

Version : **2023.01**

Dernière mise-à-jour : 2023/08/25 01:14

LCF507 - Gestion des Processus

Contenu du Module

- **LCF507 - Gestion des Processus**
 - Contenu du Module
 - Présentation
 - Les Types de Processus
 - LAB #1 - Les Commandes relatives aux Processus
 - 1.1 - La Commande ps
 - 1.2 - La Commande pgrep
 - 1.3 - La Commande pstree
 - 1.4 - La Commande top
 - 1.5 - Les Commandes fg, bg et jobs
 - 1.6 - La Commande wait
 - 1.7 - La Commande nice
 - 1.8 - La Commande renice
 - 1.9 - La Commande nohup
 - 1.10 - La Commande kill
 - 1.11 - La Commande pkill

Présentation

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

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

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

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

Saisissez la commande suivante :

```
[root@centos8 ~]# cd /proc; ls -d [0-9]*
1      1113 1158 1464 1590 1601 163  166 171 20 24 28 303 31  34 38 41 45 461 465 469 473
478 481 487 497 507 52  528 531 599 649 69  765 793 800 827 840 853 875 902
10     1125 12   1465 1594 162  1634 167 18 21 25 29 305 310 35 39 42 457 462 466 47  475
479 482 489 498 508 521 529 532 6   655 7   766 795 801 828 841 854 880 905
11     1126 13   15   16   1627 164  168 19 22 26 3  306 311 36 4  43 459 463 467 470 476 48
484 49  5   51   526 53   533 60  662 741 779 799 823 834 847 866 884 910
1111 1135 14   1585 1600 1628 165  17  2   23 27 30 309 32  37 40 44 46  464 468 472 477
480 485 496 50  518 527 530 534 632 666 751 791 8  825 837 850 874 9
```

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

```
[root@centos8 proc]# cd 1 ; ls -l
total 0
dr-xr-xr-x. 2 root root 0 Jun  3 09:01 attr
-rw-r--r--. 1 root root 0 Jun  3 09:02 autogroup
-r----- . 1 root root 0 Jun  3 09:02 auxv
-r--r--r--. 1 root root 0 Jun  3 09:01 cgroup
```

```
--w-----. 1 root root 0 Jun 3 09:02 clear_refs
-r--r--r--. 1 root root 0 Jun 3 09:01 cmdline
-rw-r--r--. 1 root root 0 Jun 3 09:01 comm
-rw-r--r--. 1 root root 0 Jun 3 09:02 coredump_filter
-r--r--r--. 1 root root 0 Jun 3 09:02 cpu_resctrl_groups
-r--r--r--. 1 root root 0 Jun 3 09:02 cpuset
lrwxrwxrwx. 1 root root 0 Jun 3 09:02 cwd -> /
-r-----. 1 root root 0 Jun 3 09:01 environ
lrwxrwxrwx. 1 root root 0 Jun 3 09:01 exe -> /usr/lib/systemd/systemd
dr-x-----. 2 root root 0 Jun 3 09:01 fd
dr-x-----. 2 root root 0 Jun 3 09:01 fdinfo
-rw-r--r--. 1 root root 0 Jun 3 09:01 gid_map
-r-----. 1 root root 0 Jun 3 09:02 io
-r--r--r--. 1 root root 0 Jun 3 09:02 limits
-rw-r--r--. 1 root root 0 Jun 3 09:01 loginuid
dr-x-----. 2 root root 0 Jun 3 09:02 map_files
-r--r--r--. 1 root root 0 Jun 3 09:01 maps
-rw-----. 1 root root 0 Jun 3 09:02 mem
-r--r--r--. 1 root root 0 Jun 3 09:01 mountinfo
-r--r--r--. 1 root root 0 Jun 3 09:01 mounts
-r-----. 1 root root 0 Jun 3 09:02 mountstats
dr-xr-xr-x. 6 root root 0 Jun 3 09:01 net
dr-x--x--x. 2 root root 0 Jun 3 09:02 ns
-r--r--r--. 1 root root 0 Jun 3 09:02 numa_maps
-rw-r--r--. 1 root root 0 Jun 3 09:02 oom_adj
-r--r--r--. 1 root root 0 Jun 3 09:02 oom_score
-rw-r--r--. 1 root root 0 Jun 3 09:02 oom_score_adj
-r-----. 1 root root 0 Jun 3 09:02 pagemap
-r-----. 1 root root 0 Jun 3 09:02 patch_state
-r-----. 1 root root 0 Jun 3 09:02 personality
-rw-r--r--. 1 root root 0 Jun 3 09:02 projid_map
lrwxrwxrwx. 1 root root 0 Jun 3 09:01 root -> /
-rw-r--r--. 1 root root 0 Jun 3 09:01 sched
-r--r--r--. 1 root root 0 Jun 3 09:02 schedstat
```

```
-r--r--r--. 1 root root 0 Jun  3 09:01 sessionid
-rw-r--r--. 1 root root 0 Jun  3 09:01 setgroups
-r--r--r--. 1 root root 0 Jun  3 09:02 smaps
-r--r--r--. 1 root root 0 Jun  3 09:02 smaps_rollup
-r-----. 1 root root 0 Jun  3 09:02 stack
-r--r--r--. 1 root root 0 Jun  3 09:01 stat
-r--r--r--. 1 root root 0 Jun  3 09:02 statm
-r--r--r--. 1 root root 0 Jun  3 09:01 status
-r-----. 1 root root 0 Jun  3 09:02 syscall
dr-xr-xr-x. 3 root root 0 Jun  3 09:01 task
-r--r--r--. 1 root root 0 Jun  3 09:02 timers
-rw-rw-rw-. 1 root root 0 Jun  3 09:02 timerslack_ns
-rw-r--r--. 1 root root 0 Jun  3 09:01 uid_map
-r--r--r--. 1 root root 0 Jun  3 09:02 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.

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.

LAB #1 - Les Commandes relatives aux Processus

1.1 - La Commande ps

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

```
[root@centos8 ~]# cd ~
[root@centos8 ~]# ps
  PID TTY          TIME CMD
 1627 pts/0    00:00:00 su
 1634 pts/0    00:00:00 bash
 1690 pts/0    00:00:00 ps
```

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

```
[root@centos8 ~]# ps -l
 F S      UID        PID      PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
 4 S        0         1627      1601  0  80   0 - 48967 -          pts/0    00:00:00 su
 4 S        0         1634      1627  0  80   0 - 6911 -          pts/0    00:00:00 bash
 0 R        0         1698      1634  0  80   0 - 11360 -          pts/0    00:00:00 ps
```

On note dans cette sortie :

F	Drapeaux du processus. La valeur 4 indique que le processus utilise les privilèges de root
S	État du processus S (sleeping), R (In run queue), Z (zombie), N (low priority), D (uninterruptible sleep), T (Traced)
UID	Numéro de l'Utilisateur
PID	Numéro Unique de Processus
PPID	PID du processus parent
C	Facteur de priorité du processus
PRI	Priorité du processus
NI	La valeur de nice
ADDR	Adresse mémoire du processus
SZ	Utilisation de la mémoire virtuelle
WCHAN	Nom de la fonction du noyau dans laquelle le processus est endormi
TTY	Nom du terminal depuis lequel le processus a été lancé
TIME	Durée d'exécution du processus
CMD	Commande exécutée

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

```
[root@centos8 ~]# ps lx | more
F  UID      PID      PPID  PRI  NI       VSZ      RSS WCHAN  STAT TTY          TIME COMMAND
4    0         1         0   20    0  245540  14252 do_epo  Ss   ?           0:01 /usr/lib/systemd/systemd --switched-root
--system --deserialize 17
1    0         2         0   20    0         0         0 -      S    ?           0:00 [kthreadd]
1    0         3         2    0  -20         0         0 -      I<   ?           0:00 [rcu_gp]
1    0         4         2    0  -20         0         0 -      I<   ?           0:00 [rcu_par_gp]
1    0         5         2   20    0         0         0 -      I    ?           0:00 [kworker/0:0-events]
1    0         6         2    0  -20         0         0 -      I<   ?           0:00 [kworker/0:0H-kblockd]
1    0         7         2   20    0         0         0 -      I    ?           0:00 [kworker/0:1-ata_sff]
1    0         8         2   20    0         0         0 -      I    ?           0:00 [kworker/u8:0-events_unbound]
1    0         9         2    0  -20         0         0 -      I<   ?           0:00 [mm_percpu_wq]
1    0        10         2   20    0         0         0 -      S    ?           0:00 [ksoftirqd/0]
1    0        11         2   20    0         0         0 -      I    ?           0:00 [rcu_sched]
1    0        12         2  -100   -         0         0 -      S    ?           0:00 [migration/0]
5    0        13         2  -100   -         0         0 -      S    ?           0:00 [watchdog/0]
```

```

1      0      14      2  20  0      0      0 -      S  ?      0:00 [cpuhp/0]
1      0      15      2  20  0      0      0 -      S  ?      0:00 [cpuhp/1]
5      0      16      2 -100 -      0      0 -      S  ?      0:00 [watchdog/1]
1      0      17      2 -100 -      0      0 -      S  ?      0:00 [migration/1]
1      0      18      2  20  0      0      0 -      S  ?      0:00 [ksoftirqd/1]
1      0      19      2  20  0      0      0 -      I  ?      0:00 [kworker/1:0-memcg_kmem_cache]
1      0      20      2   0 -20      0      0 -      I< ?      0:00 [kworker/1:0H]
1      0      21      2  20  0      0      0 -      S  ?      0:00 [cpuhp/2]
5      0      22      2 -100 -      0      0 -      S  ?      0:00 [watchdog/2]
--More--

```

On note dans cette sortie certaines informations supplémentaires :

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

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

```

[root@centos8 ~]# ps aux | more
USER          PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.6  0.3 245540 14252 ?        Ss   09:01   0:01 /usr/lib/systemd/systemd --switched-root --
system --deserialize 17
root           2  0.0  0.0      0      0 ?        S    09:01   0:00 [kthreadd]
root           3  0.0  0.0      0      0 ?        I<   09:01   0:00 [rcu_gp]
root           4  0.0  0.0      0      0 ?        I<   09:01   0:00 [rcu_par_gp]
root           5  0.0  0.0      0      0 ?        I    09:01   0:00 [kworker/0:0-events]
root           6  0.0  0.0      0      0 ?        I<   09:01   0:00 [kworker/0:0H-kblockd]
root           7  0.1  0.0      0      0 ?        I    09:01   0:00 [kworker/0:1-ata_sff]
root           8  0.0  0.0      0      0 ?        I    09:01   0:00 [kworker/u8:0-events_unbound]
root           9  0.0  0.0      0      0 ?        I<   09:01   0:00 [mm_percpu_wq]
root          10  0.0  0.0      0      0 ?        S    09:01   0:00 [ksoftirqd/0]
root          11  0.0  0.0      0      0 ?        I    09:01   0:00 [rcu_sched]
root          12  0.0  0.0      0      0 ?        S    09:01   0:00 [migration/0]

```

```

root      13  0.0  0.0      0   0 ?        S    09:01   0:00 [watchdog/0]
root      14  0.0  0.0      0   0 ?        S    09:01   0:00 [cpuhp/0]
root      15  0.0  0.0      0   0 ?        S    09:01   0:00 [cpuhp/1]
root      16  0.0  0.0      0   0 ?        S    09:01   0:00 [watchdog/1]
root      17  0.0  0.0      0   0 ?        S    09:01   0:00 [migration/1]
root      18  0.0  0.0      0   0 ?        S    09:01   0:00 [ksoftirqd/1]
root      19  0.0  0.0      0   0 ?        I    09:01   0:00 [kworker/1:0-memcg_kmem_cache]
root      20  0.0  0.0      0   0 ?        I<   09:01   0:00 [kworker/1:0H]
root      21  0.0  0.0      0   0 ?        S    09:01   0:00 [cpuhp/2]
root      22  0.0  0.0      0   0 ?        S    09:01   0:00 [watchdog/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

Les options de cette commande sont :

```
[root@centos8 ~]# ps --help all
```

Usage:

```
ps [options]
```

Basic options:

```

-A, -e      all processes
-a          all with tty, except session leaders
a           all with tty, including other users
-d          all except session leaders
-N, --deselect  negate selection
r           only running processes
T           all processes on this terminal
x           processes without controlling ttys

```

Selection by list:

-C <command>	command name
-G, --Group <GID>	real group id or name
-g, --group <group>	session or effective group name
-p, p, --pid <PID>	process id
--ppid <PID>	parent process id
-q, q, --quick-pid <PID>	process id (quick mode)
-s, --sid <session>	session id
-t, t, --tty <tty>	terminal
-u, U, --user <UID>	effective user id or name
-U, --User <UID>	real user id or name

The selection options take as their argument either:

a comma-separated list e.g. '-u root,nobody' or

a blank-separated list e.g. '-p 123 4567'

Output formats:

-F	extra full
-f	full-format, including command lines
f, --forest	ascii art process tree
-H	show process hierarchy
-j	jobs format
j	BSD job control format
-l	long format
l	BSD long format
-M, Z	add security data (for SELinux)
-O <format>	preloaded with default columns
O <format>	as -O, with BSD personality
-o, o, --format <format>	user-defined format
s	signal format
u	user-oriented format
v	virtual memory format

```
X          register format
-y         do not show flags, show rss vs. addr (used with -l)
--context  display security context (for SELinux)
--headers  repeat header lines, one per page
--no-headers do not print header at all
--cols, --columns, --width <num>
            set screen width
--rows, --lines <num>
            set screen height
```

Show threads:

```
H          as if they were processes
-L         possibly with LWP and NLWP columns
-m, m      after processes
-T         possibly with SPID column
```

Miscellaneous options:

```
-c          show scheduling class with -l option
c          show true command name
e          show the environment after command
k, --sort  specify sort order as: [+|-]key[, [+|-]key[, ...]]
L          show format specifiers
n          display numeric uid and wchan
S, --cumulative include some dead child process data
-y         do not show flags, show rss (only with -l)
-V, V, --version display version information and exit
-w, w      unlimited output width

--help <simple|list|output|threads|misc|all>
            display help and exit
```

For more details see `ps(1)`.

1.2 - 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@centos8 ~]# pgrep -u root sshd
905
1585
```

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

```
[root@centos8 ~]# pgrep -u root,trainee | more
1
2
3
4
6
7
9
10
11
12
13
14
15
16
17
18
20
21
22
23
```

```
24
26
27
--More--
```

Les options de cette commande sont :

```
[root@centos8 ~]# 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
-i, --ignore-case	match case insensitively
-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)`

1.3 - La Commande `pstree`

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

```
[root@centos8 ~]# pstree
systemd—NetworkManager—2*[{NetworkManager}]
      |
      |—agetty
      |—atd
      |—auditd—sedispatch
      |           |
      |           |—2*[{auditd}]
      |—chronyd
      |—crond
      |—dbus-daemon—{dbus-daemon}
      |—dnsmasq—dnsmasq
      |—firewalld—{firewalld}
      |—gssproxy—5*[{gssproxy}]
      |—irqbalance—{irqbalance}
      |—ksmtuned—sleep
      |—lsmd
      |—mcelog
      |—polkitd—7*[{polkitd}]
      |—rngd—4*[{rngd}]
      |—rpcbind
```

```

|—rsyslogd—2*[{rsyslogd}]
|—smartd
|—sshd—sshd—sshd—bash—su—bash—pstree
|—sssd—sssd_be
|      |—sssd_nss
|—systemd—(sd-pam)
|—systemd-journal
|—systemd-logind
|—systemd-machine
|—systemd-udev
|—tuned—3*[{tuned}]

```

Les options de cette commande sont :

```

[root@centos8 ~]# pstree --help
pstree: unrecognized option '--help'
Usage: pstree [-acglpsStuZ] [ -h | -H PID ] [ -n | -N type ]
           [ -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 (cgroup, 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
-t, --thread-names   show full thread names
-T, --hide-threads   hide threads, show only processes
-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

```

1.4 - La Commande top

Cette commande indique les processus en mémoire :

```
[root@centos8 ~]# top
```

```

top - 09:10:02 up 8 min,  1 user,  load average: 0.05, 0.09, 0.08
Tasks: 144 total,  1 running, 143 sleeping,  0 stopped,  0 zombie
%Cpu(s):  0.1 us,  0.2 sy,  0.0 ni, 99.6 id,  0.0 wa,  0.1 hi,  0.1 si,  0.0 st
MiB Mem :  3737.7 total,  3249.1 free,   231.4 used,   257.2 buff/cache
MiB Swap:  2000.0 total,  2000.0 free,    0.0 used.  3280.0 avail Mem

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1	root	20	0	245540	14260	9308	S	0.2	0.4	0:01.64	systemd
827	polkitd	20	0	1764748	25376	17348	S	0.2	0.7	0:00.12	polkitd
834	dbus	20	0	64684	5476	4480	S	0.2	0.1	0:00.20	dbus-daemon
1789	root	20	0	65420	4464	3764	R	0.2	0.1	0:00.03	top
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_gp
4	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	rcu_par_gp

6	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/0:0H-kblockd
7	root	20	0	0	0	0	I	0.0	0.0	0:00.47	kworker/0:1-mm_percpu_wq
9	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_percpu_wq
10	root	20	0	0	0	0	S	0.0	0.0	0:00.00	ksoftirqd/0
11	root	20	0	0	0	0	I	0.0	0.0	0:00.09	rcu_sched
12	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	migration/0
13	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	watchdog/0
14	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/0
15	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/1
16	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	watchdog/1
17	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	migration/1
18	root	20	0	0	0	0	S	0.0	0.0	0:00.00	ksoftirqd/1
20	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/1:0H
21	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/2
22	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	watchdog/2
23	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	migration/2
24	root	20	0	0	0	0	S	0.0	0.0	0:00.00	ksoftirqd/2
26	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/2:0H-kblockd
27	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/3
28	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	watchdog/3
29	root	rt	0	0	0	0	S	0.0	0.0	0:00.00	migration/3
30	root	20	0	0	0	0	S	0.0	0.0	0:00.00	ksoftirqd/3
32	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kworker/3:0H-kblockd
35	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kdevtmpfs
36	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	netns
37	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kauditd
38	root	20	0	0	0	0	S	0.0	0.0	0:00.00	khungtaskd
39	root	20	0	0	0	0	S	0.0	0.0	0:00.00	oom_reaper
40	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	writeback
41	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kcompactd0
42	root	25	5	0	0	0	S	0.0	0.0	0:00.00	ksmd
43	root	39	19	0	0	0	S	0.0	0.0	0:00.00	khugepaged
44	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	crypto
45	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	kintegrityd

```

 46 root      0 -20      0      0      0 I   0.0   0.0   0:00.00 kblockd
 47 root      0 -20      0      0      0 I   0.0   0.0   0:00.00 blkcg_punt_bio
 48 root      0 -20      0      0      0 I   0.0   0.0   0:00.00 tpm_dev_wq
 49 root      0 -20      0      0      0 I   0.0   0.0   0:00.00 md
 50 root      0 -20      0      0      0 I   0.0   0.0   0:00.00 edac-poller
 51 root      rt  0       0      0      0 S   0.0   0.0   0:00.00 watchdogd
 53 root      0 -20      0      0      0 I   0.0   0.0   0:00.00 pm_wq
 69 root     20  0       0      0      0 S   0.0   0.0   0:00.00 kswapd0
162 root      0 -20      0      0      0 I   0.0   0.0   0:00.00 kthrotld
163 root      0 -20      0      0      0 I   0.0   0.0   0:00.00 acpi_thermal_pm
...

```

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

Help for Interactive Commands - procps-ng 3.3.15

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,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@centos8 ~]# top
...
top - 09:11:24 up 10 min,  1 user,  load average: 0.01, 0.07, 0.07
Tasks: 144 total,   1 running, 143 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,   0.1 sy,   0.0 ni, 99.7 id,   0.0 wa,   0.2 hi,   0.0 si,   0.0 st
MiB Mem :   3737.7 total,   3248.8 free,    231.7 used,    257.2 buff/cache
MiB Swap:   2000.0 total,   2000.0 free,     0.0 used.   3279.7 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@centos8 ~]# top
...
top - 09:12:07 up 10 min,  1 user,  load average: 0.00, 0.05, 0.06
Tasks: 146 total,   1 running, 145 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.1 us,   0.4 sy,   0.7 ni, 97.6 id,   0.2 wa,   0.9 hi,   0.1 si,   0.0 st
MiB Mem :   3737.7 total,   3192.8 free,    233.5 used,    311.4 buff/cache
MiB Swap:   2000.0 total,   2000.0 free,     0.0 used.   3277.3 avail Mem

  PID USER      PR  NI   VIRT    RES    SHR S  %CPU  %MEM    TIME+  COMMAND

```

```

 874 root      20    0   290088   40696   17960 S    0.0   1.1   0:00.54 firewallld
 875 root      20    0   224868   40472   38904 S    0.0   1.1   0:00.08 sssd_nss
 902 root      20    0   426324   33684   16668 S    0.0   0.9   0:00.25 tuned
 827 polkitd    20    0  1764748  25376   17348 S    0.0   0.7   0:00.12 polkitd
 884 root      20    0   391768   18828   16220 S    0.1   0.5   0:00.11 NetworkManager
 866 root      20    0   221424   15016   12284 S    0.0   0.4   0:00.09 sssd_be
 840 root      20    0   214800   14292   12260 S    0.0   0.4   0:00.03 sssd
   1 root      20    0   245540   14260    9308 S    0.1   0.4   0:01.66 systemd
1627 root      20    0   195868   11896   10336 S    0.0   0.3   0:00.02 su
 666 root      20    0   110416   11336    8292 S    0.0   0.3   0:00.30 systemd-udev
1585 root      20    0   163700   10636    9340 S    0.0   0.3   0:00.01 sshd
1590 trainee   20    0    94128    9872    8240 S    0.0   0.3   0:00.05 systemd
 880 root      20    0    96712    9780    7568 S    0.0   0.3   0:00.25 systemd-logind
 632 root      20    0    94036    9064    8016 S    0.0   0.2   0:00.28 systemd-journal
 837 root      20    0    83656    7004    6108 S    0.0   0.2   0:00.19 systemd-machine
 905 root      20    0    92288    6996    6100 S    0.0   0.2   0:00.01 sshd
 847 rngd       20    0   381308    6500    5656 S    0.0   0.2   0:15.38 rngd
 828 root      20    0    50640    5900    4600 S    0.0   0.2   0:00.03 smartd
1600 trainee   20    0   163700    5584    4284 S    0.0   0.1   0:00.04 sshd
1601 trainee   20    0    28312    5564    3664 S    0.0   0.1   0:00.02 bash
 834 dbus       20    0    64684    5476    4480 S    0.1   0.1   0:00.22 dbus-daemon
1634 root      20    0    27644    5428    3408 S    0.0   0.1   0:00.06 bash
 791 rpc       20    0    67140    5396    4672 S    0.0   0.1   0:00.01 rpcbind
1594 trainee   20    0   175840    5228     40 S    0.0   0.1   0:00.00 (sd-pam)
1113 root      20    0   209436    5036    3516 S    0.0   0.1   0:00.09 rsyslogd
 823 root      20    0   124908    4644    4080 S    0.0   0.1   0:00.03 irqbalance
1803 root      20    0    65420    4508    3808 R    0.0   0.1   0:00.03 top

```

...

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

```
[root@centos8 ~]# top
```

...

```
top - 09:13:01 up 11 min,  1 user,  load average: 0.00, 0.04, 0.06
```

```
Tasks: 145 total,  2 running, 143 sleeping,  0 stopped,  0 zombie
%Cpu(s):  0.1 us,  0.1 sy,  0.0 ni, 99.8 id,  0.0 wa,  0.1 hi,  0.0 si,  0.0 st
MiB Mem :  3737.7 total,  3192.5 free,   233.8 used,   311.4 buff/cache
MiB Swap:  2000.0 total,  2000.0 free,    0.0 used.  3276.9 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
827	polkitd	20	0	1764748	25376	17348	S	0.1	0.7	0:00.13	polkitd
1	root	20	0	245540	14260	9308	S	0.1	0.4	0:01.67	systemd
823	root	20	0	124908	4644	4080	S	0.1	0.1	0:00.04	irqbalance
1803	root	20	0	65420	4508	3808	R	0.2	0.1	0:00.08	top

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

Les options de cette commande sont :

```
[root@centos8 ~]# top --help
top: inappropriate '-help'
Usage:
  top -hv | -bcEHiOssl -d secs -n max -u|U user -p pid(s) -o field -w [cols]
```

1.5 - Les Commandes fg, bg et jobs

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

```
# sleep 9999 &
```



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

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

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

```
[root@centos8 ~]# sleep 9999 &
[1] 1865
[root@centos8 ~]# jobs -l
[1]+  1865 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@centos8 ~]# 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@centos8 ~]# bg %2
[2]+ sleep 1234 &
[root@centos8 ~]# jobs -l
[1]-  1865 Running                  sleep 9999 &
[2]+  1874 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@centos8 ~]# kill -stop %2
[root@centos8 ~]# jobs -l
[1]-  1865 Running                sleep 9999 &
[2]+  1874 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@centos8 ~]# kill -cont %2
[root@centos8 ~]# jobs -l
[1]-  1865 Running                sleep 9999 &
[2]+  1874 Running                sleep 1234 &
```

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

```
[root@centos8 ~]# kill -stop %2

[2]+  Stopped                    sleep 1234
[root@centos8 ~]# jobs -l
[1]-  1865 Running                sleep 9999 &
[2]+  1874 Stopped (signal)       sleep 1234
[root@centos8 ~]# fg %2
sleep 1234
^C
[root@centos8 ~]#
```



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

Les options de la commande jobs sont :

```
[root@centos8 ~]# help jobs
jobs: jobs [-lnprs] [jobspec ...] or jobs -x command [args]
  Display status of jobs.
  Lists the active jobs. JOBSPEC restricts output to that job.
  Without options, the status of all active jobs is displayed.
  Options:
    -l    lists process IDs in addition to the normal information
    -n    lists only processes that have changed status since the last
          notification
    -p    lists process IDs only
    -r    restrict output to running jobs
    -s    restrict output to stopped jobs
  If -x is supplied, COMMAND is run after all job specifications that
  appear in ARGS have been replaced with the process ID of that job's
  process group leader.
  Exit Status:
  Returns success unless an invalid option is given or an error occurs.
  If -x is used, returns the exit status of COMMAND
```

1.6 - 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@centos8 ~]# jobs -l
[1]+  1865 Running                  sleep 9999 &
[root@centos8 ~]# wait %1
^C
[root@centos8 ~]# jobs -l
[1]+  1865 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**.

1.7 - 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@centos8 ~]# nice -n -20 sleep 1234
^Z
[2]+  Stopped                  nice -n -20 sleep 1234
[root@centos8 ~]# ps lx | grep sleep
0      0      1865      1634  20    0    7284      816 hrtime S    pts/0      0:00 sleep 9999
0      0      1998       853  20    0    7284      832 hrtime S    ?          0:00 sleep 60
4      0      1999      1634   0 -20    7284      728 -      T<    pts/0      0:00 sleep 1234
0      0      2001      1634  20    0   12112     1072 -      S+    pts/0      0:00 grep --color=auto sleep
[root@centos8 ~]# nice -n 19 sleep 5678
^Z
[3]+  Stopped                  nice -n 19 sleep 5678
[root@centos8 ~]# ps lx | grep sleep
0      0      1865      1634  20    0    7284      816 hrtime S    pts/0      0:00 sleep 9999
4      0      1999      1634   0 -20    7284      728 -      T<    pts/0      0:00 sleep 1234
0      0      2008      1634  39  19    7284      800 -      TN    pts/0      0:00 sleep 5678
0      0      2016       853  20    0    7284      784 hrtime S    ?          0:00 sleep 60
0      0      2018      1634  20    0   12112     1076 -      S+    pts/0      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.

Les options de cette commande sont :

```
[root@centos8 ~]# 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: <https://www.gnu.org/software/coreutils/>
Full documentation at: <https://www.gnu.org/software/coreutils/nice>
or available locally via: info '(coreutils) nice invocation'
```

1.8 - 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@centos8 ~]# jobs -l
[1]  1865 Running          sleep 9999 &
[2]- 1999 Stopped          nice -n -20 sleep 1234
```

```
[3]+ 2008 Stopped                nice -n 19 sleep 5678
[root@centos8 ~]# bg %2
[2]- nice -n -20 sleep 1234 &
[root@centos8 ~]# bg %3
[3]+ nice -n 19 sleep 5678 &
[root@centos8 ~]# jobs -l
[1] 1865 Running                sleep 9999 &
[2]- 1999 Running                nice -n -20 sleep 1234 &
[3]+ 2008 Running                nice -n 19 sleep 5678 &
[root@centos8 ~]# renice +5 1999
1999 (process ID) old priority -20, new priority 5
[root@centos8 ~]# renice -5 2008
2008 (process ID) old priority 19, new priority -5
[root@centos8 ~]# ps lx | grep sleep
0    0    1865    1634  20    0   7284    816 hrtime S    pts/0    0:00 sleep 9999
4    0    1999    1634  25    5   7284    728 -        SN    pts/0    0:00 sleep 1234
0    0    2008    1634  15   -5   7284    800 -        S<    pts/0    0:00 sleep 5678
0    0    2061     853  20    0   7284    708 hrtime S    ?        0:00 sleep 60
0    0    2064    1634  20    0  12112   1088 -        S+    pts/0    0:00 grep --color=auto sleep
```



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

Les options de cette commande sont :

```
[root@centos8 ~]# renice --help
```

Usage:

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

Alter the priority of running processes.

Options:

```
-n, --priority <num>    specify the nice increment value
-p, --pid <id>          interpret argument as process ID (default)
-g, --pgrp <id>         interpret argument as process group ID
-u, --user <name>|<id>  interpret argument as username or user ID

-h, --help              display this help
-V, --version           display version
```

For more details see `renice(1)`.

1.9 - 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 est 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 &
```

Les options de cette commande sont :

```
[root@centos8 ~]# 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 an unreadable file.
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: <<https://www.gnu.org/software/coreutils/>>
Full documentation at: <<https://www.gnu.org/software/coreutils/nohup>>
or available locally via: info '(coreutils) nohup invocation'

1.10 - La Commande kill

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

```
[root@centos8 ~]# kill -l
 1) SIGHUP   2) SIGINT   3) SIGQUIT   4) SIGILL   5) SIGTRAP
 6) SIGABRT  7) SIGBUS    8) SIGFPE   9) SIGKILL  10) SIGUSR1
11) SIGSEGV 12) SIGUSR2   13) SIGPIPE 14) SIGALRM 15) SIGTERM
16) SIGSTKFLT 17) SIGCHLD 18) SIGCONT 19) SIGSTOP 20) SIGTSTP
21) SIGTTIN 22) SIGTTOU 23) SIGURG   24) SIGXCPU 25) SIGXFSZ
26) SIGVTALRM 27) SIGPROF 28) SIGWINCH 29) SIGIO   30) SIGPWR
31) SIGSYS   34) SIGRTMIN 35) SIGRTMIN+1 36) SIGRTMIN+2 37) SIGRTMIN+3
38) SIGRTMIN+4 39) SIGRTMIN+5 40) SIGRTMIN+6 41) SIGRTMIN+7 42) SIGRTMIN+8
43) SIGRTMIN+9 44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9 56) SIGRTMAX-8 57) SIGRTMAX-7
58) SIGRTMAX-6 59) SIGRTMAX-5 60) SIGRTMAX-4 61) SIGRTMAX-3 62) SIGRTMAX-2
63) SIGRTMAX-1 64) SIGRTMAX
```





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

Parmi les numéros de signaux les plus utiles on trouve :

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

Les options de cette commande sont :

```
[root@centos8 ~]# 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
  -L        synonym for -l
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.
```

1.11 - 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@centos8 ~]# pkill -HUP rsyslogd
```

Les options de cette commande sont :

```
[root@centos8 ~]# 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
-i, --ignore-case	match case insensitively
-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)`.