

Préparation LPI

Exam 101

103.5 Gestion des processus

- Poids : 4
- Exécuter des travaux en avant et arrière plan
- Signifier à un programme de continuer après une déconnexion
- Superviser les processus actifs
- Sélectionner et trier les processus à l'affichage
- Envoyer des signaux aux processus

- Avant plan (foreground)
 - Lorsque qu'une commande est lancée, le shell ne rend pas la main et attend qu'elle se termine
 - Pour interrompre la commande en cours avant sa fin normale : « Ctrl-C »
- Arrière plan (background)
 - Pour lancer une commande et récupérer la main immédiatement, il faut la lancer en tâche de fond (background) en terminant la ligne de commande avec le caractère « & »
« \$ commande & »

- Passage d'avant en arrière plan
- Pour passe en background une commande lancée en avant-plan
 - 1 – figer la commande en cours avec « `Ctrl-Z` »
 - 2 – demander à la commande de passer en arrière-plan avec la commande « `bg` »
- Attention : à l'utilisation de « `Ctrl-Z` ». Les commandes ne sont pas tuées mais uniquement figées et donc continuent à occuper des ressources mémoire

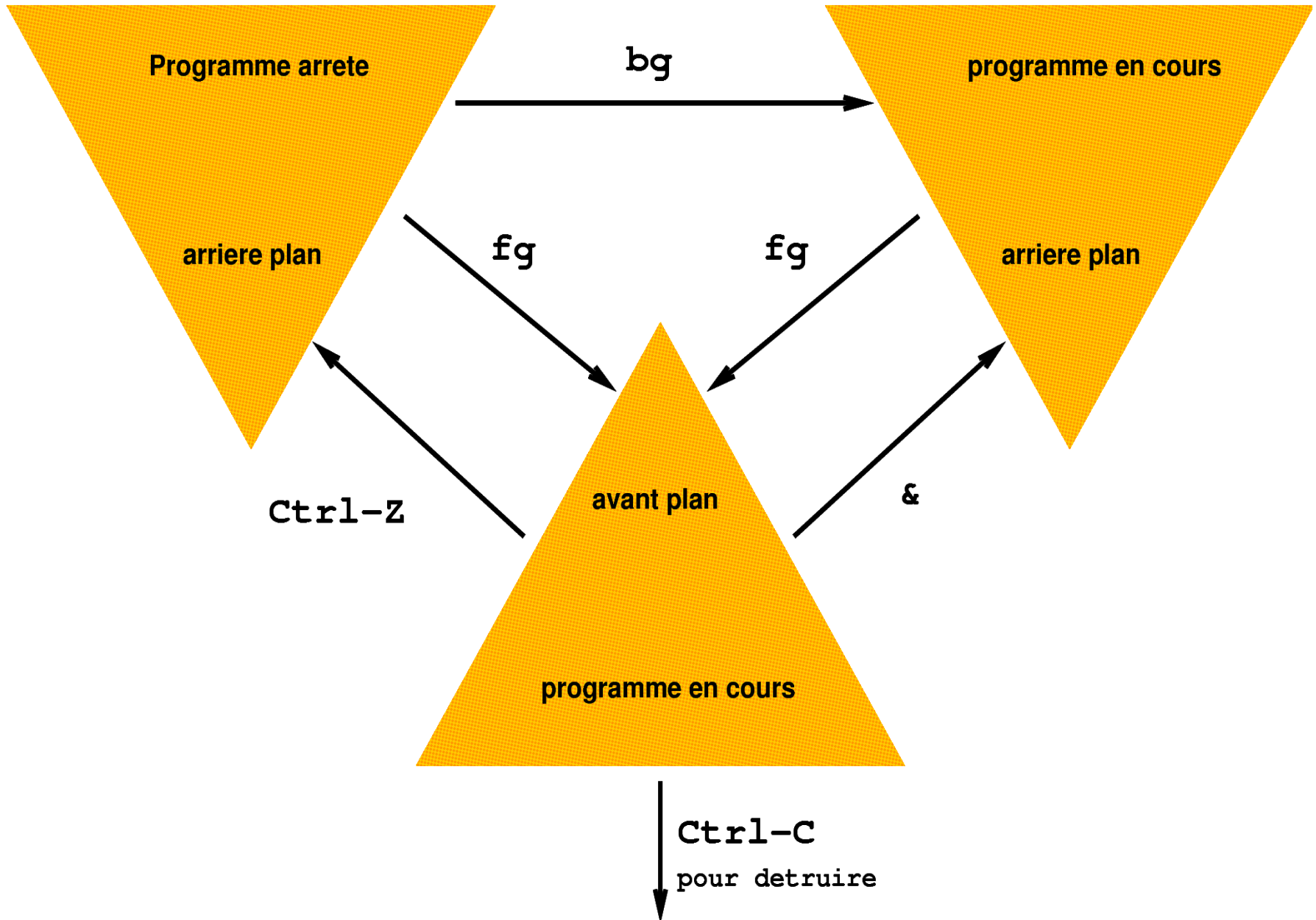
```
[aoi@test]$ find / -name Hostidfile.txt -print  
find: /lost+found: Permission non accordée
```

```
[1]+  Stopped                  find / -name Hostidfile.txt  
-print  
[aoi@test]$ bg  
[1]+  find / -name Hostidfile.txt -print &
```

- Passage d'arrière en avant plan
- Pour passer en foreground une commande lancée en arrière-plan
 - 1 – demander à la commande de passer en avant-plan avec la commande « fg »

```
[aoi@test]$ find / -name Hostidfile.txt -print  
find: /lost+found: Permission non accordée
```

```
[1]+  Stopped                  find / -name Hostidfile.txt  
-print  
[aoi@test]$ bg  
[1]+  find / -name Hostidfile.txt -print &  
[aoi@test]$ fg  
find / -name Hostidfile.txt -print  
find: /boot/lost+found: Permission non accordée  
find: /etc/skel/tmp: Permission non accordée  
....
```



- Liste des processus en arrière-plan
- Pour connaître la liste : « jobs »
- Les jobs sont numérotés (il peut y avoir plusieurs commandes lancées simultanément en arrière-plan). Ce numéro peut être repris dans les commandes « fg », « bg » et « kill »

```

XTerm vt100
% ./commande1 &
[1] 20776
% ./commande2 &
[2] 20778
% ./commande3 &
[3] 20780
% jobs
[1]  Running      ./commande1 &
[2]- Running      ./commande2 &
[3]+ Running      ./commande3 &
%

XTerm vt100
% jobs
%
  
```

- Options de la commande jobs
 - -l : ajoute le PID
 - -p : affiche uniquement le pid
 - -r : affiche uniquement les processus en cours d'exécution
 - -s : affiche uniquement les processus arrêtés

- Tuer un processus en arrière-plan
- commande : « `kill %num-job` »

```
[aoi@test]$ find / -name Hostidfile.txt -print  
find: /lost+found: Permission non accordée
```

```
[1]+  Stopped                  find / -name Hostidfile.txt -print  
[aoi@test]$ jobs  
[1]+  Stopped                  find / -name Hostidfile.txt -print  
[aoi@test]$ kill %1  
[aoi@test]$ jobs  
[1]+  Complété                 find / -name Hostidfile.txt -print
```

- Les commandes « `fg` », « `bg` » et « `kill` » ne fonctionnent que sur les processus lancés par le shell courant.
- Pour intervenir sur tous les processus il faut utiliser une commande plus générale de gestion de processus (voir plus loin) : « `ps` »

- Commande `nohup` : permet de continuer l'exécution d'une commande après une déconnexion en la rendant insensible aux signaux
`nohup COMMAND [ARG] ...`
- Si `stdin` est un terminal, redirection implicite par `/dev/null`
Equivalent à
`nohup COMMAND < /dev/null`
- Si `stdout` est un terminal, redirection implicite vers `nohup.out`.
`Stderr` est redirigée vers `stdout`.
Sinon faire une redirection explicite :
`nohup COMMAND > monfic.log`
- Attention : il faut explicitement mettre la commande en `background` si nécessaire

- A compléter
mettre un ex de commande nohup sur mkfs long par exemple

- Commandes `ps` : affiche l'état des processus en cours. Affichage de l'état au moment de l'exécution de la commande. Pour un affichage en continu, utiliser la commande `top`.
- `ps` récupère les informations affichées depuis le système de fichier `/proc`
- Liste des informations présentées
 - UID : uid propriétaire
 - PID : identifiant du processus
 - PPID : pid du processus parent
 - NI : valeur de nice (gentillesse)
 - SZ : (size) empreinte mémoire du processus (code + données + pile)

- Liste des informations présentées ...
 - STAT : état du processus
 - R (runnable) : prêt à être exécuté
 - S (sleeping) : endormi
 - D : sommeil ininterrompible
 - T (traced) : tracé ou suivi
 - Z : processus zombie
 - TTY : terminal associé
 - CMD : commande liée au processus

- Liste des options

Attention : infos différentes entre page de manuel et la réalité des options. Faire un ps -?

Comportement des options différent selon que l'on emploie le tiret ou non

- -A : tous les processus avec affichage simple (PI, TTY, TIME, CMD)
- -a : tous les processus avec un terminal associé
- a : tous les processus avec un terminal associé pour tous les utilisateurs
- r : affiche seulement les processus en cours d'exécution

- Liste des options (format de sortie)...
 - -l : format long
 - -f : format full
 - -F : format extra full
 - u : affiche l'utilisateur propriétaire et l'heure de lancement
 - e : affiche l'environnement d'exécution
 - -w ou w, -ww ou ww : affichage sur plusieurs lignes
 - f (forest) : affiche l'arbre généalogique des processus
 - -n : infos numériques (UID)

- Liste des options (tri)...
 - `O[+|-]k1[, [+|-]k2[, ...]]` : Permet de trier les résultat en fonction de clés de tri (k1, k2, ...)
+ et - permettent de définir un ordre alphabétique ou un ordre inverse
ps -ef O1,-2
 - `--sortX[+|-]key[, [+|-]key[, ...]]` : même fonction que l'option O mais key représente cette fois-ci le nom de la colonne (ppid par exemple)
 - Les clés de tri sont listées dans le man :
 - p : pid
 - P : ppid
 - u : nom d'utilisateur
 - U : UID

- Attention ! Ces deux commandes ne donnent pas le même résultat

\$ ps -ef t

UID	PID	PPID	C	STIME	TTY	STAT	TIME	CMD
root	1	0	0	21:40	?	Ss	0:01	/sbin/init
root	2	0	0	21:40	?	S<	0:00	[kthreadd]
root	3	2	0	21:40	?	S<	0:00	[migration/0]
root	4	2	0	21:40	?	S<	0:00	[ksoftirqd/0]

\$ ps -e ft

PID	TTY	STAT	TIME	COMMAND
2	?	S<	0:00	[kthreadd]
3	?	S<	0:00	_ [migration/0]
4	?	S<	0:00	_ [ksoftirqd/0]
5	?	S<	0:00	_ [watchdog/0]
6	?	S<	0:00	_ [events/0]

```
$ ps O+p
```

```

PID TTY      STAT   TIME COMMAND
3182 pts/0    Ss     0:00 bash
3234 pts/1    Ss+    0:00 bash
3270 pts/0    S+     0:00 man ps
3273 pts/0    S+     0:00 sh -c (cd "/usr/share/man/fr" && (echo ".pl 1100i"; /
3274 pts/0    S+     0:00 sh -c (cd "/usr/share/man/fr" && (echo ".pl 1100i"; /
3281 pts/0    S+     0:00 /usr/bin/less -is
3407 pts/2    Rs     0:00 bash
3549 pts/2    R+     0:00 ps O+p

```

```
$ ps O-p
```

```

PID TTY      STAT   TIME COMMAND
3553 pts/2    R+     0:00 ps O-p
3407 pts/2    Ss     0:00 bash
3281 pts/0    S+     0:00 /usr/bin/less -is
3274 pts/0    S+     0:00 sh -c (cd "/usr/share/man/fr" && (echo ".pl 1100i"; /
3273 pts/0    S+     0:00 sh -c (cd "/usr/share/man/fr" && (echo ".pl 1100i"; /
3270 pts/0    S+     0:00 man ps
3234 pts/1    Ss+    0:00 bash
3182 pts/0    Ss     0:00 bash

```

- Commande `ps tree` : affiche les processus sous une forme arborescente.

```
ps tree [-a] [-c] [-h|-Hpid] [-l] [-n] [-p] [-u]
[-Z] [-A|-G|-U] [pid|utilisateur]
```

Si le pid est omis, la racine de l'arbre est le processus init (pid=1)

Si l'utilisateur est précisé, seuls les processus de cet utilisateurs seront affichés

- Options :
 - -l : afficher les lignes longues
 - -n : tri numérique des processus ayant le même ancêtre plutôt que par leur nom
 - -p : ajoute le PID à l'affichage
 - -u : affiche les transition d'UID si l'UID du père est différent de celui du fils

```
$ pstree
init─┬─NetworkManager─┬─dhclient
      │                 └─{NetworkManager}
      └─acpid
         atd
         auditd─┬─audispd─┬─sedispatch
                  │       └─{audispd}
                  └─{auditd}
         avahi-daemon──avahi-daemon
         bluetoothd
         bonobo-activati──{bonobo-activati}
.....
```

```
$ echo $$
3234
$ pstree 3234
bash──pstree
```

- Commande `top` : affiche en temps réel les processus en cours d'exécution ainsi qu'un résumé des ressources système
`top -hv | -bcHisS -d delay -n iterations -p pid
[, pid ...]`
- Beaucoup d'options qui modifient les infos affichées, les délais de rafraîchissement,....
- Champs affichés
 - PID : identifiant du processus
 - PPID : PID du processus parent
 - USER : nom du propriétaire du processus
 - PR : priorité de la tâche
 - NI : valeur de gentillesse

- Champs affichés ...
 - VIRT : empreinte mémoire de la tâche (RES + SWAP)
 - RES : empreinte mémoire de la tâche
 - %CPU et %MEM : respectivement les pourcentage d'utilisation cpu et mémoire utilisés
 - TIME et TIME+ : temps d'exécution de la tâche
 - COMMAND : nom de la tâche

- Mode interactif
 - <Entrée> ou <Espace> : rafraîchissement de l'affichage
 - <?> ou <h> : aide
 - u : saisie d'un nom d'utilisateur ou UID → affichage des tâches possédées par cet utilisateur
 - k : saisie d'un PID → envoi du signal SIGTERM
 - q : quitter top
 - r : renice d'une tâche


```
top - 22:55:09 up 54 min, 2 users, load average: 0.00, 0.00, 0.00
Tasks: 188 total, 2 running, 140 sleeping, 0 stopped, 46 zombie
Cpu(s): 1.6%us, 2.0%sy, 0.0%ni, 96.4%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 1277124k total, 722968k used, 554156k free, 33248k buffers
Swap: 2441840k total, 0k used, 2441840k free, 358628k cached
```

```
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
950 root 20 0 36264 12m 6784 S 1.7 1.0 0:48.01 Xorg
756 haldaemo 20 0 6120 3972 3304 S 0.7 0.3 0:01.32 hald
1756 franck 20 0 17964 5312 3936 S 0.3 0.4 0:00.25 gnome-screensav
1880 franck 20 0 246m 110m 63m S 0.3 8.8 0:57.22 soffice.bin
1988 franck 20 0 38180 12m 9636 S 0.3 1.0 0:01.92 gnome-terminal
2009 franck 20 0 2472 1208 884 R 0.3 0.1 0:18.30 top
 1 root 20 0 2532 1528 1128 S 0.0 0.1 0:01.26 init
 2 root 15 -5 0 0 0 S 0.0 0.0 0:00.00 kthreadd
 3 root RT -5 0 0 0 S 0.0 0.0 0:00.00 migration/0
 4 root 15 -5 0 0 0 S 0.0 0.0 0:00.02 ksoftirqd/0
 5 root RT -5 0 0 0 S 0.0 0.0 0:00.00 watchdog/0
 6 root 15 -5 0 0 0 S 0.0 0.0 0:00.05 events/0
```

- Commande `uptime` : affiche la durée de fonctionnement d'un serveur depuis son dernier redémarrage
`uptime`
- Affiche la durée de fonctionnement, le nombre d'utilisateurs connectés, la charge du système des premières, 5 et 15 dernières minutes

```
$ uptime
```

```
23:21:48 up 1:21, 2 users, load average: 0.00, 0.00, 0.00
```

- **Commande free** : affiche la mémoire utilisée et disponible
`free [-b | -k | -m | -g] [-o] [-s delay] [-t] [-V]`
- **Options**
 - -b, -k, -m, -g : affichage en octet, kilo, mega, gigaoctets
 - -s sec : rafraîchissement de l'affichage toutes les « sec » secondes

\$ free

	total	used	free	shared	buffers	cached
Mem:	1277124	733560	543564	0	34740	364456
-/+ buffers/cache:		334364	942760			
Swap:	2441840	0	2441840			

- Commande `kill` : envoi d'un signal à un processus (SIGTERM par défaut)

```
kill [ -signal | -s signal ] pid ...  
kill [ -L | -V, --version  
kill -l [ signal ]
```

- Options
 - `-l` : liste des signaux disponibles et traduit les noms de signaux en valeur numérique et inversement

```
$ kill -l 9  
KILL
```

```
$ kill -l TERM  
15
```

- Avec un numéro de PID précédé d'un moins : envoi d'un signal à un groupe de processus (voir colonne PGID de la commande `ps`)
- Si PID = -1 : signifie tous les processus sauf init et kill lui-même

- Correspondance des signaux
 - SIGHUP \Leftrightarrow HUP \Leftrightarrow 1 : le plus souvent, relecture des fichiers de configuration
 - SIGINT \Leftrightarrow INT \Leftrightarrow 2 : Interrupt – Emis par Ctrl + C
 - SIGQUIT \Leftrightarrow QUIT \Leftrightarrow 3 : Quit – Emis par Ctrl + D
 - SIGKILL \Leftrightarrow KILL \Leftrightarrow 9 : le processus quitte sans faire les procédures de « nettoyage »
 - SIGTERM \Leftrightarrow TERM \Leftrightarrow 15 : le processus quitte après avoir fait les opérations de nettoyage

- Commande `killall` : envoi d'un signal à un ensemble de processus (SIGTERM par défaut) sur la base de leur noms

```
killall [-Z,--context pattern] [-e,--exact] [-g,--process-group] [-i,--interactive] [-q,--quiet] [-r,--regex] [-s,--signal signal] [-u,--user user] [-v,--verbose] [-w,--wait] [-I,--ignore-case] [-V,--version] [--] name ...  
killall -l  
killall -V,--version
```

- Options ...

- -i : demande confirmation avant chaque envoi de signal
- -r : sélection de processus sur la base d'expressions régulières
- -s : spécification du signal au lieu de SIGTERM par défaut le nom du signal doit être composé de SIG + nom du signal

Exemple :

```
killall -s SIGKILL httpd
```

- -u : sélection des processus détenus par un utilisateur particulier



- Commande top : comment interagir pour envoyer un signal à un processus sans quitter top
- Connaître les correspondances des 3 formes des signaux les plus courantes (9; KILL, SIGKILL)