

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

Gestion des Processus

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

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

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

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

Saisissez la commande suivante :

```
opensuse:~ # cd /proc; ls -d [0-9]*
1      11496 16    22    31    3783  4039  5881  6015  6053  6127  9503  9937
10     11505 1636  23    32    38    4040  5924  6017  6062  6138  9772
10019  11508 1647  2359  322   3830  4041  5925  6020  6063  6140  9777
1077   11525 1651  2387  33    3869  4042  5930  6022  6064  6142  9779
1083   12     1697  24    34    3889  522   5938  6025  6074  6144  9781
1093   1273  17    247   3448  39    532   5944  6028  6076  6150  9823
```

11	13	18	25	35	3919	557	5951	6029	6110	7	9906
11147	1351	19	253	3506	3920	5772	5958	6031	6113	8	9914
1122	14	2	26	36	40	5794	5960	6041	6116	9	9917
11263	15	20	27	3682	4037	5868	6	6042	6120	920	9919
11483	1594	21	3	3755	4038	5876	6010	6045	6126	942	9931

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

```
opensuse:/proc # cd 1 ; ls -l
total 0
dr-xr-xr-x 2 root root 0 May  5 13:40 attr
-rw-r--r-- 1 root root 0 May  5 13:40 autogroup
-r----- 1 root root 0 May  5 13:40 auxv
-r--r--r-- 1 root root 0 May  5 13:40 cgroup
--w----- 1 root root 0 May  5 13:40 clear_refs
-r--r--r-- 1 root root 0 May  5 13:40 cmdline
-rw-r--r-- 1 root root 0 May  5 13:40 comm
-rw-r--r-- 1 root root 0 May  5 13:40 coredump_filter
-r--r--r-- 1 root root 0 May  5 13:40 cpuset
lrwxrwxrwx 1 root root 0 May  5 13:40 cwd -> /
-r----- 1 root root 0 May  5 13:40 environ
lrwxrwxrwx 1 root root 0 May  5 13:40 exe -> /sbin/init
dr-x----- 2 root root 0 May  5 13:40 fd
dr-x----- 2 root root 0 May  5 13:40 fdinfo
-r--r--r-- 1 root root 0 May  5 13:40 io
-r--r--r-- 1 root root 0 May  5 13:40 latency
-r--r--r-- 1 root root 0 May  5 13:40 limits
-rw-r--r-- 1 root root 0 May  5 13:40 loginuid
-r--r--r-- 1 root root 0 May  5 13:40 maps
-rw----- 1 root root 0 May  5 13:40 mem
-r--r--r-- 1 root root 0 May  5 13:40 mountinfo
-r--r--r-- 1 root root 0 May  5 13:40 mounts
-r----- 1 root root 0 May  5 13:40 mountstats
dr-xr-xr-x 5 root root 0 May  5 13:40 net
```

```
-r--r--r-- 1 root root 0 May 5 13:40 numa_maps
-rw-r--r-- 1 root root 0 May 5 13:40 oom_adj
-r--r--r-- 1 root root 0 May 5 13:40 oom_score
-rw-r--r-- 1 root root 0 Apr 18 11:53 oom_score_adj
-r----- 1 root root 0 May 5 13:40 pagemap
-r----- 1 root root 0 May 5 13:40 personality
lrwxrwxrwx 1 root root 0 May 5 13:40 root -> /
-rw-r--r-- 1 root root 0 May 5 13:40 sched
-r--r--r-- 1 root root 0 May 5 13:40 schedstat
-r--r--r-- 1 root root 0 May 5 13:40 sessionid
-r--r--r-- 1 root root 0 May 5 13:40 smaps
-r----- 1 root root 0 May 5 13:40 stack
-r--r--r-- 1 root root 0 Apr 10 09:29 stat
-r--r--r-- 1 root root 0 May 5 13:40 statm
-r--r--r-- 1 root root 0 Apr 10 09:29 status
-r----- 1 root root 0 May 5 13:40 syscall
dr-xr-xr-x 3 root root 0 May 5 13:40 task
-r--r--r-- 1 root root 0 May 5 13:40 wchan
```

<note important> Vous n'avez pas besoin de consulter le contenu des fichiers et des répertoires. Il convient tout simplement de savoir que ces données existent. Naviguez donc à /root en ligne de commande. </note>

Les Types de Processus

Il existe trois types de processus :

- **interactif** qui est lancé par le shell dans une console en premier plan ou en tâche de fond
- **batch** qui est lancé par le système au moment propice
- **daemon** qui est lancé au démarrage par le système (lpd, dns etc)

Un processus peut être dans un de neuf états ou *process states* :

<note important>

- *user mode* - le processus s'exécute en mode utilisateur,
- *kernel mode* - le processus s'exécute en mode noyau,
- *waiting* - le processus est en attente pour une ressource autre que le processeur,
- *sleeping* - le processus est endormi,
- *runnable* - le processus dispose de toutes 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.

</note>

Les Commandes relatives aux Processus

La commande ps

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

```
opensuse:/proc/1 # cd ~
opensuse:~ # ps
  PID TTY          TIME CMD
 9931 pts/0        00:00:00 su
 9937 pts/0        00:00:00 bash
11542 pts/0        00:00:00 ps
```

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

```
opensuse:~ # ps -l
 F S      UID      PID   PPID    C  PRI   NI  ADDR  SZ  WCHAN   TTY          TIME CMD
 4 S        0    9931    9919    0   80    0   -    1153 wait  pts/0    00:00:00 su
 4 S        0    9937    9931    0   80    0   -    1289 wait  pts/0    00:00:00 bash
```

```
4 R      0 11543 9937 0 80 0 - 655 - 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 :

```
opensuse:~ # ps lx
F  UID  PID  PPID PRI  NI    VSZ   RSS WCHAN  STAT TTY      TIME COMMAND
4   0    1    0  20   0   2216   724 -      Ss   ?        0:00 init [5]
1   0    2    0  20   0    0     0 kthrea S    ?        0:00 [kthreadd]
1   0    3    2  20   0    0     0 run_ks S    ?        0:00 [ksoftirqd]
1   0    6    2 -100  -    0     0 cpu_st S    ?        0:00 [migration]
1   0    7    2   0 -20    0     0 rescue S<   ?        0:00 [cpuset]
1   0    8    2   0 -20    0     0 rescue S<   ?        0:00 [khelper]
1   0    9    2   0 -20    0     0 rescue S<   ?        0:00 [netns]
1   0   10    2  20   0    0     0 bdi_sy S    ?        0:00 [sync_supe]
1   0   11    2  20   0    0     0 -      S    ?        0:00 [bdi-defau]
1   0   12    2   0 -20    0     0 rescue S<   ?        0:00 [kintegrit]
```

1	0	13	2	0	-20	0	0	rescue	S<	?	0:00	[kblockd]
1	0	14	2	0	-20	0	0	rescue	S<	?	0:00	[kacpid]
1	0	15	2	0	-20	0	0	rescue	S<	?	0:00	[kacpi_not]
1	0	16	2	0	-20	0	0	rescue	S<	?	0:00	[kacpi_hot]
1	0	17	2	0	-20	0	0	rescue	S<	?	0:00	[ata_sff]
5	0	18	2	20	0	0	0	hub_th	S	?	0:00	[khubd]
5	0	19	2	20	0	0	0	serio_	S	?	0:00	[kseriod]
1	0	20	2	0	-20	0	0	rescue	S<	?	0:00	[md]
1	0	21	2	0	-20	0	0	rescue	S<	?	0:00	[kondemand]
1	0	22	2	20	0	0	0	-	S	?	0:00	[khungtask]
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	20	0	0	0	fsnoti	S	?	0:00	[fsnotify_]
1	0	26	2	0	-20	0	0	rescue	S<	?	0:00	[aio]
1	0	27	2	0	-20	0	0	rescue	S<	?	0:00	[crypto]
1	0	31	2	0	-20	0	0	rescue	S<	?	0:00	[kthrotld]
1	0	32	2	20	0	0	0	scsi_e	S	?	0:00	[scsi_eh_0]
5	0	33	2	20	0	0	0	worker	S	?	0:00	[kworker/u]
1	0	34	2	20	0	0	0	scsi_e	S	?	0:00	[scsi_eh_1]
1	0	35	2	20	0	0	0	scsi_e	S	?	0:07	[scsi_eh_2]
5	0	36	2	20	0	0	0	worker	S	?	0:00	[kworker/u]
1	0	38	2	0	-20	0	0	rescue	S<	?	0:00	[kpsmoused]
1	0	39	2	0	-20	0	0	rescue	S<	?	0:00	[usbhid_re]
1	0	40	2	0	-20	0	0	rescue	S<	?	0:00	[ceph-msgr]
1	0	247	2	20	0	0	0	kjourn	S	?	0:03	[kjournald]
1	0	253	2	20	0	0	0	-	S	?	0:02	[flush-8:0]
5	0	322	1	16	-4	3028	1212	-	S<s	?	0:00	/sbin/udevd
1	0	522	2	20	0	0	0	kaudit	S	?	0:00	[kauditd]
1	0	532	2	0	-20	0	0	rescue	S<	?	0:00	[kstriped]
1	0	557	2	20	0	0	0	kjourn	S	?	0:00	[kjournald]
1	0	942	1	20	0	1920	588	-	Ss	?	0:00	/sbin/acpid
4	0	1077	1	20	0	14416	2444	-	Sl	?	0:00	/usr/sbin/g
4	0	1083	1077	20	0	18388	3792	-	Sl	?	0:00	/usr/lib/gd
4	0	1093	1083	20	0	47336	37056	-	Ss+	tty7	4:10	/usr/bin/Xo

0	0	1122	1	20	0	1908	240	-	Ss	?	0:00	startpar -f
4	0	1273	1	20	0	27596	3208	-	Sl	?	0:00	/usr/sbin/c
4	0	1351	1	20	0	25144	4608	-	Sl	?	0:00	/usr/lib/po
4	0	1636	1083	20	0	16588	3076	-	Sl	?	0:00	/usr/lib/gd
4	0	1647	1	20	0	17476	3616	-	Sl	?	0:00	/usr/lib/up
4	0	1697	1	20	0	5248	2840	-	S	?	0:00	/sbin/dhcli
1	0	2359	1	20	0	2296	164	-	Ss	?	0:00	/sbin/dhcpc
1	0	2387	1	20	0	6772	520	-	Sl	?	0:04	/usr/sbin/V
5	0	3506	1	20	0	35884	1204	-	Sl	?	0:00	/sbin/rsysl
5	0	3682	1	16	-4	10812	568	-	S<sl	?	0:00	/sbin/audit
1	0	3755	1	20	0	1908	228	-	Ss	?	0:00	/usr/sbin/n
5	0	3783	1	20	0	2368	448	-	Ss	?	0:00	/sbin/rpcbi
4	0	3830	1	20	0	7540	2684	-	Ss	?	0:00	/usr/sbin/c
1	0	3869	1	20	0	5268	824	-	Ss	?	0:09	/usr/sbin/s
4	0	3889	1	20	0	4208	1376	-	Ss	?	0:00	/usr/lib/po
1	0	3920	1	20	0	3140	976	-	Ss	?	0:00	/usr/sbin/c
4	0	4037	1	20	0	2180	688	-	Ss+	tty1	0:00	/sbin/minge
4	0	4038	1	20	0	2180	692	-	Ss+	tty2	0:00	/sbin/minge
4	0	4039	1	20	0	2180	692	-	Ss+	tty3	0:00	/sbin/minge
4	0	4040	1	20	0	2180	688	-	Ss+	tty4	0:00	/sbin/minge
4	0	4041	1	20	0	2180	692	-	Ss+	tty5	0:00	/sbin/minge
4	0	4042	1	20	0	2180	688	-	Ss+	tty6	0:00	/sbin/minge
4	0	6020	1	20	0	22804	3452	-	Sl	?	0:00	/usr/lib/ud
1	0	6022	6020	20	0	6144	716	-	S	?	0:08	udisks-daem
5	0	9503	322	18	-2	3024	1224	-	S<	?	0:00	/sbin/udev
4	0	9931	9919	20	0	4612	1652	wait	S	pts/0	0:00	su -
4	0	9937	9931	20	0	5156	2340	wait	S	pts/0	0:00	-bash
1	0	11147	2	20	0	0	0	worker	S	?	0:01	[kworker/0]
1	0	11263	2	20	0	0	0	worker	S	?	0:00	[kworker/0]
5	0	11483	322	18	-2	2904	1192	-	S<	?	0:00	/sbin/udev
1	0	11551	2	20	0	0	0	worker	S	?	0:00	[kworker/0]
4	0	11552	9937	20	0	2620	804	-	R+	pts/0	0:00	ps lx

On note dans cette sortie certaines informations supplémentaires :

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

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

```

opensuse:~ # ps aux
USER      PID  %CPU  %MEM    VSZ   RSS  TTY      STAT START   TIME COMMAND
root         1   0.0   0.0   2216    724 ?        Ss   06:59   0:00 init [5]
root         2   0.0   0.0     0      0 ?        S    06:59   0:00 [kthreadd]
root         3   0.0   0.0     0      0 ?        S    06:59   0:00 [ksoftirqd/0]
root         6   0.0   0.0     0      0 ?        S    06:59   0:00 [migration/0]
root         7   0.0   0.0     0      0 ?        S<   06:59   0:00 [cpuset]
root         8   0.0   0.0     0      0 ?        S<   06:59   0:00 [khelper]
root         9   0.0   0.0     0      0 ?        S<   06:59   0:00 [netns]
root        10   0.0   0.0     0      0 ?        S    06:59   0:00 [sync_supers]
root        11   0.0   0.0     0      0 ?        S    06:59   0:00 [bdi-default]
root        12   0.0   0.0     0      0 ?        S<   06:59   0:00 [kintegrityd]
root        13   0.0   0.0     0      0 ?        S<   06:59   0:00 [kblockd]
root        14   0.0   0.0     0      0 ?        S<   06:59   0:00 [kacpid]
root        15   0.0   0.0     0      0 ?        S<   06:59   0:00 [kacpi_notify]
root        16   0.0   0.0     0      0 ?        S<   06:59   0:00 [kacpi_hotplug]
root        17   0.0   0.0     0      0 ?        S<   06:59   0:00 [ata_sff]
root        18   0.0   0.0     0      0 ?        S    06:59   0:00 [khubd]
root        19   0.0   0.0     0      0 ?        S    06:59   0:00 [kseriod]
root        20   0.0   0.0     0      0 ?        S<   06:59   0:00 [md]
root        21   0.0   0.0     0      0 ?        S<   06:59   0:00 [kondemand]
root        22   0.0   0.0     0      0 ?        S    06:59   0:00 [khungtaskd]
root        23   0.0   0.0     0      0 ?        S    06:59   0:00 [kswapd0]
root        24   0.0   0.0     0      0 ?        SN   06:59   0:00 [ksmd]
root        25   0.0   0.0     0      0 ?        S    06:59   0:00 [fsnotify_mark]
root        26   0.0   0.0     0      0 ?        S<   06:59   0:00 [aio]
root        27   0.0   0.0     0      0 ?        S<   06:59   0:00 [crypto]
root        31   0.0   0.0     0      0 ?        S<   06:59   0:00 [kthrotld]

```

root	32	0.0	0.0	0	0 ?	S	06:59	0:00	[scsi_eh_0]
root	33	0.0	0.0	0	0 ?	S	06:59	0:00	[kworker/u:1]
root	34	0.0	0.0	0	0 ?	S	06:59	0:00	[scsi_eh_1]
root	35	0.0	0.0	0	0 ?	S	06:59	0:07	[scsi_eh_2]
root	36	0.0	0.0	0	0 ?	S	06:59	0:00	[kworker/u:2]
root	38	0.0	0.0	0	0 ?	S<	06:59	0:00	[kpsmouse]
root	39	0.0	0.0	0	0 ?	S<	06:59	0:00	[usbhid_resume]
root	40	0.0	0.0	0	0 ?	S<	06:59	0:00	[ceph-msgr]
root	247	0.0	0.0	0	0 ?	S	06:59	0:03	[kjournald]
root	253	0.0	0.0	0	0 ?	S	06:59	0:02	[flush-8:0]
root	322	0.0	0.1	3028	1212 ?	S<s	06:59	0:00	/sbin/udev
root	522	0.0	0.0	0	0 ?	S	06:59	0:00	[kauditd]
root	532	0.0	0.0	0	0 ?	S<	06:59	0:00	[kstripped]
root	557	0.0	0.0	0	0 ?	S	06:59	0:00	[kjournald]
101	920	0.0	0.1	3820	1788 ?	Ss	06:59	0:00	/bin/dbus-daemo
root	942	0.0	0.0	1920	588 ?	Ss	06:59	0:00	/sbin/acpid
root	1077	0.0	0.2	14416	2444 ?	Sl	06:59	0:00	/usr/sbin/gdm
root	1083	0.0	0.3	18388	3792 ?	Sl	06:59	0:00	/usr/lib/gdm/gd
root	1093	1.0	3.6	47336	37056 tty7	Ss+	06:59	4:11	/usr/bin/Xorg :
root	1122	0.0	0.0	1908	240 ?	Ss	06:59	0:00	startpar -f --
root	1273	0.0	0.3	27596	3208 ?	Sl	06:59	0:00	/usr/sbin/conso
root	1351	0.0	0.4	25144	4608 ?	Sl	06:59	0:00	/usr/lib/polkit
gdm	1594	0.0	0.7	29052	7496 ?	Sl	06:59	0:00	/usr/lib/polkit
root	1636	0.0	0.3	16588	3076 ?	Sl	06:59	0:00	/usr/lib/gdm/gd
root	1647	0.0	0.3	17476	3616 ?	Sl	06:59	0:00	/usr/lib/upower
rtkit	1651	0.0	0.1	19036	1180 ?	SNl	06:59	0:00	/usr/lib/rtkit/
root	1697	0.0	0.2	5248	2840 ?	S	06:59	0:00	/sbin/dhclient6
root	2359	0.0	0.0	2296	164 ?	Ss	07:00	0:00	/sbin/dhccpd -p
root	2387	0.0	0.0	6772	520 ?	Sl	07:00	0:04	/usr/sbin/VBoxS
avahi	3448	0.0	0.1	3196	1412 ?	S	07:00	0:00	avahi-daemon: r
root	3506	0.0	0.1	35884	1204 ?	Sl	07:00	0:00	/sbin/rsyslogd
root	3682	0.0	0.0	10812	568 ?	S<sl	07:00	0:00	/sbin/auditd -s
root	3755	0.0	0.0	1908	228 ?	Ss	07:00	0:00	/usr/sbin/nscd
root	3783	0.0	0.0	2368	448 ?	Ss	07:00	0:00	/sbin/rpcbind

root	3830	0.0	0.2	7540	2684	?	Ss	07:00	0:00	/usr/sbin/cupsd
root	3869	0.0	0.0	5268	824	?	Ss	07:00	0:09	/usr/sbin/smppp
root	3889	0.0	0.1	4208	1376	?	Ss	07:00	0:00	/usr/lib/postfi
postfix	3919	0.0	0.1	4376	1716	?	S	07:00	0:00	qmgr -l -t fifo
root	3920	0.0	0.0	3140	976	?	Ss	07:00	0:00	/usr/sbin/cron
root	4037	0.0	0.0	2180	688	tty1	Ss+	07:00	0:00	/sbin/mingetty
root	4038	0.0	0.0	2180	692	tty2	Ss+	07:00	0:00	/sbin/mingetty
root	4039	0.0	0.0	2180	692	tty3	Ss+	07:00	0:00	/sbin/mingetty
root	4040	0.0	0.0	2180	688	tty4	Ss+	07:00	0:00	/sbin/mingetty
root	4041	0.0	0.0	2180	692	tty5	Ss+	07:00	0:00	/sbin/mingetty
root	4042	0.0	0.0	2180	688	tty6	Ss+	07:00	0:00	/sbin/mingetty
trainee	5772	0.0	0.4	42248	4104	?	Sl	07:07	0:00	/usr/bin/gnome-
trainee	5794	0.0	0.7	36740	7456	?	Ssl	07:07	0:01	/usr/bin/gnome-
trainee	5868	0.0	0.1	6032	1692	?	Sl	07:07	0:00	/usr/bin/VBoxCl
trainee	5876	0.0	0.1	6016	1568	?	Sl	07:07	0:00	/usr/bin/VBoxCl
trainee	5881	0.0	0.1	5944	1124	?	Sl	07:07	0:00	/usr/bin/VBoxCl
trainee	5924	0.0	0.0	3444	492	?	S	07:07	0:00	dbus-launch --e
trainee	5925	0.0	0.1	4336	1764	?	Ss	07:07	0:06	/bin/dbus-daemo
trainee	5930	0.0	0.5	10392	5120	?	S	07:07	0:00	/usr/lib/GConf/
trainee	5938	0.0	1.4	115140	14540	?	Ssl	07:07	0:05	/usr/lib/gnome-
trainee	5944	0.0	0.3	8508	3184	?	S	07:07	0:00	/usr/lib/gvfs/g
trainee	5951	0.0	0.2	32904	2728	?	Ssl	07:07	0:00	/usr/lib/gvfs//
trainee	5958	0.0	1.1	31772	11412	?	Sl	07:07	0:07	/usr/bin/metaci
trainee	5960	0.0	0.4	95472	4708	?	Ssl	07:07	0:00	/usr/bin/pulsea
trainee	6010	0.0	0.3	21536	3236	?	Sl	07:07	0:00	/usr/lib/pulse/
trainee	6015	0.1	1.8	74508	18692	?	Sl	07:07	0:25	gnome-panel
trainee	6017	0.0	0.4	35520	4448	?	S	07:07	0:00	/usr/lib/gvfs/g
root	6020	0.0	0.3	22804	3452	?	Sl	07:07	0:00	/usr/lib/udisks
root	6022	0.0	0.0	6144	716	?	S	07:07	0:08	udisks-daemon:
trainee	6025	0.0	0.2	18372	2320	?	Sl	07:07	0:00	/usr/lib/gvfs/g
trainee	6028	0.0	0.2	8348	2432	?	S	07:07	0:00	/usr/lib/gvfs/g
trainee	6029	0.0	2.2	84028	22716	?	Sl	07:07	0:15	nautilus
trainee	6031	0.0	0.3	50880	3772	?	Ssl	07:07	0:00	/usr/lib/bonobo
trainee	6041	0.0	1.7	52680	17552	?	Sl	07:07	0:07	/usr/lib/gnome-

trainee	6042	0.0	0.1	4516	1244	?	S	07:07	0:00	bash /usr/bin/t
trainee	6045	0.0	1.9	73504	19652	?	Sl	07:07	0:01	mono /usr/lib/t
trainee	6053	0.0	0.3	9656	3516	?	S	07:07	0:00	/usr/lib/gvfs/g
trainee	6062	0.0	0.7	40528	8040	?	Sl	07:07	0:00	/usr/lib/evolut
trainee	6063	0.0	0.8	76760	8852	?	SNl	07:07	0:04	/usr/lib/tracke
trainee	6064	0.0	0.1	4516	1336	?	S	07:07	0:00	/bin/sh /usr/bi
trainee	6074	0.0	1.2	53956	12204	?	Sl	07:07	0:00	nm-applet --sm-
trainee	6076	0.0	1.1	101572	11772	?	S	07:07	0:00	gnome-volume-co
trainee	6110	0.0	0.8	23288	8216	?	S	07:07	0:00	/usr/lib/gdu-no
trainee	6113	0.0	2.2	75048	22936	?	Sl	07:07	0:06	mono /usr/lib/g
trainee	6116	0.0	1.3	43760	13532	?	Sl	07:07	0:00	gpk-update-icon
trainee	6120	0.0	1.6	34648	16352	?	S	07:07	0:01	/usr/bin/python
trainee	6126	0.0	1.1	68060	11512	?	Sl	07:07	0:08	/usr/lib/tracke
trainee	6127	0.0	0.5	26836	5912	?	Sl	07:07	0:00	/usr/lib/polkit
trainee	6138	0.0	0.9	48700	9716	?	Sl	07:07	0:00	bluetooth-apple
trainee	6140	0.0	1.2	42388	12724	?	Sl	07:07	0:01	gnome-power-man
trainee	6142	0.0	0.2	7592	2152	?	S	07:07	0:00	/usr/lib/gvfs/g
trainee	6144	0.0	0.5	28564	5676	?	Ss	07:07	0:01	gnome-screensav
trainee	6150	0.0	0.2	7468	2104	?	S	07:07	0:00	/usr/lib/gvfs/g
root	9503	0.0	0.1	3024	1224	?	S<	10:48	0:00	/sbin/udev --d
trainee	9772	1.2	6.0	374380	61444	?	Sl	10:52	2:08	/opt/google/chr
trainee	9777	0.0	1.0	125672	10496	?	Sl	10:52	0:01	/opt/google/chr
trainee	9779	0.0	1.1	96740	12044	?	S	10:52	0:00	/opt/google/chr
trainee	9781	0.0	0.2	1065680	2460	?	S	10:52	0:00	/opt/google/chr
trainee	9823	0.0	2.8	168704	28400	?	Sl	10:52	0:01	/opt/google/chr
trainee	9906	4.0	7.1	203076	72672	?	Sl	10:53	6:45	/opt/google/chr
trainee	9914	0.1	1.4	60684	14812	?	Sl	10:54	0:16	gnome-terminal
trainee	9917	0.0	0.0	1904	248	?	S	10:54	0:00	gnome-pty-helpe
trainee	9919	0.0	0.2	5076	2196	pts/0	Ss	10:54	0:00	bash
root	9931	0.0	0.1	4612	1652	pts/0	S	10:54	0:00	su -
root	9937	0.0	0.2	5156	2340	pts/0	S	10:54	0:00	-bash
trainee	10019	0.0	3.3	174500	33612	?	Sl	10:58	0:05	/opt/google/chr
root	11147	0.0	0.0	0	0	?	S	13:03	0:01	[kworker/0:2]
root	11263	0.0	0.0	0	0	?	S	13:19	0:00	[kworker/0:0]

```

root      11483  0.0  0.1   2904   1192 ?        S<    13:37   0:00 /sbin/udevd --d
postfix   11496  0.0  0.1   4176   1180 ?        S     13:37   0:00 pickup -l -t fi
trainee   11505  0.4  1.8  47700  18280 ?        Sl    13:37   0:01 /usr/bin/applic
trainee   11508  2.4  2.0  92212  20440 ?        Sl    13:37   0:06 gedit
root      11551  0.0  0.0      0      0 ?        S     13:41   0:00 [kworker/0:1]
trainee   11558  0.5  0.7  35220   7496 ?        SNl   13:42   0:00 /usr/lib/tracke
root      11560  0.0  0.0   2620    872 pts/0    R+    13:42   0:00 ps aux

```

On note dans cette sortie certaines informations supplémentaires :

USER	L'utilisateur du processus
%CPU	Ressources du microprocesseur utilisées par le processus
%MEM	Ressources en mémoire vive utilisées par le processus

Options de la commande ps

Les options de cette commande sont :

```

opensuse:~ # ps --help
***** simple selection *****
-A all processes
-N negate selection
-a all w/ tty except session leaders
-d all except session leaders
-e all processes
T all processes on this terminal
a all w/ tty, including other users
g OBSOLETE -- DO NOT USE
r only running processes
x processes w/o controlling ttys
***** output format *****
-o,o user-defined -f full
-j,j job control  s  signal

***** selection by list *****
-C by command name
-G by real group ID (supports names)
-U by real user ID (supports names)
-g by session OR by effective group name
-p by process ID
-s processes in the sessions given
-t by tty
-u by effective user ID (supports names)
U processes for specified users
t by tty
***** long options *****
--Group --User --pid --cols --ppid
--group --user --sid --rows --info

```

```
-O,0 preloaded -o v virtual memory --cumulative --format --deselect
-l,l long          u user-oriented --sort --tty --forest --version
-F extra full      X registers      --heading --no-heading --context
                      ***** misc options *****
-V,V show version  L list format codes f ASCII art forest
-m,m,-L,-T,H threads S children in sum -y change -l format
-M,Z security data c true command name -c scheduling class
-w,w wide output   n numeric WCHAN,UID -H process hierarchy
```

La commande pstree

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

```
opensuse:~ # pstree
init--3*[VBoxClient--{VBoxClient}]
    |--VBoxService--7*[{VBoxService}]
    |--acpid
    |--application-bro--{application-br}
    |--auditd--{auditd}
    |--avahi-daemon
    |--bash--tomboy--2*[{tomboy}]
    |--bonobo-activati--2*[{bonobo-activat}]
    |--chrome--chrome--{chrome}
    |       |--20*[{chrome}]
    |--chrome--3*[chrome--3*[{chrome}]]
    |       |--nacl_helper_boo
    |--console-kit-dae--64*[{console-kit-da}]
    |--cron
    |--cupsd
    |--2*[dbus-daemon]
    |--dbus-launch
    |--dhclient6
    |--dhcpcd
```

```

|—gconfd-2
|—gdm—gdm-simple-slav—Xorg
|                                     |—gdm-session-wor—gnome-session—applet.py
|                                     |                                     |—bluetooth-appl+
|                                     |                                     |—evolution-alar+
|                                     |                                     |—gdu-notificati+
|                                     |                                     |—gnome-do—gno+
|                                     |                                     |—gnome-panel—+
|                                     |                                     |—gnome-power-ma+
|                                     |                                     |—gnome-volume-c+
|                                     |                                     |—gpk-update-ico+
|                                     |                                     |—metacity—{me+
|                                     |                                     |—nautilus—{na+
|                                     |                                     |—nm-applet—{n+
|                                     |                                     |—polkit-gnome-a+
|                                     |                                     |—tracker-miner-+
|                                     |                                     |—tracker-store-+++
|                                     |                                     |—2*[{gnome-sess+
|                                     |—{gdm-session-wo}
|                                     |—{gdm-simple-sla}
|                                     |—{gdm}
|—gedit—{gedit}
|—gnome-keyring-d—4*[{gnome-keyring-}]
|—gnome-screensav
|—gnome-settings-—{gnome-settings}
|—gnome-terminal—bash—su—bash—pstree
|                                     |—gnome-pty-helpe
|                                     |—2*[{gnome-terminal}]
|—gvfs-afc-volume—{gvfs-afc-volum}
|—gvfs-fuse-daemo—3*[{gvfs-fuse-daem}]
|—gvfs-gdu-volume
|—gvfs-gphoto2-vo
|—gvfsd
|—gvfsd-burn

```

```

├─gvfsd-metadata
├─gvfsd-trash
├─main-menu──{main-menu}
├─master├─pickup
│       └─qmgr
├─6*[mingetty]
├─nscd
├─polkit-gnome-au──{polkit-gnome-a}
├─polkitd──{polkitd}
├─pulseaudio├─gconf-helper──{gconf-helper}
│           └─2*[{pulseaudio}]
├─rpcbind
├─rsyslogd──4*[{rsyslogd}]
├─rtkit-daemon──2*[{rtkit-daemon}]
├─smpppd
├─startpar
├─tracker-extract──{tracker-extrac}
├─udev──2*[udev]
├─udisks-daemon├─udisks-daemon
│              └─2*[{udisks-daemon}]
└─upowerd──{upowerd}

```

Options de la commande pstree

```

opensuse:~ # pstree --help
pstree: unrecognized option '--help'
Usage: pstree [ -a ] [ -c ] [ -h | -H PID ] [ -l ] [ -n ] [ -p ] [ -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, --vt100        use VT100 line drawing characters
-l, --long         don't truncate long lines
-n, --numeric-sort sort output by PID
-p, --show-pids    show PIDs; implies -c
-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 :

```

top - 13:44:53 up 6:45, 3 users, load average: 0.02, 0.07, 0.09
Tasks: 134 total, 1 running, 133 sleeping, 0 stopped, 0 zombie
Cpu(s): 2.7%us, 1.7%sy, 0.0%ni, 95.7%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 1012320k total, 718048k used, 294272k free, 94588k buffers
Swap: 2048280k total, 0k used, 2048280k free, 348036k cached

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
9906	trainee	20	0	198m	73m	21m	S	3.6	7.4	6:52.46	chrome
9914	trainee	20	0	60684	14m	11m	S	0.7	1.5	0:17.92	gnome-terminal
1093	root	20	0	174m	36m	11m	S	0.3	3.7	4:16.23	Xorg
2387	root	20	0	6772	520	336	S	0.3	0.1	0:05.00	VBoxService
1	root	20	0	2216	724	620	S	0.0	0.1	0:00.98	init
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	20	0	0	0	0	S	0.0	0.0	0:00.60	ksoftirqd/0

```

 6 root      RT   0      0      0      0 S   0.0  0.0   0:00.00 migration/0
 7 root      0 -20      0      0      0 S   0.0  0.0   0:00.00 cpuset
 8 root      0 -20      0      0      0 S   0.0  0.0   0:00.00 khelper
 9 root      0 -20      0      0      0 S   0.0  0.0   0:00.00 netns
10 root      20   0      0      0      0 S   0.0  0.0   0:00.09 sync_supers
11 root      20   0      0      0      0 S   0.0  0.0   0:00.00 bdi-default
12 root      0 -20      0      0      0 S   0.0  0.0   0:00.00 kintegrityd
13 root      0 -20      0      0      0 S   0.0  0.0   0:00.00 kblockd
14 root      0 -20      0      0      0 S   0.0  0.0   0:00.00 kacpid
15 root      0 -20      0      0      0 S   0.0  0.0   0:00.00 kacpi_notify
...

```

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

Help for Interactive Commands - procps version 3.2.8

Window 1:Def: Cumulative mode Off. System: Delay 10.0 secs; Secure mode Off.

```

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

f,o      . Fields/Columns: 'f' add or remove; 'o' change display order
F or O   . Select sort field
<,>     . Move sort field: '<' next col left; '>' next col right
R,H      . Toggle: 'R' normal/reverse sort; 'H' show threads
c,i,S    . Toggle: 'c' cmd name/line; 'i' idle tasks; 'S' cumulative time
x,y      . Toggle highlights: 'x' sort field; 'y' running tasks
z,b      . Toggle: 'z' color/mono; 'b' bold/reverse (only if 'x' or 'y')
u        . Show specific user only
n or #   . Set maximum tasks displayed

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

```

```
q          Quit
          ( commands shown with '.' require a visible task display window )
Press 'h' or '?' for help with Windows,
any other key to continue
```

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

Au lancement, le temps de rafraîchissement de la liste est de 3 secondes. Pour modifier ce temps à 1 seconde, appuyez sur la touche **s** puis la touche **1** et validez :

```
top - 13:45:54 up 6:46, 3 users, load average: 0.01, 0.06, 0.09
Tasks: 134 total, 1 running, 133 sleeping, 0 stopped, 0 zombie
Cpu(s): 2.1%us, 1.8%sy, 0.0%ni, 96.1%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 1012320k total, 718880k used, 293440k free, 94612k buffers
Swap: 2048280k total, 0k used, 2048280k free, 348032k cached
Change delay from 3.0 to: 1
...
```

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

```
top - 13:46:31 up 6:46, 3 users, load average: 0.00, 0.05, 0.08
Tasks: 134 total, 1 running, 133 sleeping, 0 stopped, 0 zombie
Cpu(s): 2.2%us, 1.5%sy, 0.0%ni, 95.6%id, 0.0%wa, 0.0%hi, 0.7%si, 0.0%st
Mem: 1012320k total, 720668k used, 291652k free, 94620k buffers
Swap: 2048280k total, 0k used, 2048280k free, 348028k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
9906	trainee	20	0	199m	74m	21m	S	3.7	7.5	6:56.30	chrome
9772	trainee	20	0	365m	60m	32m	S	0.0	6.1	2:08.50	chrome
1093	root	20	0	174m	36m	11m	S	0.0	3.7	4:17.83	Xorg
10019	trainee	20	0	170m	32m	20m	S	0.0	3.3	0:05.94	chrome
9823	trainee	20	0	164m	27m	18m	S	0.0	2.8	0:01.39	chrome
6113	trainee	20	0	75048	22m	12m	S	0.0	2.3	0:06.68	gnome-do
6029	trainee	20	0	84028	22m	15m	S	0.0	2.2	0:16.01	nautilus

```

11508 trainee 20 0 92212 19m 14m S 0.0 2.0 0:11.64 gedit
6045 trainee 20 0 73504 19m 11m S 0.0 1.9 0:01.08 tomboy
6015 trainee 20 0 74508 18m 13m S 0.0 1.8 0:26.17 gnome-panel
11505 trainee 20 0 47700 17m 13m S 0.0 1.8 0:01.32 application-bro
6041 trainee 20 0 52680 17m 13m S 0.0 1.7 0:07.62 main-menu
6120 trainee 20 0 34648 15m 9224 S 0.0 1.6 0:01.53 applet.py
9914 trainee 20 0 60684 14m 11m S 0.0 1.5 0:18.45 gnome-terminal
5938 trainee 20 0 112m 14m 10m S 0.0 1.4 0:06.03 gnome-settings-
6116 trainee 20 0 43760 13m 10m S 0.0 1.3 0:00.72 gpk-update-icon
6140 trainee 20 0 42388 12m 9.9m S 0.0 1.3 0:01.18 gnome-power-man
...

```

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

```

top - 13:47:42 up 6:47, 3 users, load average: 0.08, 0.06, 0.08
Tasks: 133 total, 1 running, 132 sleeping, 0 stopped, 0 zombie
Cpu(s): 6.2%us, 2.8%sy, 0.1%ni, 90.9%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 1012320k total, 719736k used, 292584k free, 94636k buffers
Swap: 2048280k total, 0k used, 2048280k free, 348032k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
11584 root        20   0   2516   988   736 R   0.1   0.1   0:00.14 top

```

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

Options de la commande top

```

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

```

Les commandes fg et bg

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

```
# sleep 9999 &
```

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

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

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

```
opensuse:~ # sleep 9999 &
[1] 11621
opensuse:~ # jobs -l
[1]+ 11621 Running                  sleep 9999 &
```

<note important> Notez que le numéro de tâche est indiqué entre [crochets] tandis que le PID ne l'est pas. Le signe + qui suit le numéro de tâche [1] indique que la tâche est la dernière à avoir été manipulée. </note>

Si on souhaite envoyer un processus en arrière plan de façon à libérer le shell pour d'autres commandes, il faut d'abord suspendre le processus en question. Normalement on suspend un processus en utilisant la combinaison de touches **CtrlZ**.

Par exemple :

```
opensuse:~ # 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 :

```
opensuse:~ # bg %2
[2]+  sleep 1234 &
opensuse:~ # jobs -l
[1]- 11621 Running          sleep 9999 &
[2]+ 11629 Running          sleep 1234 &
```

<note important> Notez que lors du passage en arrière plan, le processus reprend son exécution normalement. Le caractère - qui suit le numéro de tâche [1] indique que la tâche est l'avant-dernière à avoir été manipulée. </note>

Pour ramener le processus en avant plan, il faut de nouveau interrompre le processus concerné. Or cette fois-ci, nous ne pouvons pas utiliser la commande `CtrlZ`. Il faut donc envoyer un **signal** au processus en utilisant la commande **kill** avec l'opérateur **-stop**.

```
opensuse:~ # kill -stop %2

[2]+  Stopped                sleep 1234
opensuse:~ # jobs -l
[1]- 11621 Running          sleep 9999 &
[2]+ 11629 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** :

```
opensuse:~ # kill -cont %2
opensuse:~ # jobs -l
[1]- 11621 Running          sleep 9999 &
[2]+ 11629 Running          sleep 1234 &
```

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

```
opensuse:~ # kill -stop %2

[2]+  Stopped                sleep 1234
opensuse:~ # jobs -l
[1]- 11621 Running          sleep 9999 &
[2]+ 11629 Stopped (signal)  sleep 1234
```

```
opensuse:~ # fg %2
sleep 1234
^C
opensuse:~ #
```

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

Options de la commande jobs

Les options de la commande jobs sont :

```
opensuse:~ # 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 :

```
opensuse:~ # jobs -l
[1]+ 11621 Running                  sleep 9999 &
opensuse:~ # wait %1
^C
opensuse:~ # jobs -l
[1]+ 11621 Running                  sleep 9999 &
```

<note important> Notez que l'utilisation des touches `CtrlC` tue le processus généré par la commande **wait** et non le processus généré par la commande **sleep**. </note>

La commande nice

Cette commande affiche ou modifie la priorité d'un processus. La priorité par défaut de nice est 10. La valeur de nice la plus prioritaire est -20. La valeur la moins prioritaire est 19.

Pour consulter la valeur de nice, nous pouvons lancer la commande nice en utilisant nice pour contrôler sa priorité :

```
opensuse:~ # nice nice
10
```

Dans ce deuxième cas, nous pouvons lancer la commande nice -n 3 sur le lancement de nice lui-même en contrôlant la priorité avec nice :

```
opensuse:~ # nice nice -n 3 nice
13
```

Vous constaterez que la valeur de la priorité est de 13.

En lançant une commande nice avec une valeur supérieure de 19, vous constaterez que la valeur retenue est de 19, soit la valeur maximale permise :

```
opensuse:~ # nice -n 25 nice
19
```

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

Options de la commande

Les options de cette commande sont :

```
opensuse:~ # nice --help
Usage: nice [OPTION] [COMMAND [ARG]...]
Run COMMAND with an adjusted niceness, which affects process scheduling.
With no COMMAND, print the current niceness.  Nicenesses range from
-20 (most favorable scheduling) to 19 (least favorable).
```

```
-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.

Report nice bugs to bug-coreutils@gnu.org

GNU coreutils home page: <http://www.gnu.org/software/coreutils/>

General help using GNU software: <http://www.gnu.org/gethelp/>

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.

```
opensuse:~ # jobs -l
[1]+ 11621 Running                  sleep 9999 &
opensuse:~ # renice +5 11621
11621 (process ID) old priority 0, new priority 5
```

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

Options de la commande

Les options de cette commande sont :

```
opensuse:~ # renice --help
```

Usage:

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

Options:

-g, --pgrp <id>	interpret as process group ID
-h, --help	print help
-n, --priority <num>	set the nice increment value
-p, --pid <id>	force to be interpreted as process ID
-u, --user <name id>	interpret as username or user ID
-v, --version	print version

```
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 :

```
opensuse:~ # 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.

```
Report nohup bugs to bug-coreutils@gnu.org
GNU coreutils home page: <http://www.gnu.org/software/coreutils/>
General help using GNU software: <http://www.gnu.org/gethelp/>
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** :

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

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

Parmi les numéros de signaux les plus utiles on trouve :

Numéro	Description
-1	Le signal Hang Up est envoyé à tous les enfants d'un processus quand il se termine
-2	Interruption du processus - équivalent à <code>CtrlC</code>

Numéro	Description
-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

~~DISCUSSION:off~~

Donner votre Avis

```
{(rater>id=opensuse_11_l112|name=cette page|type=rate|trace=user|tracedetails=1)}
```