Dernière mise-à-jour : 2020/01/30 03:28

LSF112 - 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 :

```
SUSE12SP1:~ # cd /proc; ls -d [0-9]*
1     11899  1250   14326  1474  1499  1557  1671   191  21   23     255  261    268  27591  37    57    746  8
10    12     1259   14345  1475  15    1576  1689   192  210  23741  256  262    269  29987  377   6     748  814
1024  12127  12591  1466   1478  1509  16    17     199  212  23742  257  263    270  3      379   6144  749  825
11    12149  1267   1469   1482  1514  1657  17841  2    214  24     258  264    271  30082  4650  7     791  9
1154  12150  13     1470   1485  1515  1658  18     201  22   240    259  26579  272  357    470   739   795  94
1188  1230   14     1471   1492  1523  1661  19     208  226  25     26   266    273  36     5     745   796
```

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

```
SUSE12SP1:/proc # cd 1 ; ls -l
total 0
dr-xr-xr-x 2 root root 0 Oct 15 11:00 attr
-r-------- 1 root root 0 Oct 15 11:00 auxv
-r--r--r-- 1 root root 0 Oct 13 06:56 cgroup
--w------- 1 root root 0 Oct 15 11:00 clear_refs
-r--r--r-- 1 root root 0 Oct 13 06:56 cmdline
-rw-r--r-- 1 root root 0 Oct 13 06:56 comm
-rw-r--r-- 1 root root 0 Oct 15 11:00 coredump_filter
-r--r--r-- 1 root root 0 Oct 15 11:00 cpuset
lrwxrwxrwx 1 root root 0 Oct 15 11:00 cwd -> /
-r-------- 1 root root 0 Oct 15 11:00 environ
lrwxrwxrwx 1 root root 0 Oct 13 06:56 exe -> /usr/lib/systemd/systemd
dr-x------ 2 root root 0 Oct 13 06:56 fd
dr-x------ 2 root root 0 Oct 15 11:00 fdinfo
-rw-r--r-- 1 root root 0 Oct 15 11:00 gid_map
-r-------- 1 root root 0 Oct 15 11:00 io
-r-------- 1 root root 0 Oct 15 11:00 kgr_in_progress
-r--r--r-- 1 root root 0 Oct 15 11:00 latency
-r--r--r-- 1 root root 0 Oct 15 11:00 limits
-rw-r--r-- 1 root root 0 Oct 13 06:56 loginuid
-r--r--r-- 1 root root 0 Oct 13 06:56 maps
-rw------- 1 root root 0 Oct 15 11:00 mem
-r--r--r-- 1 root root 0 Oct 13 06:56 mountinfo
-r--r--r-- 1 root root 0 Oct 15 11:00 mounts
-r-------- 1 root root 0 Oct 15 11:00 mountstats
dr-xr-xr-x 5 root root 0 Oct 13 06:56 net
dr-x--x--x 2 root root 0 Oct 15 11:00 ns
-r--r--r-- 1 root root 0 Oct 15 11:00 numa_maps
-rw-r--r-- 1 root root 0 Oct 15 11:00 oom_adj
-r--r--r-- 1 root root 0 Oct 15 11:00 oom_score
-rw-r--r-- 1 root root 0 Oct 13 06:56 oom_score_adj
-r--r--r-- 1 root root 0 Oct 15 11:00 pagemap
-r--r--r-- 1 root root 0 Oct 15 11:00 personality
```

```
-rw-r--r-- 1 root root 0 Oct 15 11:00 projid_map
lrwxrwxrwx 1 root root 0 Oct 13 06:56 root -> /
-rw-r--r-- 1 root root 0 Oct 15 11:00 sched
-r--r--r-- 1 root root 0 Oct 15 11:00 schedstat
-r--r--r-- 1 root root 0 Oct 13 06:56 sessionid
-rw-r--r-- 1 root root 0 Oct 15 11:00 setgroups
-r--r--r-- 1 root root 0 Oct 15 11:00 smaps
-r--r--r-- 1 root root 0 Oct 15 11:00 stack
-r--r--r-- 1 root root 0 Oct 13 06:56 stat
-r--r--r-- 1 root root 0 Oct 15 11:00 statm
-r--r--r-- 1 root root 0 Oct 13 06:56 status
-r--r--r-- 1 root root 0 Oct 15 11:00 syscall
dr-xr-xr-x 3 root root 0 Oct 13 06:56 task
-rw-r--r-- 1 root root 0 Oct 15 11:00 uid_map
-r--r--r-- 1 root root 0 Oct 15 11:00 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 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.

Les Commandes relatives aux Processus

La commande ps

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

```
SUSE12SP1:/proc/1 # cd ~
SUSE12SP1:~ # ps
  PID TTY          TIME CMD
14326 pts/1    00:00:00 su
14345 pts/1    00:00:00 bash
30394 pts/1    00:00:00 ps
```

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

```
SUSE12SP1:~ # ps -l
F S   UID   PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S     0 14326 12150  0  80   0 - 15933 wait   pts/1    00:00:00 su
4 S     0 14345 14326  0  80   0 -  3676 wait   pts/1    00:00:00 bash
0 R     0 30486 14345  0  80   0 -  6299 -      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 :

```
SUSE12SP1:~ # ps lx | more
F   UID   PID  PPID PRI  NI    VSZ   RSS WCHAN  STAT TTY        TIME COMMAND
4     0     1     0  20   0  34044  4516 SyS_ep Ss   ?          0:07 /usr/lib/systemd/systemd --switched-root --
system --deserialize 2
1
1     0     2     0  20   0      0     0 kthrea S    ?          0:00 [kthreadd]
1     0     3     2  20   0      0     0 smpboo S    ?          0:11 [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    ?          0:01 [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_gp S    ?          0:06 [rcu_sched]
5     0    10     2 -100  -      0     0 smpboo 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   0 -20      0     0 rescue S<   ?          0:00 [perf]
1     0    15     2   0 -20      0     0 rescue S<   ?          0:00 [writeback]
1     0    16     2   0 -20      0     0 rescue S<   ?          0:00 [kintegrityd]
1     0    17     2   0 -20      0     0 rescue S<   ?          0:00 [bioset]
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 [kblockd]
1     0    21     2   0 -20      0     0 rescue S<   ?          0:00 [kgraft]
1     0    22     2  20   0      0     0 watchd S    ?          0:00 [khungtaskd]
1     0    23     2  20   0      0     0 kswapd S    ?          0:00 [kswapd0]
1     0    24     2  25   5      0     0 ksm_sc SN   ?          0:00 [ksmd]
1     0    25     2  39  19      0     0 khugep SN   ?          0:00 [khugepaged]
1     0    26     2  20   0      0     0 fsnoti S    ?          0:00 [fsnotify_mark]
1     0    36     2   0 -20      0     0 rescue S<   ?          0:00 [kthrotld]
1     0    37     2   0 -20      0     0 rescue S<   ?          0:00 [kpsmoused]
1     0    57     2   0 -20      0     0 rescue S<   ?          0:00 [deferwq]
1     0    94     2  20   0      0     0 kaudit S    ?          0:00 [kauditd]
1     0   191     2   0 -20      0     0 rescue S<   ?          0:00 [ata_sff]
1     0   192     2  20   0      0     0 hub_th S    ?          0:00 [khubd]
1     0   199     2  20   0      0     0 scsi_e S    ?          0:00 [scsi_eh_0]
1     0   201     2   0 -20      0     0 rescue S<   ?          0:00 [scsi_tmf_0]
1     0   208     2  20   0      0     0 scsi_e S    ?          0:00 [scsi_eh_1]
1     0   210     2   0 -20      0     0 rescue S<   ?          0:00 [scsi_tmf_1]
--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 :

```
SUSE12SP1:~ # ps aux | more
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
```

```
root         1  0.0  0.2  34044  4516 ?        Ss   03:47   0:07 /usr/lib/systemd/systemd --switched-root --
system --deserialize 21
root         2  0.0  0.0      0     0 ?        S    03:47   0:00 [kthreadd]
root         3  0.0  0.0      0     0 ?        S    03:47   0:11 [ksoftirqd/0]
root         5  0.0  0.0      0     0 ?        S<   03:47   0:00 [kworker/0:0H]
root         6  0.0  0.0      0     0 ?        S    03:47   0:01 [kworker/u2:0]
root         7  0.0  0.0      0     0 ?        S    03:47   0:00 [migration/0]
root         8  0.0  0.0      0     0 ?        S    03:47   0:00 [rcu_bh]
root         9  0.0  0.0      0     0 ?        S    03:47   0:06 [rcu_sched]
root        10  0.0  0.0      0     0 ?        S    03:47   0:00 [watchdog/0]
root        11  0.0  0.0      0     0 ?        S<   03:47   0:00 [khelper]
root        12  0.0  0.0      0     0 ?        S    03:47   0:00 [kdevtmpfs]
root        13  0.0  0.0      0     0 ?        S<   03:47   0:00 [netns]
root        14  0.0  0.0      0     0 ?        S<   03:47   0:00 [perf]
root        15  0.0  0.0      0     0 ?        S<   03:47   0:00 [writeback]
root        16  0.0  0.0      0     0 ?        S<   03:47   0:00 [kintegrityd]
root        17  0.0  0.0      0     0 ?        S<   03:47   0:00 [bioset]
root        18  0.0  0.0      0     0 ?        S<   03:47   0:00 [crypto]
root        19  0.0  0.0      0     0 ?        S<   03:47   0:00 [kblockd]
root        21  0.0  0.0      0     0 ?        S<   03:47   0:00 [kgraft]
root        22  0.0  0.0      0     0 ?        S    03:47   0:00 [khungtaskd]
root        23  0.0  0.0      0     0 ?        S    03:47   0:00 [kswapd0]
root        24  0.0  0.0      0     0 ?        SN   03:47   0:00 [ksmd]
root        25  0.0  0.0      0     0 ?        SN   03:47   0:00 [khugepaged]
root        26  0.0  0.0      0     0 ?        S    03:47   0:00 [fsnotify_mark]
root        36  0.0  0.0      0     0 ?        S<   03:47   0:00 [kthrotld]
root        37  0.0  0.0      0     0 ?        S<   03:47   0:00 [kpsmoused]
root        57  0.0  0.0      0     0 ?        S<   03:47   0:00 [deferwq]
root        94  0.0  0.0      0     0 ?        S    03:47   0:00 [kauditd]
root       191  0.0  0.0      0     0 ?        S<   03:47   0:00 [ata_sff]
root       192  0.0  0.0      0     0 ?        S    03:47   0:00 [khubd]
root       199  0.0  0.0      0     0 ?        S    03:47   0:00 [scsi_eh_0]
root       201  0.0  0.0      0     0 ?        S<   03:47   0:00 [scsi_tmf_0]
root       208  0.0  0.0      0     0 ?        S    03:47   0:00 [scsi_eh_1]
```

```
root       210  0.0  0.0      0     0 ?        S<   03:47   0:00 [scsi_tmf_1]
root       212  0.0  0.0      0     0 ?        S    03:47   0:00 [scsi_eh_2]
--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 :

```
SUSE12SP1:~ # 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 :

```
SUSE12SP1:~ # pgrep -u root sshd
```

```
1576
12127
```

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

```
SUSE12SP1:~ # pgrep -u root,trainee | more
1
2
3
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
21
22
23
24
25
26
36
37
57
94
```

```
191
192
199
201
208
210
212
214
--More--
```

Option de la commande pgrep

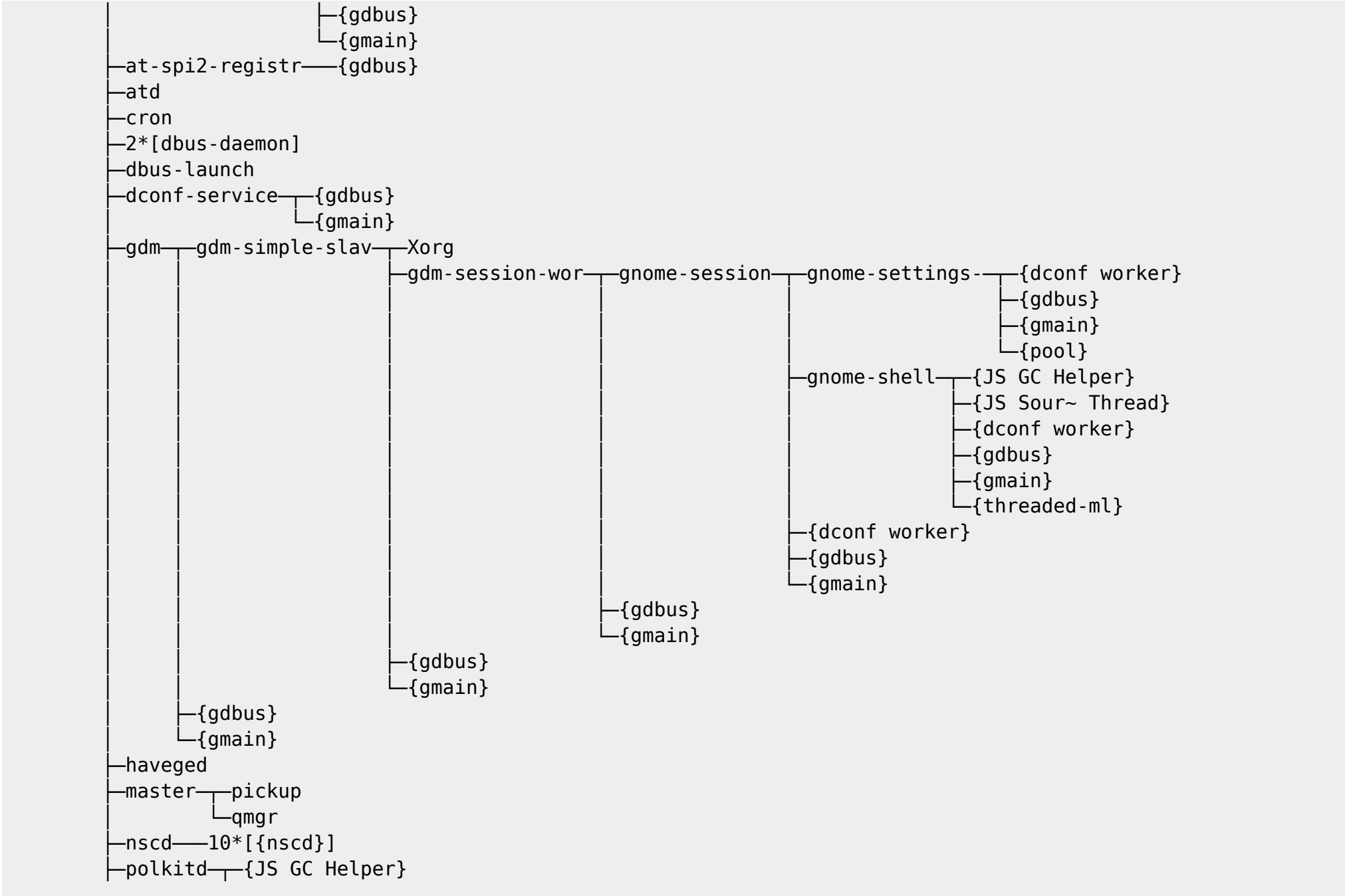
```
SUSE12SP1:~ # pgrep --help

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 :

```
SUSE12SP1:~ # pstree
systemd─┬─Remote Access───30*[{Remote Access}]
        ├─VBoxService─┬─{automount}
        │             ├─{control}
        │             ├─{cpuhotplug}
        │             ├─{memballoon}
        │             ├─{timesync}
        │             ├─{vminfo}
        │             └─{vmstats}
        ├─accounts-daemon─┬─{gdbus}
        │                 └─{gmain}
        ├─agetty
        ├─at-spi-bus-laun─┬─dbus-daemon
        │                 ├─{dconf worker}
```


├─{gdbus}
└─{gmain}
├─at-spi2-registr───{gdbus}
├─atd
├─cron
├─2*[dbus-daemon]
├─dbus-launch
├─dconf-service─┬─{gdbus}
└─{gmain}
├─gdm─┬─gdm-simple-slav─┬─Xorg
├─gdm-session-wor─┬─gnome-session─┬─gnome-settings-─┬─{dconf worker}
├─{gdbus}
├─{gmain}
└─{pool}
├─gnome-shell─┬─{JS GC Helper}
├─{JS Sour~ Thread}
├─{dconf worker}
├─{gdbus}
├─{gmain}
└─{threaded-ml}
├─{dconf worker}
├─{gdbus}
└─{gmain}
├─{gdbus}
└─{gmain}
├─{gdbus}
└─{gmain}
├─{gdbus}
└─{gmain}
├─haveged
├─master─┬─pickup
└─qmgr
├─nscd───10*[{nscd}]
├─polkitd─┬─{JS GC Helper}

```
        │          ├─{JS Sour~ Thread}
        │          ├─{gdbus}
        │          ├─{gmain}
        │          └─{runaway-killer-}
        ├─pulseaudio─┬─{alsa-sink-Intel}
        │            └─{alsa-source-Int}
        ├─rsyslogd─┬─{in:imklog}
        │          ├─{in:immark}
        │          ├─{in:imuxsock}
        │          └─{rs:main Q:Reg}
        ├─rtkit-daemon───2*[{rtkit-daemon}]
        ├─sh───sleep
        ├─sshd───sshd───sshd───bash───su───bash───pstree
        ├─2*[systemd───(sd-pam)]
        ├─systemd-journal
        ├─systemd-logind
        ├─systemd-udevd
        ├─upowerd─┬─{gdbus}
        │         └─{gmain}
        ├─wickedd
        ├─wickedd-auto4
        ├─wickedd-dhcp4
        ├─wickedd-dhcp6
        └─wickedd-nanny
```

Options de la commande pstree

```
SUSE12SP1:~ # 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     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 :

```
SUSE12SP1:~ # top

top - 11:06:25 up  7:19,  2 users,  load average: 0.07, 0.07, 0.11
Tasks: 115 total,   1 running, 114 sleeping,   0 stopped,   0 zombie
%Cpu(s):  1.2 us,  1.2 sy,  0.0 ni, 97.7 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem:   1686796 total,  1320588 used,   366208 free,      876 buffers
KiB Swap:  2103292 total,        0 used,  2103292 free.  1026148 cached Mem
```

```
  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
 1689 root      20   0 1025844  64800  10168 S 3.769 3.842  16:20.15 Remote Access
31729 root      20   0   14044   1504   1060 R 0.665 0.089   0:00.29 top
11899 root      20   0       0      0      0 S 0.222 0.000   0:05.23 kworker/0:1
12149 trainee   20   0   87676   1792    900 S 0.222 0.106   0:00.64 sshd
    1 root      20   0   34044   4516   2156 S 0.000 0.268   0:07.62 systemd
    2 root      20   0       0      0      0 S 0.000 0.000   0:00.01 kthreadd
    3 root      20   0       0      0      0 S 0.000 0.000   0:12.00 ksoftirqd/0
    5 root       0 -20       0      0      0 S 0.000 0.000   0:00.00 kworker/0:0H
    6 root      20   0       0      0      0 S 0.000 0.000   0:01.97 kworker/u2:0
    7 root      rt   0       0      0      0 S 0.000 0.000   0:00.00 migration/0
    8 root      20   0       0      0      0 S 0.000 0.000   0:00.00 rcu_bh
    9 root      20   0       0      0      0 S 0.000 0.000   0:06.16 rcu_sched
   10 root      rt   0       0      0      0 S 0.000 0.000   0:00.97 watchdog/0
   11 root       0 -20       0      0      0 S 0.000 0.000   0:00.00 khelper
   12 root      20   0       0      0      0 S 0.000 0.000   0:00.00 kdevtmpfs
   13 root       0 -20       0      0      0 S 0.000 0.000   0:00.00 netns
   14 root       0 -20       0      0      0 S 0.000 0.000   0:00.00 perf
   15 root       0 -20       0      0      0 S 0.000 0.000   0:00.00 writeback
   16 root       0 -20       0      0      0 S 0.000 0.000   0:00.00 kintegrityd
   17 root       0 -20       0      0      0 S 0.000 0.000   0:00.00 bioset
   18 root       0 -20       0      0      0 S 0.000 0.000   0:00.00 crypto
   19 root       0 -20       0      0      0 S 0.000 0.000   0:00.00 kblockd
   21 root       0 -20       0      0      0 S 0.000 0.000   0:00.00 kgraft
   22 root      20   0       0      0      0 S 0.000 0.000   0:00.00 khungtaskd
   23 root      20   0       0      0      0 S 0.000 0.000   0:00.00 kswapd0
   24 root      25   5       0      0      0 S 0.000 0.000   0:00.00 ksmd
   25 root      39  19       0      0      0 S 0.000 0.000   0:00.40 khugepaged
   26 root      20   0       0      0      0 S 0.000 0.000   0:00.00 fsnotify_mark
   36 root       0 -20       0      0      0 S 0.000 0.000   0:00.00 kthrotld
...
```

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 3.0 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,O . Filter by: 'u'/'U' effective/any user; 'o'/'O' other criteria
  n,#,^O  . 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 :

```
...
```

```
top - 11:07:32 up  7:20,  2 users,  load average: 0.02, 0.06, 0.10
Tasks: 113 total,   1 running, 112 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.7 us,  1.8 sy,  0.0 ni, 97.5 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem:   1686796 total,  1320620 used,   366176 free,      876 buffers
KiB Swap:  2103292 total,        0 used,  2103292 free.  1026228 cached Mem
Change delay from 3.0 to 1
...
```

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

```
...
top - 11:08:24 up  7:21,  2 users,  load average: 0.01, 0.05, 0.10
Tasks: 115 total,   1 running, 114 sleeping,   0 stopped,   0 zombie
%Cpu(s):  1.1 us,  0.8 sy,  0.0 ni, 98.1 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem:   1686796 total,  1320884 used,   365912 free,      876 buffers
KiB Swap:  2103292 total,        0 used,  2103292 free.  1026312 cached Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
 1509 gdm       20   0 1312532 150848  38056 S 0.000 8.943   1:23.78 gnome-shell
 1689 root      20   0 1025844  64800  10168 S 3.689 3.842  16:24.57 Remote Access
 1492 gdm       20   0  874936  20064  13140 S 0.000 1.189   0:03.40 gnome-settings-
 1250 root      20   0  188672  19320   8216 S 0.000 1.145   0:09.72 Xorg
 1267 polkitd   20   0  516572  17272   4940 S 0.000 1.024   0:01.36 polkitd
 1471 gdm       20   0  541948   6716   5232 S 0.000 0.398   0:00.77 gnome-session
 1466 root      20   0  232428   6572   3516 S 0.000 0.390   0:01.10 gdm-session-wor
 1230 root      20   0  279600   5936   3128 S 0.000 0.352   0:00.06 gdm-simple-slav
 1514 gdm       20   0  354580   5880   3644 S 0.000 0.349   0:00.59 pulseaudio
  357 root      20   0   35048   5720   5368 S 0.000 0.339   0:04.64 systemd-journal
 1188 root      20   0  198372   5564   2900 S 0.000 0.330   0:00.03 gdm
  379 root      20   0   12032   5520    676 S 0.000 0.327   0:07.23 haveged
 1499 root      20   0  223632   4900   3892 S 0.000 0.290   0:01.84 upowerd
    1 root      20   0   34044   4516   2156 S 0.000 0.268   0:07.64 systemd
  739 message+  20   0   44408   4128   1612 S 0.000 0.245   0:10.51 dbus-daemon
12127 root      20   0   87676   3960   3068 S 0.000 0.235   0:00.03 sshd
```

```
 1259 root      20   0  274852   3660   2940 S 0.000 0.217   0:01.32 accounts-daemon
14345 root      20   0   14704   3540   1764 S 0.000 0.210   0:00.46 bash
12150 trainee   20   0   14728   3496   1756 S 0.000 0.207   0:00.14 bash
 1485 gdm       20   0  122904   3360   2732 S 0.000 0.199   0:00.02 at-spi2-registr
  814 root      20   0   29596   3340   2224 S 0.000 0.198   0:00.06 wickedd
 1478 gdm       20   0  335320   3200   2676 S 0.000 0.190   0:00.03 at-spi-bus-laun
  746 root      20   0   29472   3176   2184 S 0.000 0.188   0:00.05 wickedd-dhcp4
  745 root      20   0   29472   3064   2080 S 0.000 0.182   0:00.04 wickedd-dhcp6
 1576 root      20   0   47108   3032   2412 S 0.000 0.180   0:00.02 sshd
  825 root      20   0   29500   3020   2048 S 0.000 0.179   0:00.03 wickedd-nanny
  748 root      20   0   29472   2920   1948 S 0.000 0.173   0:00.02 wickedd-auto4
23742 trainee   20   0   64260   2712     76 S 0.000 0.161   0:00.00 (sd-pam)
 1470 gdm       20   0   64164   2604     76 S 0.000 0.154   0:00.00 (sd-pam)
 1523 gdm       20   0  178104   2576   2152 S 0.000 0.153   0:00.01 dconf-service
...
```

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

```
...
top - 11:08:53 up  7:21,  2 users,  load average: 0.01, 0.04, 0.10
Tasks: 115 total,   1 running, 114 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.7 us,  1.6 sy,  0.0 ni, 97.6 id,  0.1 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem:   1686796 total,  1321032 used,   365764 free,      876 buffers
KiB Swap:  2103292 total,        0 used,  2103292 free.  1026348 cached Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
 1689 root      20   0 1025844  64800  10168 S 3.859 3.842  16:25.68 Remote Access
  379 root      20   0   12032   5520    676 S 0.221 0.327   0:07.26 haveged
  739 message+  20   0   44408   4128   1612 S 0.110 0.245   0:10.52 dbus-daemon
31729 root      20   0   14044   1504   1060 R 0.221 0.089   0:00.80 top
 1154 root      20   0  233004   1060    748 S 0.221 0.063   0:24.94 VBoxService
    3 root      20   0       0      0      0 S 0.110 0.000   0:12.08 ksoftirqd/0
    9 root      20   0       0      0      0 S 0.110 0.000   0:06.20 rcu_sched
```

```
11899 root      20   0       0      0      0 S 0.221 0.000   0:05.50 kworker/0:1
```

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

Options de la commande top

```
SUSE12SP1:~ # 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 &
```

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**.

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 :

```
SUSE12SP1:~ # sleep 9999 &
[1] 723
SUSE12SP1:~ # jobs -l
```

```
[1]+   723 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 a 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 :

```
SUSE12SP1:~ # 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 :

```
SUSE12SP1:~ # bg %2
[2]+ sleep 1234 &
SUSE12SP1:~ # jobs -l
[1]-   723 Running                 sleep 9999 &
[2]+   898 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 a 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**.

```
SUSE12SP1:~ # kill -stop %2
SUSE12SP1:~ # jobs -l
[1]-   723 Running                 sleep 9999 &
[2]+   898 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** :

```
SUSE12SP1:~ # kill -cont %2
SUSE12SP1:~ # jobs -l
[1]-   723 Running                 sleep 9999 &
[2]+   898 Running                 sleep 1234 &
```

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

```
SUSE12SP1:~ # kill -stop %2
SUSE12SP1:~ # jobs -l
[1]-   723 Running                 sleep 9999 &
[2]+   898 Stopped (signal)        sleep 1234
SUSE12SP1:~ # fg %2
sleep 1234
^C
SUSE12SP1:~ #
```

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

Options de la commande jobs

Les options de la commande jobs sont :

```
SUSE12SP1:~ # 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 :

```
SUSE12SP1:~ # jobs -l
[1]+   723 Running                 sleep 9999 &
SUSE12SP1:~ # wait %1
^C
SUSE12SP1:~ # jobs -l
[1]+   723 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 :

```
SUSE12SP1:~ # nice -n -20 sleep 1234
^Z
[2]+  Stopped                 nice -n -20 sleep 1234
SUSE12SP1:~ # ps lx | grep sleep
0     0   723 14345  20   0   4488   564 -      S    pts/1      0:00 sleep 9999
4     0  2251 14345   0 -20   4488   560 signal T<   pts/1      0:00 sleep 1234
0     0  2373  1557  20   0   4488   564 -      S    ?          0:00 sleep 5
0     0  2375 14345  20   0   9240   928 pipe_w S+   pts/1      0:00 grep --color=auto sleep
SUSE12SP1:~ # nice -n 19 sleep 5678
^Z
[3]+  Stopped                 nice -n 19 sleep 5678
SUSE12SP1:~ # ps lx | grep sleep
0     0   723 14345  20   0   4488   564 -      S    pts/1      0:00 sleep 9999
4     0  2251 14345   0 -20   4488   560 signal T<   pts/1      0:00 sleep 1234
0     0  2420 14345  39  19   4488   564 signal TN   pts/1      0:00 sleep 5678
0     0  2494  2487  20   0   4488   560 -      S    ?          0:00 sleep 1
0     0  2508 14345  20   0   9240   928 pipe_w S+   pts/1      0:00 grep --color=auto 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 :

```
SUSE12SP1:~ # 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/>
Report nice translation bugs to <http://translationproject.org/team/>
For complete documentation, run: 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.

```
SUSE12SP1:~ # jobs -l
[1]    723 Running                 sleep 9999 &
[2]-  2251 Stopped                 nice -n -20 sleep 1234
[3]+  2420 Stopped                 nice -n 19 sleep 5678
```

```
SUSE12SP1:~ # bg %2
[2]- nice -n -20 sleep 1234 &
SUSE12SP1:~ # bg %3
[3]+ nice -n 19 sleep 5678 &
SUSE12SP1:~ # jobs -l
[1]    723 Running                 sleep 9999 &
[2]-  2251 Running                 nice -n -20 sleep 1234 &
[3]+  2420 Running                 nice -n 19 sleep 5678 &
SUSE12SP1:~ #
SUSE12SP1:~ # renice +5 2251
2251 (process ID) old priority -20, new priority 5
SUSE12SP1:~ # renice -5 2420
2420 (process ID) old priority 19, new priority -5
SUSE12SP1:~ # ps lx | grep sleep
0     0   723 14345  20   0   4488   564 -      S    pts/1      0:00 sleep 9999
4     0  2251 14345  25   5   4488   560 -      SN   pts/1      0:00 sleep 1234
0     0  2420 14345  15  -5   4488   564 -      S<   pts/1      0:00 sleep 5678
0     0  3211  1557  20   0   4488   560 -      S    ?          0:00 sleep 5
0     0  3213 14345  20   0   9240   932 pipe_w S+   pts/1      0:00 grep --color=auto 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 :

```
SUSE12SP1:~ # 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 :

```
SUSE12SP1:~ # 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/>
Report nohup translation bugs to <http://translationproject.org/team/>
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** :

```
SUSE12SP1:~ # 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 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

Options de la commande kill

```
SUSE12SP1:~ # 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 :

```
SUSE12SP1:~ # pkill -HUP rsyslogd
```

Options de la commande pkill

```
SUSE12SP1:~ # 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).
```

<html>

</html>
