

Dernière mise-à-jour : 2020/01/30 03:27

## 103.5 - Créer, surveiller et tuer des processus (4/60)

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@centos7 ~]# cd /proc; ls -d [0-9]*
1      1347  20    2421  27     3005  3132  3249  3319  3697  407    436  495  571
10     14     21     246   2740   3074  3144  3252  3359  370    4070  454  498  596
1015   1410   218    2481  28     3076  3160  3256  3395  373    4071  458  5    6
11     15     22     25    2849   3085  3196  3266  3408  3756  4077  466  502  60
12     16     220    258   2890   3088  3201  3273  3418  3759  41     478  505  625
1217   1677   221    259   29     3090  3202  3274  3428  3760  4111  482  506  7
1219   1687   222    26    2902   3092  3219  3285  3434  38     4112  484  509  700
1228   1688   223    260   2923   3094  3224  3289  3436  381    4113  485  520  8
```

|      |    |     |     |      |      |      |      |      |      |     |     |     |    |
|------|----|-----|-----|------|------|------|------|------|------|-----|-----|-----|----|
| 1229 | 17 | 224 | 261 | 2931 | 3095 | 3228 | 3291 | 3460 | 3873 | 431 | 488 | 521 | 83 |
| 1231 | 18 | 227 | 262 | 2932 | 3108 | 3231 | 3304 | 3477 | 3883 | 432 | 489 | 523 | 9  |
| 13   | 19 | 229 | 263 | 2997 | 3124 | 3239 | 3309 | 362  | 40   | 433 | 491 | 525 |    |
| 1327 | 2  | 24  | 264 | 3    | 3128 | 3243 | 3311 | 3639 | 4007 | 434 | 492 | 552 |    |

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

```
[root@centos7 proc]# cd 1 ; ls -l
total 0
dr-xr-xr-x. 2 root root 0 Oct 27 16:20 attr
-rw-r--r--. 1 root root 0 Oct 27 16:20 autogroup
-r-----. 1 root root 0 Oct 27 16:20 auxv
-r--r--r--. 1 root root 0 Oct 27 16:06 cgroup
--w-----. 1 root root 0 Oct 27 16:20 clear_refs
-r--r--r--. 1 root root 0 Oct 27 16:06 cmdline
-rw-r--r--. 1 root root 0 Oct 27 16:06 comm
-rw-r--r--. 1 root root 0 Oct 27 16:20 coredump_filter
-r--r--r--. 1 root root 0 Oct 27 16:20 cpuset
lrwxrwxrwx. 1 root root 0 Oct 27 16:20 cwd -> /
-r-----. 1 root root 0 Oct 27 16:06 environ
lrwxrwxrwx. 1 root root 0 Oct 27 16:06 exe -> /usr/lib/systemd/systemd
dr-x-----. 2 root root 0 Oct 27 16:06 fd
dr-x-----. 2 root root 0 Oct 27 16:20 fdinfo
-rw-r--r--. 1 root root 0 Oct 27 16:20 gid_map
-r-----. 1 root root 0 Oct 27 16:20 io
-r--r--r--. 1 root root 0 Oct 27 16:20 limits
-rw-r--r--. 1 root root 0 Oct 27 16:06 loginuid
-r--r--r--. 1 root root 0 Oct 27 16:06 maps
-rw-----. 1 root root 0 Oct 27 16:20 mem
-r--r--r--. 1 root root 0 Oct 27 16:06 mountinfo
-r--r--r--. 1 root root 0 Oct 27 16:06 mounts
-r-----. 1 root root 0 Oct 27 16:20 mountstats
dr-xr-xr-x. 6 root root 0 Oct 27 16:06 net
dr-x--x--x. 2 root root 0 Oct 27 16:20 ns
```

```
-r--r--r--. 1 root root 0 Oct 27 16:20 numa_maps
-rw-r--r--. 1 root root 0 Oct 27 16:20 oom_adj
-r--r--r--. 1 root root 0 Oct 27 16:20 oom_score
-rw-r--r--. 1 root root 0 Oct 27 16:20 oom_score_adj
-r--r--r--. 1 root root 0 Oct 27 16:20 pagemap
-r--r--r--. 1 root root 0 Oct 27 16:20 personality
-rw-r--r--. 1 root root 0 Oct 27 16:20 projid_map
lrwxrwxrwx. 1 root root 0 Oct 27 16:06 root -> /
-rw-r--r--. 1 root root 0 Oct 27 16:20 sched
-r--r--r--. 1 root root 0 Oct 27 16:06 sessionid
-r--r--r--. 1 root root 0 Oct 27 16:20 smaps
-r--r--r--. 1 root root 0 Oct 27 16:20 stack
-r--r--r--. 1 root root 0 Oct 27 16:06 stat
-r--r--r--. 1 root root 0 Oct 27 16:20 statm
-r--r--r--. 1 root root 0 Oct 27 16:06 status
-r--r--r--. 1 root root 0 Oct 27 16:20 syscall
dr-xr-xr-x. 3 root root 0 Oct 27 16:06 task
-rw-r--r--. 1 root root 0 Oct 27 16:20 uid_map
-r--r--r--. 1 root root 0 Oct 27 16:20 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@centos7 ~]# cd ~
[root@centos7 ~]# ps
  PID TTY          TIME CMD
 4070 pts/0    00:00:00 su
 4077 pts/0    00:00:00 bash
 4285 pts/0    00:00:00 ps
```

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

```
[root@centos7 ~]# ps -l
F S      UID      PID  PPID  C  PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S        0   4070   3760  0   80   0 - 50611 wait  pts/0    00:00:00 su
```

```

4 S      0  4077  4070  0  80   0 - 29027 wait   pts/0    00:00:00 bash
0 R      0  4309  4077  0  80   0 - 30319 -      pts/0    00:00:00 ps

```

On note dans cette sortie :

|              |                                                                                                                       |
|--------------|-----------------------------------------------------------------------------------------------------------------------|
| <b>F</b>     | Drapeaux du processus. La valeur 4 indique que le processus utilise les privilèges de root                            |
| <b>S</b>     | État du processus S (sleeping), R (In run queue), Z (zombie), N (low priority), D (uninterruptible sleep), T (Traced) |
| <b>UID</b>   | Numéro de l'Utilisateur                                                                                               |
| <b>PID</b>   | Numéro Unique de Processus                                                                                            |
| <b>PPID</b>  | PID du processus parent                                                                                               |
| <b>C</b>     | Facteur de priorité du processus                                                                                      |
| <b>PRI</b>   | Priorité du processus                                                                                                 |
| <b>NI</b>    | La valeur de nice                                                                                                     |
| <b>ADDR</b>  | Adresse mémoire du processus                                                                                          |
| <b>SZ</b>    | Utilisation de la mémoire virtuelle                                                                                   |
| <b>WCHAN</b> | Nom de la fonction du noyau dans laquelle le processus est endormi                                                    |
| <b>TTY</b>   | Nom du terminal depuis lequel le processus a été lancé                                                                |
| <b>TIME</b>  | Durée d'exécution du processus                                                                                        |
| <b>CMD</b>   | 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@centos7 ~]# ps lx | more
F  UID  PID  PPID  PRI  NI   VSZ   RSS WCHAN  STAT TTY      TIME COMMAND
4    0    1    0  20    0  59600  7028 ep_pol  Ss   ?        0:03 /usr/lib/systemd/systemd --switched-root
--system --deserialize 24
1    0    2    0  20    0    0    0 kthrea  S     ?        0:00 [kthreadd]
1    0    3    2  20    0    0    0 smpboo  S     ?        0:01 [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 -100   -    0    0 smpboo  S     ?        0:00 [migration/0]
1    0    8    2  20    0    0    0 rcu_gp  S     ?        0:00 [rcu_bh]
1    0    9    2  20    0    0    0 rcu_no  S     ?        0:00 [rcuob/0]

```

```

1      0      10      2  20    0      0      0 rcu_gp S    ?      0:01 [rcu_sched]
1      0      11      2  20    0      0      0 rcu_no S    ?      0:02 [rcuos/0]
5      0      12      2 -100   -      0      0 smpboo S    ?      0:00 [watchdog/0]
1      0      13      2   0 -20      0      0 rescue S<   ?      0:00 [khelper]
5      0      14      2  20    0      0      0 devtmp S    ?      0:00 [kdevtmpfs]
1      0      15      2   0 -20      0      0 rescue S<   ?      0:00 [netns]
1      0      16      2   0 -20      0      0 rescue S<   ?      0:00 [writeback]
1      0      17      2   0 -20      0      0 rescue S<   ?      0:00 [kintegrityd]
1      0      18      2   0 -20      0      0 rescue S<   ?      0:00 [bioset]
1      0      19      2   0 -20      0      0 rescue S<   ?      0:00 [kblockd]
1      0      20      2  20    0      0      0 hub_th S    ?      0:00 [khubd]
1      0      21      2   0 -20      0      0 rescue S<   ?      0:00 [md]
1      0      24      2  20    0      0      0 watchd S    ?      0:00 [khungtaskd]
--More--

```

On note dans cette sortie certaines informations supplémentaires :

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

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

```

[root@centos7 ~]# ps aux | more
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1   0.3   0.3  59600  7028 ?        Ss   16:06   0:03 /usr/lib/systemd/systemd --switched-root --s
ystem --deserialize 24
root         2   0.0   0.0     0     0 ?        S    16:06   0:00 [kthreadd]
root         3   0.1   0.0     0     0 ?        S    16:06   0:01 [ksoftirqd/0]
root         5   0.0   0.0     0     0 ?        S<   16:06   0:00 [kworker/0:0H]
root         6   0.0   0.0     0     0 ?        S    16:06   0:00 [kworker/u2:0]
root         7   0.0   0.0     0     0 ?        S    16:06   0:00 [migration/0]
root         8   0.0   0.0     0     0 ?        S    16:06   0:00 [rcu_bh]
root         9   0.0   0.0     0     0 ?        S    16:06   0:00 [rcuob/0]

```

```

root      10  0.1  0.0      0      0 ?      S   16:06  0:01 [rcu_sched]
root      11  0.2  0.0      0      0 ?      R   16:06  0:02 [rcuos/0]
root      12  0.0  0.0      0      0 ?      S   16:06  0:00 [watchdog/0]
root      13  0.0  0.0      0      0 ?      S<  16:06  0:00 [khelper]
root      14  0.0  0.0      0      0 ?      S   16:06  0:00 [kdevtmpfs]
root      15  0.0  0.0      0      0 ?      S<  16:06  0:00 [netns]
root      16  0.0  0.0      0      0 ?      S<  16:06  0:00 [writeback]
root      17  0.0  0.0      0      0 ?      S<  16:06  0:00 [kintegrityd]
root      18  0.0  0.0      0      0 ?      S<  16:06  0:00 [bioaset]
root      19  0.0  0.0      0      0 ?      S<  16:06  0:00 [kblockd]
root      20  0.0  0.0      0      0 ?      S   16:06  0:00 [khubd]
root      21  0.0  0.0      0      0 ?      S<  16:06  0:00 [md]
root      24  0.0  0.0      0      0 ?      S   16:06  0:00 [khungtaskd]
--More--

```

On note dans cette sortie certaines informations supplémentaires :

|             |                                                          |
|-------------|----------------------------------------------------------|
| <b>USER</b> | L'utilisateur du processus                               |
| <b>%CPU</b> | Ressources du microprocesseur utilisées par le processus |
| <b>%MEM</b> | Ressources en mémoire vive utilisées par le processus    |

### Options de la commande ps

Les options de cette commande sont :

```
[root@centos7 ~]# 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@centos7 ~]# pgrep -u root sshd
1219
```

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

```
[root@centos7 ~]# 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
24
25
26
--More--
```

## Option de la commande pgrep

```
[root@centos7 ~]# pgrep --help
```

### Usage:

```
pgrep [options] <pattern>
```

### Options:

|                          |                                                |
|--------------------------|------------------------------------------------|
| -d, --delimiter <string> | specify output delimiter                       |
| -l, --list-name          | list PID and process name                      |
| -a, --list-full          | list PID and full command line                 |
| -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 <PGID,...>  | 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@centos7 ~]# pstree
systemd├─ModemManager─2*[{ModemManager}]
      │├─NetworkManager├─dhclient
      ││                 └─3*[{NetworkManager}]
      │├─3*[VBoxClient─VBoxClient─{VBoxClient}]
      │├─VBoxClient─VBoxClient─2*[{VBoxClient}]
      │├─VBoxService─7*[{VBoxService}]
      │├─2*[abrt-watch-log]
      │├─abrttd
      │├─accounts-daemon─2*[{accounts-daemon}]
      │├─alsactl
      │├─at-spi-bus-laun├─dbus-daemon─{dbus-daemon}
      ││               └─3*[{at-spi-bus-laun}]
      │├─at-spi2-registr─{at-spi2-registr}
      │├─atd
      │├─auditd├─audispd├─sedispatch
      ││         │       └─{audispd}
      ││         └─{auditd}
```

```

|—avahi-daemon—avahi-daemon
|—bluetoothd
|—chronyd
|—colord—2*[{colord}]
|—crond
|—cupsd
|—2*[dbus-daemon—{dbus-daemon}]
|—dbus-launch
|—dconf-service—2*[{dconf-service}]
|—evolution-addre—4*[{evolution-addre}]
|—evolution-calen—4*[{evolution-calen}]
|—evolution-sourc—2*[{evolution-sourc}]
|—firewalld—{firewalld}
|—gconfd-2
|—gdm—gdm-simple-slav—Xorg
|   |
|   |—gdm-session-wor—gnome-session—abrt-applet—{abrt-applet}
|   |   |
|   |   |—gnome-settings—4*[{gnome-settings-}]
|   |   |—gnome-shell—firefox—42*[{firefox}]
|   |   |   |—gedit—4*[{gedit}]
|   |   |   |—5*[{gnome-shell}]
|   |   |—seapplet
|   |   |—ssh-agent
|   |   |—tracker-miner-f—4*[{tracker-miner-f}]
|   |   |—3*[{gnome-session}]
|   |   |—2*[{gdm-session-wor}]
|   |   |—2*[{gdm-simple-slav}]
|   |   |—2*[{gdm}]
|   |—gnome-keyring-d—4*[{gnome-keyring-d}]
|   |—gnome-shell-cal—4*[{gnome-shell-cal}]
|   |—gnome-terminal—bash—su—bash—pstree
|   |   |—gnome-pty-helpe
|   |   |—3*[{gnome-terminal-}]

```

...

## Options de la commande pstree

```
[root@centos7 ~]# 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
-Z,
--security-context    show SELinux security contexts
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@centos7 ~]# top
```

```
top - 16:28:28 up 21 min,  2 users,  load average: 1.50, 1.21, 0.86
Tasks: 160 total,   3 running, 157 sleeping,   0 stopped,   0 zombie
%Cpu(s):  8.3 us,   2.7 sy,   0.0 ni, 89.0 id,   0.0 wa,   0.0 hi,   0.0 si,   0.0 st
KiB Mem : 1791624 total,  114728 free,   697212 used,   979684 buff/cache
KiB Swap: 3071996 total, 3071996 free,         0 used.  921100 avail Mem
```

| PID  | USER    | PR | NI  | VIRT    | RES    | SHR   | S | %CPU | %MEM | TIME+   | COMMAND         |
|------|---------|----|-----|---------|--------|-------|---|------|------|---------|-----------------|
| 3249 | trainee | 20 | 0   | 1595324 | 235224 | 40572 | S | 13.6 | 13.1 | 3:43.96 | gnome-shell     |
| 1410 | root    | 20 | 0   | 336640  | 50172  | 8980  | R | 4.4  | 2.8  | 1:15.09 | Xorg            |
| 3756 | trainee | 20 | 0   | 626148  | 19044  | 12284 | S | 0.9  | 1.1  | 0:06.53 | gnome-terminal- |
| 3883 | trainee | 20 | 0   | 1049588 | 208252 | 49496 | S | 0.9  | 11.6 | 0:20.22 | firefox         |
| 4904 | root    | 20 | 0   | 130024  | 1780   | 1240  | R | 0.6  | 0.1  | 0:00.09 | top             |
| 3    | root    | 20 | 0   | 0       | 0      | 0     | S | 0.3  | 0.0  | 0:01.18 | ksoftirqd/0     |
| 525  | dbus    | 20 | 0   | 38480   | 3104   | 1444  | S | 0.3  | 0.2  | 0:01.53 | dbus-daemon     |
| 596  | root    | 20 | 0   | 338392  | 1072   | 756   | S | 0.3  | 0.1  | 0:01.03 | VBoxService     |
| 1    | root    | 20 | 0   | 59600   | 7028   | 3968  | S | 0.0  | 0.4  | 0:03.35 | systemd         |
| 2    | root    | 20 | 0   | 0       | 0      | 0     | S | 0.0  | 0.0  | 0:00.00 | kthreadd        |
| 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  | 0:00.17 | kworker/u2:0    |
| 7    | root    | rt | 0   | 0       | 0      | 0     | S | 0.0  | 0.0  | 0:00.00 | migration/0     |
| 8    | root    | 20 | 0   | 0       | 0      | 0     | S | 0.0  | 0.0  | 0:00.00 | rcu_bh          |
| 9    | root    | 20 | 0   | 0       | 0      | 0     | S | 0.0  | 0.0  | 0:00.00 | rcuob/0         |
| 10   | root    | 20 | 0   | 0       | 0      | 0     | S | 0.0  | 0.0  | 0:01.38 | rcu_sched       |
| 11   | root    | 20 | 0   | 0       | 0      | 0     | R | 0.0  | 0.0  | 0:02.55 | rcuos/0         |
| ...  |         |    |     |         |        |       |   |      |      |         |                 |

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

## Help for Interactive Commands - procps-ng version 3.3.10

Window 1:Def: Cumulative mode Off. System: Delay 20.0 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'/'0' other criteria  
n,#,^0 . Set: 'n'/'#' max tasks displayed; Show: Ctrl+'0' 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@centos7 ~]# top
```

```
...
top - 16:31:05 up 24 min,  2 users,  load average: 0.90, 1.10, 0.87
Tasks: 161 total,   3 running, 158 sleeping,   0 stopped,   0 zombie
%Cpu(s): 11.3 us,  7.2 sy,   0.0 ni, 81.5 id,   0.0 wa,   0.0 hi,   0.0 si,   0.0 st
KiB Mem : 1791624 total,  113084 free,  698800 used,  979740 buff/cache
KiB Swap: 3071996 total, 3071996 free,      0 used.  919492 avail Mem
Change delay from 3.0 to 1
...
```

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

```
[root@centos7 ~]# top
...
top - 16:32:41 up 26 min,  2 users,  load average: 0.65, 0.96, 0.84
Tasks: 161 total,   2 running, 159 sleeping,   0 stopped,   0 zombie
%Cpu(s):  2.9 us,  1.0 sy,   0.0 ni, 96.1 id,   0.0 wa,   0.0 hi,   0.0 si,   0.0 st
KiB Mem : 1791624 total,  114468 free,  697428 used,  979728 buff/cache
KiB Swap: 3071996 total, 3071996 free,      0 used.  920864 avail Mem
```

| PID  | USER    | PR | NI | VIRT    | RES    | SHR   | S | %CPU | %MEM | TIME+   | COMMAND         |
|------|---------|----|----|---------|--------|-------|---|------|------|---------|-----------------|
| 3249 | trainee | 20 | 0  | 1597372 | 237248 | 40572 | S | 4.1  | 13.2 | 5:10.83 | gnome-shell     |
| 3883 | trainee | 20 | 0  | 1049588 | 207680 | 49496 | S | 0.0  | 11.6 | 0:20.28 | firefox         |
| 1410 | root    | 20 | 0  | 336640  | 50172  | 8980  | S | 1.8  | 2.8  | 1:41.56 | Xorg            |
| 3873 | trainee | 20 | 0  | 1023212 | 30228  | 15840 | S | 0.0  | 1.7  | 0:21.83 | gedit           |
| 3359 | trainee | 20 | 0  | 1054704 | 24780  | 15860 | S | 0.0  | 1.4  | 0:01.06 | nautilus        |
| 3144 | trainee | 20 | 0  | 1000480 | 23484  | 14756 | S | 0.0  | 1.3  | 0:01.54 | gnome-settings- |
| 482  | root    | 20 | 0  | 329596  | 23200  | 6252  | S | 0.0  | 1.3  | 0:01.38 | firewalld       |
| 3756 | trainee | 20 | 0  | 626148  | 19044  | 12284 | S | 0.5  | 1.1  | 0:07.65 | gnome-terminal- |
| 3231 | trainee | 20 | 0  | 687500  | 18164  | 12164 | S | 0.0  | 1.0  | 0:00.09 | goa-daemon      |
| 506  | root    | 20 | 0  | 550176  | 16092  | 5612  | S | 0.0  | 0.9  | 0:01.02 | tuned           |
| 3418 | trainee | 20 | 0  | 564964  | 15820  | 4776  | S | 0.0  | 0.9  | 0:00.98 | tracker-store   |
| 1015 | root    | 20 | 0  | 103740  | 15652  | 3244  | S | 0.0  | 0.9  | 0:00.06 | dhclient        |
| 625  | polkitd | 20 | 0  | 518564  | 14932  | 4788  | S | 0.0  | 0.8  | 0:10.60 | polkitd         |
| 3477 | trainee | 20 | 0  | 450108  | 13964  | 9392  | S | 0.0  | 0.8  | 0:00.33 | abrt-applet     |

|      |         |    |    |        |       |      |   |     |     |         |                 |
|------|---------|----|----|--------|-------|------|---|-----|-----|---------|-----------------|
| 1217 | root    | 20 | 0  | 477048 | 13688 | 8796 | S | 0.0 | 0.8 | 0:00.37 | libvirtd        |
| 3395 | trainee | 20 | 0  | 894520 | 13624 | 8476 | S | 0.0 | 0.8 | 0:00.38 | evolution-calen |
| 3460 | trainee | 39 | 19 | 677116 | 12672 | 7388 | S | 0.0 | 0.7 | 0:00.32 | tracker-miner-f |

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

```
[root@centos7 ~]# top
...
top - 16:33:45 up 27 min,  2 users,  load average: 0.72, 0.94, 0.85
Tasks: 160 total,   3 running, 157 sleeping,   0 stopped,   0 zombie
%Cpu(s): 41.2 us,  7.2 sy,   0.0 ni, 51.3 id,   0.0 wa,   0.0 hi,   0.4 si,   0.0 st
KiB Mem : 1791624 total,  109348 free,   702524 used,   979752 buff/cache
KiB Swap: 3071996 total, 3071996 free,         0 used.  915768 avail Mem

   PID USER      PR  NI   VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
  3249 trainee   20   0 1597372 237248  40572 S   51.3  13.2   5:31.72 gnome-shell
  1410 root       20   0  347244  57824   8980 R   11.5   3.2   1:48.83 Xorg
  3873 trainee   20   0 1023212  30228  15840 S    1.6   1.7   0:24.10 gedit
  3756 trainee   20   0  626292  19044  12284 S    0.5   1.1   0:08.03 gnome-terminal-
  4904 root       20   0  130024   1780   1240 R    0.2   0.1   0:00.30 top
    596 root       20   0  338392   1072    756 S    0.2   0.1   0:01.28 VBoxService
  5080 root       20   0         0        0        0 S    0.2   0.0   0:00.23 kworker/0:1
```

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

## Options de la commande top

```
[root@centos7 ~]# 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 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 &
```



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@centos7 ~]# sleep 9999 &
[1] 9870
[root@centos7 ~]# jobs -l
[1]+  9870 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@centos7 ~]# 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@centos7 ~]# bg %2
[2]+  sleep 1234 &
[root@centos7 ~]# jobs -l
[1]-  9870 Running              sleep 9999 &
[2]+  9930 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@centos7 ~]# kill -stop %2
[root@centos7 ~]# jobs -l
[1]-  9870 Running              sleep 9999 &
[2]+  9930 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@centos7 ~]# kill -cont %2
[root@centos7 ~]# jobs -l
[1]-  9870 Running              sleep 9999 &
[2]+  9930 Running              sleep 1234 &
```

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

```
[root@centos7 ~]# kill -stop %2
[root@centos7 ~]# jobs -l
[1]-  9870 Running                sleep 9999 &
[2]+  9930 Stopped (signal)       sleep 1234
[root@centos7 ~]# fg %2
sleep 1234
^C
[root@centos7 ~]#
```



**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@centos7 ~]# 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@centos7 ~]# jobs -l  
[1]+  9870 Running                  sleep 9999 &  
[root@centos7 ~]# wait %1  
^C  
[root@centos7 ~]# jobs -l  
[1]+  9870 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 nohup

Cette commande permet à un processus de poursuivre son exécution après la déconnexion. Un processus enfant meurt quand le processus parent meurt 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@centos7 ~]# 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/>>

For complete documentation, run: 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@centos7 ~]# 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 <b>kill</b> . Le processus se termine normalement |

### Options de la commande kill

```

[root@centos7 ~]# 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@centos ~]# pkill -HUP rsyslogd
```

## Options de la commande pkill

```
[root@centos7 ~]# 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 <PGID,...> 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 uptime

Cette commande nous indique l'heure actuelle, la durée depuis laquelle le système fonctionne, le nombre d'utilisateurs actuellement connectés et la charge système moyenne pour les dernières 1 minute, 5 minutes et 15 minutes :

```
[root@centos7 ~]# uptime
19:18:57 up 1 day,  8:30,  1 user,  load average: 0.10, 0.09, 0.07
```

## Options de la commande



**A faire** : Utilisez l'option **-help** de la commande **uptime** pour visualiser les options de la commande.

## La Commande screen

La commande **screen** est un « multiplexeur de terminaux » permettant d'ouvrir jusqu'à 10 ( numérotés de 0 à 9 ) terminaux dans une même console, de passer de l'un à l'autre et de les récupérer plus tard.

La commande **screen** n'est pas installée par défaut sous RHEL/CentOS 7. Installez donc le paquet du même nom que la commande :

```
[root@centos7 ~]# which screen
/usr/bin/which: no screen in (/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/root/bin)
[root@centos7 ~]# yum install screen
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: ftp.rezopole.net
 * extras: centos.mirror.ate.info
 * updates: centos.mirror.fr.planethoster.net
Resolving Dependencies
--> Running transaction check
---> Package screen.x86_64 0:4.1.0-0.23.20120314git3c2946.el7_2 will be installed
--> Finished Dependency Resolution
```

Dependencies Resolved

```
=====
Package      Arch      Version                               Repository  Size
=====
```

Installing:

```
screen      x86_64      4.1.0-0.23.20120314git3c2946.el7_2      updates      552 k

Transaction Summary
=====
Install 1 Package

Total download size: 552 k
Installed size: 914 k
Is this ok [y/d/N]: y
Downloading packages:
screen-4.1.0-0.23.20120314git3c2946.el7_2.x86_64.rpm      | 552 kB  00:00:01
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : screen-4.1.0-0.23.20120314git3c2946.el7_2.x86_64      1/1
  Verifying  : screen-4.1.0-0.23.20120314git3c2946.el7_2.x86_64      1/1

Installed:
screen.x86_64 0:4.1.0-0.23.20120314git3c2946.el7_2

Complete!
[root@centos7 ~]# which screen
/bin/screen
```

Créez maintenant une session avec screen :

```
[root@centos7 ~]# screen -S mysession
```



**Important** - Vous verrez en tant que titre de votre terminal la chaîne **[screen 0: root@centos7:~]**.

Appuyez maintenant sur **CTRL A** puis relachez la touche **A** et appuyez sur la touche **C** pour créer un deuxième screen imbriqué dans la même session.



**Important** - Vous verrez en tant que titre de votre terminal la chaîne **[screen 1: root@centos7:~]**.

Pour repasser au premier screen, appuyez sur **CTRL A A**. Ceci permet de basculer entre les deux derniers screens.

Pour voir les screens actifs, utilisez la commande suivante :

```
[root@centos7 ~]# screen -ls
There is a screen on:
      25812.mysession (Attached)
1 Socket in /var/run/screen/S-root.
```

Dans votre screen, saisissez les commandes suivantes :

```
[root@centos7 ~]# sleep 9999 &
[1] 2272
[root@centos7 ~]# jobs
[1]+  Running                  sleep 9999 &
[root@centos7 ~]#
```

Pour détacher le screen, appuyez sur **CTRL A** puis relachez la touche **A** et appuyez sur la touche **D** :

```
[root@centos7 ~]# screen -S mysession
[detached from 25812.mysession]
[root@centos7 ~]#
```

Pour rattacher le screen, saisissez la commande suivante :

```
[root@centos7 ~]# screen -r
```

Utilisez la commande `jobs` pour vérifier si le processus créé par la commande `sleep` est toujours en cours de fonctionnement :

```
[root@centos7 ~]# jobs
[1]+  Running                  sleep 9999 &
```

Pour naviguer entre les screens il convient d'appuyer sur `CTRL A` puis relachez la touche `A` et appuyez sur la touche `N` ou d'appuyer sur `CTRL A` puis relachez la touche `A` et appuyez sur la touche `P`.

Détachez de nouveau le screen actuel en appuyant sur `CTRL A` puis en relachant la touche `A` et en appuyant sur la touche `D` :

```
[root@centos7 ~]# screen -S mysession
[detached from 25812.mysession]
[root@centos7 ~]#
```

Créez maintenant un autre screen, cette fois-ci, non imbriqué :

```
[root@centos7 ~]# screen -S mysession1
```

Constatez le résultat :

```
[root@centos7 ~]# screen -ls
There are screens on:
    14942.mysession1      (Attached)
    25812.mysession (Detached)
2 Sockets in /var/run/screen/S-root.
```

Ratachez maintenant le screen **mysession** :

```
[root@centos7 ~]# screen -r 25812
```

Constatez de nouveau le résultat :

```
[root@centos7 ~]# sleep 9999 &
```

```
[1] 2272
[root@centos7 ~]# jobs
[1]+  Running                  sleep 9999 &
[root@centos7 ~]# jobs
[1]+  Running                  sleep 9999 &
[root@centos7 ~]# screen -ls
There is a screen on:
      25812.mysession (Attached)
1 Socket in /var/run/screen/S-root.

[root@centos7 ~]# screen -ls
There are screens on:
      14942.mysession1      (Attached)
      25812.mysession (Attached)
2 Sockets in /var/run/screen/S-root.
```

## Options de la commande



**A faire** : Utilisez l'option **-help** de la commande **screen** pour visualiser les options de la commande.

<html>

Copyright © 2004-2017 Hugh Norris.<br><br> <a rel="license" href="http://creativecommons.org/licenses/by-nc-nd/3.0/fr/"></a><br />Ce(tte) oeuvre est mise à disposition selon les termes de la <a rel="license" href="http://creativecommons.org/licenses/by-nc-nd/3.0/fr/">Licence Creative Commons Attribution - Pas d'Utilisation Commerciale - Pas de Modification 3.0 France</a>.

</html>

---

---