

Version : **2026.01**

Dernière mise-à-jour : 2026/03/07 08:47

LDF507 - Gestion des Processus

Contenu du Module

- **LDF507 - 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@debian11:~# cd /proc; ls -d [0-9]*
1      128      213      21985  22049  22095  22976  23037  28304  30129  368    453    59
10     12807    21895    21989  22051  22096  22991  24     28305  30151  374    458    6
11     13       21900    21995  22052  22119  22992  244    28309  30185  375    50     60
119    15        21901    22005  22060  22122  22994  25     28310  30202  377    51     69
12     16        21916    22007  22063  22131  22995  258    29     30223  380    52     72
120    17        21917    22010  22065  22136  22996  27     29208  30227  381    53     73
121    175      21920    22024  22066  22146  23     27914  29263  31     4      536    8
122    176      21921    22027  22076  22151  23008  27915  29345  32     400    54     9
123    18       21928    22036  22084  22947  23013  28     29614  353    422    55     95
125    2        21970    22040  22093  22970  23015  28194  3      364    433    57
127    20       21980    22045  22094  22974  23033  28295  30     366    452    58
```

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

```
root@debian11:/proc# cd 1 ; ls -l
total 0
-r--r--r--  1 root root 0 Jun  2 11:06 arch_status
```

```

dr-xr-xr-x  2 root root 0 May 10 14:37 attr
-rw-r--r--  1 root root 0 Jun  2 11:06 autogroup
-r-----  1 root root 0 Jun  2 11:06 auxv
-r--r--r--  1 root root 0 May 10 14:37 cgroup
--w-----  1 root root 0 Jun  2 11:06 clear_refs
-r--r--r--  1 root root 0 May 10 14:37 cmdline
-rw-r--r--  1 root root 0 May 10 14:37 comm
-rw-r--r--  1 root root 0 May 10 14:37 coredump_filter
-r--r--r--  1 root root 0 Jun  2 11:06 cpu_resctrl_groups
-r--r--r--  1 root root 0 Jun  2 11:06 cpuset
lrwxrwxrwx  1 root root 0 Jun  2 09:55 cwd -> /
-r-----  1 root root 0 May 10 14:37 environ
lrwxrwxrwx  1 root root 0 May 10 14:37 exe -> /usr/lib/systemd/systemd
dr-x-----  2 root root 0 May 10 14:37 fd
dr-x-----  2 root root 0 Jun  2 09:55 fdinfo
-rw-r--r--  1 root root 0 May 10 14:37 gid_map
-r-----  1 root root 0 Jun  2 11:06 io
-r--r--r--  1 root root 0 May 10 14:37 limits
-rw-r--r--  1 root root 0 May 10 14:37 loginuid
dr-x-----  2 root root 0 Jun  2 11:06 map_files
-r--r--r--  1 root root 0 May 10 14:37 maps
-rw-----  1 root root 0 Jun  2 11:06 mem
-r--r--r--  1 root root 0 May 10 14:37 mountinfo
-r--r--r--  1 root root 0 Jun  2 11:06 mounts
-r-----  1 root root 0 Jun  2 11:06 mountstats
dr-xr-xr-x 54 root root 0 May 10 14:37 net
dr-x--x--x  2 root root 0 Jun  2 11:06 ns
-r--r--r--  1 root root 0 Jun  2 11:06 numa_maps
-rw-r--r--  1 root root 0 Jun  2 11:06 oom_adj
-r--r--r--  1 root root 0 Jun  2 11:06 oom_score
-rw-r--r--  1 root root 0 May 10 14:37 oom_score_adj
-r-----  1 root root 0 Jun  2 11:06 pagemap
-r-----  1 root root 0 Jun  2 11:06 patch_state
-r-----  1 root root 0 Jun  2 11:06 personality

```

```
-rw-r--r-- 1 root root 0 Jun  2 11:06 projid_map
lrwxrwxrwx 1 root root 0 May 10 14:37 root -> /
-rw-r--r-- 1 root root 0 May 10 14:37 sched
-r--r--r-- 1 root root 0 Jun  2 11:06 schedstat
-r--r--r-- 1 root root 0 May 10 14:37 sessionid
-rw-r--r-- 1 root root 0 May 10 14:37 setgroups
-r--r--r-- 1 root root 0 Jun  2 11:06 smaps
-r--r--r-- 1 root root 0 Jun  2 11:06 smaps_rollup
-r----- 1 root root 0 Jun  2 11:06 stack
-r--r--r-- 1 root root 0 Jun  2 09:55 stat
-r--r--r-- 1 root root 0 Jun  2 11:06 statm
-r--r--r-- 1 root root 0 May 10 14:37 status
-r----- 1 root root 0 Jun  2 11:06 syscall
dr-xr-xr-x 3 root root 0 May 10 14:37 task
-rw-r--r-- 1 root root 0 Jun  2 11:06 timens_offsets
-r--r--r-- 1 root root 0 Jun  2 11:06 timers
-rw-rw-rw- 1 root root 0 Jun  2 11:06 timerslack_ns
-rw-r--r-- 1 root root 0 May 10 14:37 uid_map
-r--r--r-- 1 root root 0 Jun  2 11:06 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@debian11:/proc/1# cd ~
```

```
root@debian11:~# ps
```

PID	TTY	TIME	CMD
28309	pts/1	00:00:00	su
28310	pts/1	00:00:00	bash
30237	pts/1	00:00:00	ps

```
root@debian11:~# ps --forest
```

PID	TTY	TIME	CMD
28309	pts/1	00:00:00	su
28310	pts/1	00:00:00	_ bash

```
30272 pts/1    00:00:00    \_ ps
```

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

```
root@debian11:~# ps -l
F S  UID      PID      PPID  C  PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S   0       28309    28305  0   80   0 -   2529 -      pts/1    00:00:00 su
4 S   0       28310    28309  0   80   0 -   2137 -      pts/1    00:00:00 bash
4 R   0       30238    28310  0   80   0 -   2405 -      pts/1    00:00:00 ps
```

On note dans cette sortie :

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

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

```
root@debian11:~# ps lx | more
F  UID      PID      PPID  PRI  NI       VSZ       RSS WCHAN  STAT TTY          TIME COMMAND
4   0         1         0   20   0    165192    10516 -      Ss   ?           0:14 /sbin/init
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      6      2  0 -20      0      0 -      I<  ?      0:00 [kworker/0:0H-eve
nts_highpri]
1      0      8      2  0 -20      0      0 -      I<  ?      0:00 [mm_percpu_wq]
1      0      9      2 20  0      0      0 -      S  ?      0:00 [rcu_tasks_rude_]
1      0     10      2 20  0      0      0 -      S  ?      0:00 [rcu_tasks_trace]
1      0     11      2 20  0      0      0 -      S  ?      0:00 [ksoftirqd/0]
1      0     12      2 20  0      0      0 -      I  ?      0:08 [rcu_sched]
1      0     13      2 -100 -      0      0 -      S  ?      0:12 [migration/0]
1      0     15      2 20  0      0      0 -      S  ?      0:00 [cpuhp/0]
1      0     16      2 20  0      0      0 -      S  ?      0:00 [cpuhp/1]
1      0     17      2 -100 -      0      0 -      S  ?      0:12 [migration/1]
1      0     18      2 20  0      0      0 -      S  ?      0:00 [ksoftirqd/1]
1      0     20      2  0 -20      0      0 -      I<  ?      0:00 [kworker/1:0H-eve
nts_highpri]
5      0     23      2 20  0      0      0 -      S  ?      0:00 [kdevtmpfs]
1      0     24      2  0 -20      0      0 -      I<  ?      0:00 [netns]
1      0     25      2 20  0      0      0 -      S  ?      0:00 [kauditd]
1      0     27      2 20  0      0      0 -      S  ?      0:00 [khungtaskd]
--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@debian11:~# ps aux | more
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1   0.0   0.2 165192 10516 ?        Ss   May10    0:14 /sbin/init
root         2   0.0   0.0      0      0 ?        S    May10    0:00 [kthreadd]

```

```

root      3  0.0  0.0      0      0 ?      I<   May10   0:00 [rcu_gp]
root      4  0.0  0.0      0      0 ?      I<   May10   0:00 [rcu_par_gp]
root      6  0.0  0.0      0      0 ?      I<   May10   0:00 [kworker/0:0H-events_hi
ghpri]
root      8  0.0  0.0      0      0 ?      I<   May10   0:00 [mm_percpu_wq]
root      9  0.0  0.0      0      0 ?      S    May10   0:00 [rcu_tasks_rude_]
root     10  0.0  0.0      0      0 ?      S    May10   0:00 [rcu_tasks_trace]
root     11  0.0  0.0      0      0 ?      S    May10   0:00 [ksoftirqd/0]
root     12  0.0  0.0      0      0 ?      I    May10   0:08 [rcu_sched]
root     13  0.0  0.0      0      0 ?      S    May10   0:12 [migration/0]
root     15  0.0  0.0      0      0 ?      S    May10   0:00 [cpuhp/0]
root     16  0.0  0.0      0      0 ?      S    May10   0:00 [cpuhp/1]
root     17  0.0  0.0      0      0 ?      S    May10   0:12 [migration/1]
root     18  0.0  0.0      0      0 ?      S    May10   0:00 [ksoftirqd/1]
root     20  0.0  0.0      0      0 ?      I<   May10   0:00 [kworker/1:0H-events_hi
ghpri]
root     23  0.0  0.0      0      0 ?      S    May10   0:00 [kdevtmpfs]
root     24  0.0  0.0      0      0 ?      I<   May10   0:00 [netns]
root     25  0.0  0.0      0      0 ?      S    May10   0:00 [kauditd]
root     27  0.0  0.0      0      0 ?      S    May10   0:00 [khungtaskd]
--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@debian11:~# 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@debian11:~# pgrep -u root sshd
458
28295
```

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

```
root@debian11:~# pgrep -u root,trainee | more
1
2
3
4
6
8
9
10
11
12
13
```

```
15
16
17
18
20
23
24
25
27
28
29
30
--More--
```

Les options de cette commande sont :

```
root@debian11:~# 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

```
-O, --older <seconds>    select where older than seconds
-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
-r, --runstates <state>  match runstates [D,S,Z,...]
--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 `ps`

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

```
root@debian11:~# ps
systemd--ModemManager--2*[{ModemManager}]
      |--NetworkManager--2*[{NetworkManager}]
      |--agetty
      |--atd
      |--avahi-daemon--avahi-daemon
      |--colord--2*[{colord}]
      |--cron
```

```

|—cups-browsed—2*[{cups-browsed}]
|—cupsd
|—dbus-daemon
|—exim4
|—lightdm—2*[Xorg—9*[{Xorg}]]
|           |—lightdm—xfce4-session—Thunar—2*[{Thunar}]
|           |           |—applet.py
|           |           |—light-locker—3*[{light-locker}]
|           |           |—nm-applet—3*[{nm-applet}]
|           |           |—polkit-gnome-au—2*[{polkit-gnome-au}]
|           |           |—ssh-agent
|           |           |—xfce4-notifyd—2*[{xfce4-notifyd}]
|           |           |—xfce4-panel—panel-10-notifi—2*[{panel-1+
|           |           |           |—panel-14-action—2*[{panel-1+
|           |           |           |—panel-6-systray—2*[{panel-6+
|           |           |           |—panel-8-pulseau—2*[{panel-8+
|           |           |           |—panel-9-power-m—2*[{panel-9+
|           |           |           |2*[{xfce4-panel}]]
|           |           |—xfce4-power-man—2*[{xfce4-power-man}]
|           |           |—xfdesktop—2*[{xfdesktop}]
|           |           |—xfsettingsd—2*[{xfsettingsd}]
|           |           |—xfwm4—10*[{xfwm4}]
|           |           |—xiccd—2*[{xiccd}]
|           |           |2*[{xfce4-session}]
|           |           |2*[{lightdm}]
|           |—lightdm—lightdm-gtk-gre—10*[{lightdm-gtk-gre}]
|           |           |2*[{lightdm}]
|           |—lightdm
|           |2*[{lightdm}]
|—packagekitd—2*[{packagekitd}]
|—polkitd—2*[{polkitd}]
|—rsyslogd—3*[{rsyslogd}]
|—rtkit-daemon—2*[{rtkit-daemon}]
|—sshd—sshd—sshd—bash—su—bash—pstree

```

```

systemd
├── (sd-pam)
├── at-spi-bus-laun
│   ├── dbus-daemon
│   └── 3*[{at-spi-bus-laun}]
├── at-spi2-registr
│   └── 2*[{at-spi2-registr}]
├── dbus-daemon
├── gvfsd
│   └── 2*[{gvfsd}]
├── pipewire
│   ├── pipewire-media-
│   │   └── {pipewire-media-}
│   └── {pipewire}
├── pulseaudio
│   └── {pulseaudio}
systemd
├── (sd-pam)
├── at-spi-bus-laun
│   ├── dbus-daemon
│   └── 3*[{at-spi-bus-laun}]
├── at-spi2-registr
│   └── 2*[{at-spi2-registr}]
├── dbus-daemon
├── dconf-service
│   └── 2*[{dconf-service}]
├── gpg-agent
├── gvfs-udisks2-vo
│   └── 3*[{gvfs-udisks2-vo}]
├── gvfsd
│   ├── gvfsd-trash
│   │   └── 2*[{gvfsd-trash}]
│   └── 2*[{gvfsd}]
├── gvfsd-metadata
│   └── 2*[{gvfsd-metadata}]
├── pipewire
│   ├── pipewire-media-
│   │   └── {pipewire-media-}
│   └── {pipewire}
├── pulseaudio
│   └── {pulseaudio}
└── xfconfd
    └── 2*[{xfconfd}]
systemd-journal
systemd-logind
systemd-timesyn
└── {systemd-timesyn}
systemd-udev
udisksd
└── 4*[{udisksd}]
upowerd
└── 2*[{upowerd}]
wpa_supplicant
xfce4-terminal
├── bash
└── 2*[{xfce4-terminal}]

```

Les options de cette commande sont :

```
root@debian11:~# pstree --help
pstree: unrecognized option '--help'
Usage: pstree [-acglpsStTuZ] [ -h | -H PID ] [ -n | -N type ]
           [ -A | -G | -U ] [ PID | USER ]
    or: pstree -V
```

Display a tree of processes.

-a, --arguments	show command line arguments
-A, --ascii	use ASCII line drawing characters
-c, --compact-not	don't compact identical subtrees
-C, --color=TYPE	color process by attribute (age)
-g, --show-pgids	show process group ids; implies -c
-G, --vt100	use VT100 line drawing characters
-h, --highlight-all	highlight current process and its ancestors
-H PID, --highlight-pid=PID	highlight this process and its ancestors
-l, --long	don't truncate long lines
-n, --numeric-sort	sort output by PID
-N TYPE, --ns-sort=TYPE	sort output by this namespace type (cgroup, ipc, mnt, net, pid, time, 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 security attributes

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@debian11:~# top
```

```
top - 09:22:03 up 26 days, 18:44, 2 users, load average: 0.00, 0.00, 0.00
```

```
Tasks: 141 total, 1 running, 140 sleeping, 0 stopped, 0 zombie
```

```
%Cpu(s): 0.3 us, 0.2 sy, 0.0 ni, 99.5 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
```

```
MiB Mem : 3931.2 total, 1711.3 free, 405.5 used, 1814.4 buff/cache
```

```
MiB Swap: 975.0 total, 975.0 free, 0.0 used. 3238.3 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
22947	root	20	0	867344	56320	41236	S	0.3	1.4	0:11.05	Xorg
22994	lightdm	20	0	1168584	95868	59348	S	0.3	2.4	1:13.05	lightdm-gtk-gre
28304	trainee	20	0	14832	5888	4616	S	0.3	0.1	0:00.10	sshd
1	root	20	0	165192	10516	7596	S	0.0	0.3	0:14.72	systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.71	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-events_highpri
8	root	0	-20	0	0	0	I	0.0	0.0	0:00.00	mm_percpu_wq
9	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_tasks_rude_
10	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_tasks_trace
11	root	20	0	0	0	0	S	0.0	0.0	0:00.09	ksoftirqd/0
12	root	20	0	0	0	0	I	0.0	0.0	0:08.85	rcu_sched
13	root	rt	0	0	0	0	S	0.0	0.0	0:12.98	migration/0
15	root	20	0	0	0	0	S	0.0	0.0	0:00.00	cpuhp/0

```

16 root      20    0      0      0      0 S   0.0   0.0   0:00.00 cpuhp/1
17 root      rt     0      0      0      0 S   0.0   0.0   0:12.26 migration/1
18 root      20    0      0      0      0 S   0.0   0.0   0:00.13 ksoftirqd/1
20 root       0  -20      0      0      0 I   0.0   0.0   0:00.00 kworker/1:0H-events_highpri
23 root      20    0      0      0      0 S   0.0   0.0   0:00.00 kdevtmpfs
24 root       0  -20      0      0      0 I   0.0   0.0   0:00.00 netns
25 root      20    0      0      0      0 S   0.0   0.0   0:00.00 kauditd
27 root      20    0      0      0      0 S   0.0   0.0   0:00.99 khungtaskd
28 root      20    0      0      0      0 S   0.0   0.0   0:00.00 oom_reaper
29 root       0  -20      0      0      0 I   0.0   0.0   0:00.00 writeback
30 root      20    0      0      0      0 S   0.0   0.0   1:15.18 kcompactd0

```

...

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

Help for Interactive Commands - procps-ng 3.3.17

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,I Toggle: 'l' load avg; 't' task/cpu; 'm' memory; 'I' Irix mode
0,1,2,3,4 Toggle: '0' zeros; '1/2/3' cpu/numa views; '4' cpus two abreast
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,J,C . Toggle: 'R' Sort; 'H' Threads; 'J' Num justify; 'C' Coordinates
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)
V,v . Toggle: 'V' forest view; 'v' hide/show forest view children

k,r Manipulate tasks: 'k' kill; 'r' renice
d or s Set update interval

```

W,Y,!    Write config file 'W'; Inspect other output 'Y'; Combine Cpus '!'
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@debian11:~# top
...
top - 09:24:04 up 26 days, 18:46,  2 users,  load average: 0.00, 0.00, 0.00
Tasks: 141 total,   1 running, 140 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,   0.0 sy,   0.0 ni, 99.8 id,   0.1 wa,   0.0 hi,   0.0 si,   0.0 st
MiB Mem :   3931.2 total,   1711.3 free,   405.5 used,   1814.4 buff/cache
MiB Swap:    975.0 total,    975.0 free,     0.0 used.   3238.3 avail Mem
Change delay from 3.0 to 1
   PID USER      PR  NI   VIRT    RES    SHR S  %CPU  %MEM    TIME+  COMMAND
...

```

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

```

root@debian11:~# top
...
top - 09:25:13 up 26 days, 18:48,  2 users,  load average: 0.00, 0.00, 0.00
Tasks: 141 total,   1 running, 140 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,   0.0 sy,   0.0 ni,100.0 id,   0.0 wa,   0.0 hi,   0.0 si,   0.0 st
MiB Mem :   3931.2 total,   1711.1 free,   405.7 used,   1814.4 buff/cache
MiB Swap:    975.0 total,    975.0 free,     0.0 used.   3238.0 avail Mem

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
22994	lightdm	20	0	1168584	95868	59348	S	0.0	2.4	1:13.09	lightdm-gtk-gre
452	root	20	0	861748	78164	38060	S	0.0	1.9	1:07.28	Xorg
22007	trainee	20	0	973148	73384	57724	S	0.0	1.8	0:03.88	xfwm4
22947	root	20	0	867344	56320	41236	S	0.0	1.4	0:11.05	Xorg
22045	trainee	20	0	329636	53740	32396	S	0.0	1.3	0:05.50	xfdesktop
22036	trainee	20	0	319156	40340	31788	S	0.0	1.0	0:06.02	xfce4-panel
22146	trainee	20	0	386976	36676	28888	S	0.0	0.9	0:00.20	xfce4-terminal
22051	trainee	20	0	62288	36480	18088	S	0.0	0.9	0:00.49	applet.py
22096	trainee	20	0	240616	35676	28976	S	0.0	0.9	0:00.27	panel-10-notifi
22052	trainee	20	0	355636	33224	26600	S	0.0	0.8	0:00.17	nm-applet
22094	trainee	20	0	464892	31108	24892	S	0.0	0.8	2:42.24	panel-8-pulseau
22095	trainee	20	0	197048	28304	22796	S	0.0	0.7	0:00.29	panel-9-power-m
213	root	20	0	56684	28064	24908	S	0.0	0.7	0:08.07	systemd-journal
22093	trainee	20	0	341076	24972	18016	S	0.0	0.6	0:00.25	panel-6-systray
22040	trainee	20	0	341956	24776	17892	S	0.0	0.6	0:00.06	Thunar
22024	trainee	20	0	227856	24132	18852	S	0.0	0.6	0:00.17	xfsettingsd
22119	trainee	20	0	195152	23308	18416	S	0.0	0.6	0:00.23	panel-14-action
22063	trainee	20	0	268476	23096	18368	S	0.0	0.6	0:01.99	light-locker
21928	trainee	20	0	267564	22348	17344	S	0.0	0.6	0:00.16	xfce4-session
22049	trainee	20	0	264284	19340	15216	S	0.0	0.5	0:00.02	xfce4-notifyd
12807	root	20	0	289480	18932	16368	S	0.0	0.5	0:07.01	packagekitd
22060	trainee	20	0	190476	17604	15312	S	0.0	0.4	0:00.10	xfce4-power-man
368	root	20	0	254628	16664	14172	S	0.0	0.4	0:46.42	NetworkManager
22065	trainee	20	0	189472	15168	13384	S	0.0	0.4	0:00.04	polkit-gnome-au
22122	trainee	20	0	348880	13816	10064	S	0.0	0.3	0:00.03	gvfs-udisks2-vo
22084	colord	20	0	242896	13092	9484	S	0.0	0.3	0:00.12	colord

...

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

```
root@debian11:~# top
```

...

```
top - 09:26:38 up 26 days, 18:49, 2 users, load average: 0.00, 0.00, 0.00
```

```
Tasks: 142 total,   1 running, 141 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,   0.0 sy,   0.0 ni,100.0 id,   0.0 wa,   0.0 hi,   0.0 si,   0.0 st
MiB Mem :  3931.2 total,   1711.1 free,   405.7 used,   1814.4 buff/cache
MiB Swap:   975.0 total,   975.0 free,    0.0 used.  3238.0 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
30295	root	20	0	10100	3512	3016	R	1.0	0.1	0:00.51	top

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

Les options de cette commande sont :

```
root@debian11:~# top --help
top: inappropriate '-help'
Usage:
  top -hv | -bcEeHiOSs1 -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 &
```



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 :

```
root@debian11:~# sleep 9999 &
[1] 30316

root@debian11:~# jobs -l
[1]+ 30316 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@debian11:~# 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@debian11:~# bg %2
[2]+ sleep 1234 &

root@debian11:~# jobs -l
[1]- 30316 Running                  sleep 9999 &
[2]+ 30320 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@debian11:~# kill -s stop %2

root@debian11:~# jobs -l
[1]- 30316 Running          sleep 9999 &
[2]+ 30320 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@debian11:~# kill -s cont %2

root@debian11:~# jobs -l
[1]- 30316 Running          sleep 9999 &
[2]+ 30320 Running         sleep 1234 &
```

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

```
root@debian11:~# kill -s stop %2

root@debian11:~# jobs -l
[1]- 30316 Running          sleep 9999 &
[2]+ 30320 Stopped (signal) sleep 1234

root@debian11:~# fg %2
sleep 1234
^C
root@debian11:~#
```



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

Les options de la commande jobs sont :

```
root@debian11:~# 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@debian11:~# jobs -l
[1]+ 30316 Running                  sleep 9999 &

root@debian11:~# wait %1
^C

root@debian11:~# jobs -l
```


[1]+ 30316 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@debian11:~# nice -n -20 sleep 1234
^Z
[2]+  Stopped                  nice -n -20 sleep 1234

root@debian11:~# ps lx | grep sleep
0      0    30316   28310  20    0    5304    508 -      S    pts/1    0:00 sleep 9999
4      0    30370   28310    0 -20    5304    576 -      T<   pts/1    0:00 sleep 1234
0      0    30373   28310  20    0    6180    716 -      S+   pts/1    0:00 grep sleep

root@debian11:~# nice -n 19 sleep 5678
^Z
[3]+  Stopped                  nice -n 19 sleep 5678

root@debian11:~# ps lx | grep sleep
0      0    30316   28310  20    0    5304    508 -      S    pts/1    0:00 sleep 9999
4      0    30370   28310    0 -20    5304    576 -      T<   pts/1    0:00 sleep 1234
0      0    30374   28310  39   19    5304    572 -      TN   pts/1    0:00 sleep 5678
0      0    30379   28310  20    0    6180    724 -      S+   pts/1    0:00 grep sleep
```

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



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

Les options de cette commande sont :

```
root@debian11:~# 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 <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@debian11:~# jobs -l
[1] 30316 Running                  sleep 9999 &
```

```
[2]- 30370 Stopped          nice -n -20 sleep 1234
[3]+ 30374 Stopped          nice -n 19 sleep 5678

root@debian11:~# bg %2
[2]- nice -n -20 sleep 1234 &

root@debian11:~# bg %3
[3]+ nice -n 19 sleep 5678 &

root@debian11:~# jobs -l
[1] 30316 Running           sleep 9999 &
[2]- 30370 Running          nice -n -20 sleep 1234 &
[3]+ 30374 Running          nice -n 19 sleep 5678 &

root@debian11:~# renice +5 30370
30370 (process ID) old priority -20, new priority 5

root@debian11:~# renice -5 30374
30374 (process ID) old priority 19, new priority -5

root@debian11:~# ps lx | grep sleep
0      0    30316   28310  20    0   5304   508 -    S    pts/1    0:00 sleep 9999
4      0    30370   28310  25    5   5304   576 -   SN   pts/1    0:00 sleep 1234
0      0    30374   28310  15   -5   5304   572 -    S<   pts/1    0:00 sleep 5678
0      0    30398   28310  20    0   6180   712 -    S+   pts/1    0:00 grep sleep
```



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

Les options de cette commande sont :

```
root@debian11:~# renice --help
```

Usage:

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

Alter the priority of running processes.

Options:

-n, --priority <num>	specify the nice value
-p, --pid	interpret arguments as process ID (default)
-g, --pgrp	interpret arguments as process group ID
-u, --user	interpret arguments 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@debian11:~# 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 <<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@debian11:~# 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@debian11:~# 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@debian11:~# pkill -HUP rsyslogd
```

Les options de cette commande sont :

```
root@debian11:~# pkill --help
```

Usage:

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

Options:

-<sig>, --signal <sig>	signal to send (either number or name)
-q, --queue <value>	integer value to be sent with the signal
-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
-O, --older <seconds>	select where older than seconds
-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
-r, --runstates <state> match runstates [D,S,Z,...]
--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)`.