                 ├─seahorse-ag+
     │      │                 │                 │                 ├─ssh-agent
     │      │                 │                 │                 ├─update-noti+
```

```
│      │                  │                  │                  └─{x-session-+
│      │                  │                  └─{gdm-session-wo}
│      │                  └─{gdm-simple-sla}
│      └─{gdm3}
├─gedit
├─6*[getty]
├─gnome-keyring-d───2*[{gnome-keyring-}]
├─gnome-screensav
├─gnome-settings-
├─gnome-terminal─┬─bash───su───bash───pstree
│                ├─gnome-pty-helpe
│                └─{gnome-terminal}
├─gvfs-afc-volume───{gvfs-afc-volum}
├─gvfs-gdu-volume
├─gvfs-gphoto2-vo
├─gvfsd
├─gvfsd-burn
├─gvfsd-metadata
├─gvfsd-trash
├─kerneloops
├─mixer_applet2───{mixer_applet2}
├─modem-manager
├─notification-da
├─polkit-gnome-au
├─polkitd
├─portmap
├─rpc.statd
├─rsyslogd───6*[{rsyslogd}]
├─startpar
├─udevd───2*[udevd]
├─udisks-daemon───udisks-daemon
├─upowerd
├─webwatch───sleep
└─wpa_supplicant
```

Options de la commande pstree

```
root@debian:~# pstree --help
pstree : option non reconnue « --help »
Usage: pstree [ -a ] [ -c ] [ -h | -H PID ] [ -l ] [ -n ] [ -p ] [ -u ]
              [ -A | -G | -U ] [ PID | UTILISATEUR ]
       pstree -V
Affiche l'arborescence des processus.

  -a, --arguments     afficher les paramètres de la ligne de commande
  -A, --ascii         utiliser les caractères de tracé ASCII
  -c, --compact       ne pas grouper des branches identiques
  -h, --highlight-all surligner le processus courant et ses parents
  -H PID,
  --highlight-pid PID surligner le processus spécifié et ses parents
  -G, --vt100         utiliser les caractères de tracé VT100
  -l, --long          ne pas tronquer les longues lignes
  -n, --numeric-sort  trier le résultat par PID
  -p, --show-pids     afficher les PIDs (implique -c)
  -u, --uid-changes   montrer les transitions de uid
  -U, --unicode       utiliser les caractères de tracé UTF-8 (Unicode)
  -V, --version       afficher les informations sur la version
  PID    commence à ce PID; le défaut est 1 (init)
  USER   montre seulement les arbres nichés aux processus de cet utilisateur
```

La commande top

Cette commande indique les processus en mémoire :

```
top - 10:41:35 up  3:13,  2 users,  load average: 0.00, 0.00, 0.00
Tasks: 123 total,   1 running, 122 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.7%us,  0.7%sy,  0.0%ni, 98.6%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
```

```
Mem:   1034480k total,   488232k used,   546248k free,    51744k buffers
Swap:  1951856k total,        0k used,  1951856k free,   304416k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 1027 root      20   0  157m  22m 7676 S  0.8  2.2   2:56.59 Xorg
 4354 trainee   20   0  112m  23m  14m S  0.7  2.3   0:10.38 gedit
 3585 trainee   20   0 83904  12m 9696 S  0.3  1.2   0:09.84 gnome-terminal
 1810 trainee   20   0 88792  18m  13m S  0.2  1.8   0:12.60 gnome-panel
  141 root      20   0     0    0    0 S  0.1  0.0   0:05.11 scsi_eh_2
 1804 trainee   20   0 22716 9948 7052 S  0.1  1.0   0:02.78 gnome-settings-
 1813 root      20   0  5080  764  536 S  0.1  0.1   0:04.65 udisks-daemon
 4557 root      20   0  2460 1156  880 R  0.1  0.1   0:00.02 top
    1 root      20   0  2028  676  584 S  0.0  0.1   0:00.82 init
    2 root      20   0     0    0    0 S  0.0  0.0   0:00.00 kthreadd
    3 root      RT   0     0    0    0 S  0.0  0.0   0:00.00 migration/0
    4 root      20   0     0    0    0 S  0.0  0.0   0:00.08 ksoftirqd/0
    5 root      RT   0     0    0    0 S  0.0  0.0   0:00.00 watchdog/0
    6 root      20   0     0    0    0 S  0.0  0.0   0:00.28 events/0
    7 root      20   0     0    0    0 S  0.0  0.0   0:00.00 cpuset
    8 root      20   0     0    0    0 S  0.0  0.0   0:00.00 khelper
    9 root      20   0     0    0    0 S  0.0  0.0   0:00.00 netns
   10 root      20   0     0    0    0 S  0.0  0.0   0:00.00 async/mgr
   11 root      20   0     0    0    0 S  0.0  0.0   0:00.00 pm
   12 root      20   0     0    0    0 S  0.0  0.0   0:00.02 sync_supers
...
```

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

```
Help for Interactive Commands - procps version 3.2.8
Window 1:Def: Cumulative mode Off.  System: Delay 10.0 secs; Secure mode Off.

  Z,B       Global: 'Z' change color mappings; 'B' disable/enable bold
  l,t,m     Toggle Summaries: 'l' load avg; 't' task/cpu stats; 'm' mem info
  1,I       Toggle SMP view: '1' single/separate states; 'I' Irix/Solaris mode
```

```
  f,o     . Fields/Columns: 'f' add or remove; 'o' change display order
  F or O  . Select sort field
  <,>     . Move sort field: '<' next col left; '>' next col right
  R,H     . Toggle: 'R' normal/reverse sort; 'H' show threads
  c,i,S   . Toggle: 'c' cmd name/line; 'i' idle tasks; 'S' cumulative time
  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       . Show specific user only
  n or #  . Set maximum tasks displayed

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

<note important> Pour revenir à l'affichage précédent, appuyez sur la barre d'espace. </note>

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 :

```
top - 10:43:05 up  3:14,  2 users,  load average: 0.05, 0.02, 0.00
Tasks: 123 total,   1 running, 122 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.4%us,  0.2%sy,  0.0%ni, 99.4%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:   1034480k total,   488356k used,   546124k free,    51864k buffers
Swap:  1951856k total,        0k used,  1951856k free,   304416k cached
Change delay from 3.0 to: 1
...
```

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

```
top - 10:44:22 up  3:16,  2 users,  load average: 0.01, 0.02, 0.00
```

```
Tasks: 123 total,   2 running, 121 sleeping,   0 stopped,   0 zombie
Cpu(s):  4.2%us,  1.2%sy,  0.0%ni, 94.6%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:   1034480k total,   488480k used,   546000k free,    51936k buffers
Swap:  1951856k total,        0k used,  1951856k free,   304416k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 4354 trainee   20   0  112m  23m  14m S  0.0  2.3   0:13.06 gedit
 1027 root      20   0  157m  22m 7676 R  3.9  2.2   2:59.76 Xorg
 1810 trainee   20   0 88792  18m  13m S  0.0  1.8   0:12.84 gnome-panel
 1819 trainee   20   0  101m  17m  13m S  0.0  1.7   0:03.11 nautilus
 1838 trainee   20   0 31068  14m 8904 S  0.0  1.5   0:00.18 python
 1879 trainee   20   0 89612  13m  10m S  0.0  1.3   0:00.13 mixer_applet2
 3585 trainee   20   0 83904  12m 9696 S  2.1  1.2   0:10.83 gnome-terminal
 1873 trainee   20   0  127m  11m   9m S  0.0  1.2   0:00.57 nm-applet
 1806 trainee   20   0 20988  10m 8512 S  0.0  1.0   0:05.73 metacity
 4181 trainee   20   0 74592  10m 8168 S  0.0  1.0   0:00.25 notification-da
 1872 trainee   20   0 74016  10m 8400 S  0.0  1.0   0:00.99 update-notifier
 1804 trainee   20   0 22716 9948 7052 S  0.0  1.0   0:02.81 gnome-settings-
 1798 trainee   20   0 73616 9808 8176 S  0.0  0.9   0:00.37 gnome-power-man
 1870 trainee   20   0 29332 8720 7228 S  0.0  0.8   0:00.08 evolution-alarm
...
```

Pour visualiser les processus qui utilisent le processeur, appuyez sur la touche i :

```
top - 10:44:59 up  3:16,  2 users,  load average: 0.00, 0.01, 0.00
Tasks: 123 total,   3 running, 120 sleeping,   0 stopped,   0 zombie
Cpu(s):  1.9%us,  1.7%sy,  0.0%ni, 96.4%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:   1034480k total,   488480k used,   546000k free,    51984k buffers
Swap:  1951856k total,        0k used,  1951856k free,   304420k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 1027 root      20   0  157m  22m 7676 R  1.9  2.2   3:00.52 Xorg
 1431 root      20   0 17144 3024 2176 R  0.0  0.3   0:00.15 console-kit-dae
```

```
 4557 root      20   0  2460 1156  880 R  0.0  0.1   0:00.14 top
```

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

Options de la commande top

```
root@debian:~# top --help
    top: procps version 3.2.8
usage:  top -hv | -bcisSH -d delay -n iterations [-u user | -U user] -p pid [,pid ...]
```

Les commandes fg et bg

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 &
```

<note important> 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**. </note>

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@debian:~# sleep 9999 &
[1] 4652
root@debian:~# jobs -l
[1]+  4652 Running                 sleep 9999 &
```

<note 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 a avoir été manipulée. </note>

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@debian:~# 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@debian:~# bg %2
[2]+ sleep 1234 &
root@debian:~# jobs -l
[1]-  4652 Running                 sleep 9999 &
[2]+  4657 Running                 sleep 1234 &
```

<note 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 a avoir été manipulée. </note>

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@debian:~# kill -stop %2
root@debian:~# jobs -l
[1]-  4652 Running                 sleep 9999 &
[2]+  4657 Signal d'arrêt         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@debian:~# kill -cont %2
root@debian:~# jobs -l
[1]-  4652 Running                 sleep 9999 &
```

```
[2]+  4657 Running                 sleep 1234 &
```

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

```
root@debian:~# kill -stop %2
root@debian:~# jobs -l
[1]-  4652 Running                 sleep 9999 &
[2]+  4657 Signal d'arrêt         sleep 1234
root@debian:~# fg %2
sleep 1234
^C
root@debian:~#
```

<note important> Notez l'utilisation des touches CtrlCtrl pour tuer le processus en avant plan. </note>

Options de la commande jobs

Les options de la commande jobs sont :

```
root@debian:~# 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    list 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@debian:~# jobs -l
[1]+  4652 Running                 sleep 9999 &
root@debian:~# wait %1
^C
root@debian:~# jobs -l
[1]+  4652 Running                 sleep 9999 &
```

<note 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**. </note>

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@debian:~# nice -n -20 sleep 1234
^Z
[1]+  Stopped                 nice -n -20 sleep 1234
root@debian:~# ps lx | grep sleep
4     0  2012  1998   0 -20   3172   496 -      T<   pts/0      0:00 sleep 1234
```

```
0     0  2013  1998  20   0   3172   500 -      S    pts/0      0:00 sleep 9999
0     0  2015  1998  20   0   3316   796 -      S+   pts/0      0:00 grep sleep
root@debian:~# nice -n 19 sleep 5678
^Z
[3]+  Stopped                 nice -n 19 sleep 5678
root@debian:~# ps lx | grep sleep
4     0  2012  1998   0 -20   3172   496 -      T<   pts/0      0:00 sleep 1234
0     0  2013  1998  20   0   3172   500 -      S    pts/0      0:00 sleep 9999
0     0  2016  1998  39  19   3172   496 -      TN   pts/0      0:00 sleep 5678
0     0  2018  1998  20   0   3316   796 -      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.

<note important> Notez que seul root peut lancer des processus avec une valeur négative. </note>

Options de la commande

Les options de cette commande sont :

```
root@debian:~# nice --help
Utilisation : nice [OPTION] [COMMAND] [ARG]...
Exécute COMMAND avec un niveau de priorité ajusté.
Sans COMMAND, affiche le niveau actuel de priorité. L'étendue des niveaux va de
-20 (priorité la plus favorable d'ordonnancement) à 19 (la moins favorable).

  -n, --adjustment=N  ajoute la valeur entière N à la valeur de la priorité
                      (10 par défaut)
      --help     affiche l'aide et quitte
      --version  affiche des informations de version et quitte

NOTE : votre shell peut avoir sa propre version de nice, lequel habituellement
remplace la version décrite ici. Consultez la documentation de votre shell pour
les détails concernant les options prises en charge.
```

```
Signalez les anomalies de « nice » à <bug-coreutils@gnu.org>
Page d'accueil de « GNU coreutils » : <http://www.gnu.org/software/coreutils/>
Aide générale sur les logiciels GNU : <http://www.gnu.org/gethelp/>
Traduction de « nice » à <http://translationproject.org/team/fr.html>
Pour une documentation complète, lancer « 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@debian:~# jobs -l
[1]-  2012 Arrêté                 nice -n -20 sleep 1234
[2]   2013 Running                  sleep 9999 &
[3]+  2016 Arrêté                 nice -n 19 sleep 5678
root@debian:~# bg %1
[1]- nice -n -20 sleep 1234 &
root@debian:~# bg %3
[3]+ nice -n 19 sleep 5678 &
root@debian:~# jobs -l
[1]   2012 Running                  nice -n -20 sleep 1234 &
[2]-  2013 Running                  sleep 9999 &
[3]+  2016 Running                  nice -n 19 sleep 5678 &
root@debian:~# renice +5 2012
2012: old priority -20, new priority 5
root@debian:~# renice 5 2012
2012: old priority 5, new priority 5
root@debian:~# renice -5 2016
2016: old priority 19, new priority -5
root@debian:~# ps lx | grep sleep
4     0  2012  1998  25   5   3172   496 -      SN   pts/0      0:00 sleep 1234
0     0  2013  1998  20   0   3172   500 -      S    pts/0      0:00 sleep 9999
0     0  2016  1998  15  -5   3172   496 -      S<   pts/0      0:00 sleep 5678
```

```
0     0  2028  1998  20   0   3316   796 -      S+   pts/0      0:00 grep sleep
```

<note important> Notez que seul root peut décrémenter la valeur de priorité avec la commande renice. </note>

Options de la commande

Les options de cette commande sont :

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

Usage:
 renice [-n] priority [-p|--pid] pid  [... pid]
 renice [-n] priority  -g|--pgrp pgrp [... pgrp]
 renice [-n] priority  -u|--user user [... user]
 renice -h | --help
 renice -v | --version
```

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@debian:~# nohup --help
```

```
Utilisation : nohup COMMAND [ARG]...
         ou : nohup OPTION
Exécute COMMAND en ignorant les signaux de déconnexion.

      --help     affiche l'aide et quitte
      --version  affiche des informations de version et quitte

Si l'entrée standard est un terminal, la redirige depuis /dev/null.
Si l'entrée standard est un terminal, ajoute si possible la sortie à
« nohup.out » ou à « $HOME/nohup.out » sinon.
Si le fichier standard d'erreur est un terminal, la redirige sur la sortie
standard.
Pour enregistrer la sortie dans FILE, utilisez « nohup COMMAND > FILE ».

NOTE : votre shell peut avoir sa propre version de nohup, lequel habituellement
remplace la version décrite ici. Consultez la documentation de votre shell pour
les détails concernant les options prises en charge.

Signalez les anomalies de « nohup » à <bug-coreutils@gnu.org>
Page d'accueil de « GNU coreutils » : <http://www.gnu.org/software/coreutils/>
Aide générale sur les logiciels GNU : <http://www.gnu.org/gethelp/>
Traduction de « nohup » à <http://translationproject.org/team/fr.html>
Pour une documentation complète, lancer « 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@debian:~# 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
```

<note 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**. </note>

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 CtrlC
-3	La même chose que -2 mais avec la génération d'un fichier de déboggage
-9	Le signal qui tue un processus brutalement
-15	Le signal envoyé par défaut par la commande **kill**. Le processus se termine normalement

~~DISCUSSION:off~~

Donner votre Avis

{(rater>id=debian_6_l112|name=cette page|type=rate|trace=user|tracedetails=1)}