TP7 Linux

Gestion des processus
Shell Bash (Ubuntu)

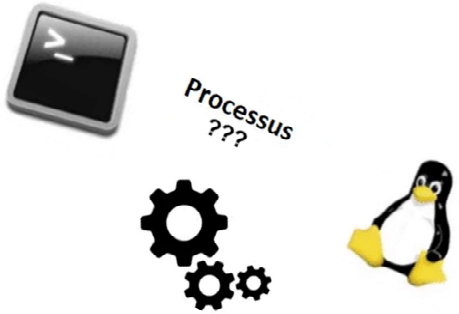
Contact : sihemmansour@yahoo.fr

Enseignante: MANSOUR Sihem

Plan du cours

- Définition d'un processus
- Cycle de vie d'un processus
- Commandes Shell appliquées sur les processus
- Exécution du processus en avant plan et en en arrière plan
- Envoie des signaux aux processus

- Définition d'un processus
- Cycle de vie d'un processus



Définition d'un processus

- Un processus est un programme en cours d'exécution.
- Exemple: L'exécution de l'application FireFox est un processus

```
essai22@essai22-VirtualBox:~$ firefox
```

- Chaque processus sous Ubuntu possède un identifiant unique PID (entier positif) qui permet de l'identifier
- Chaque processus possède aussi un père qui lui a donné naissance, l'identifiant du père d'un processus est PPID.

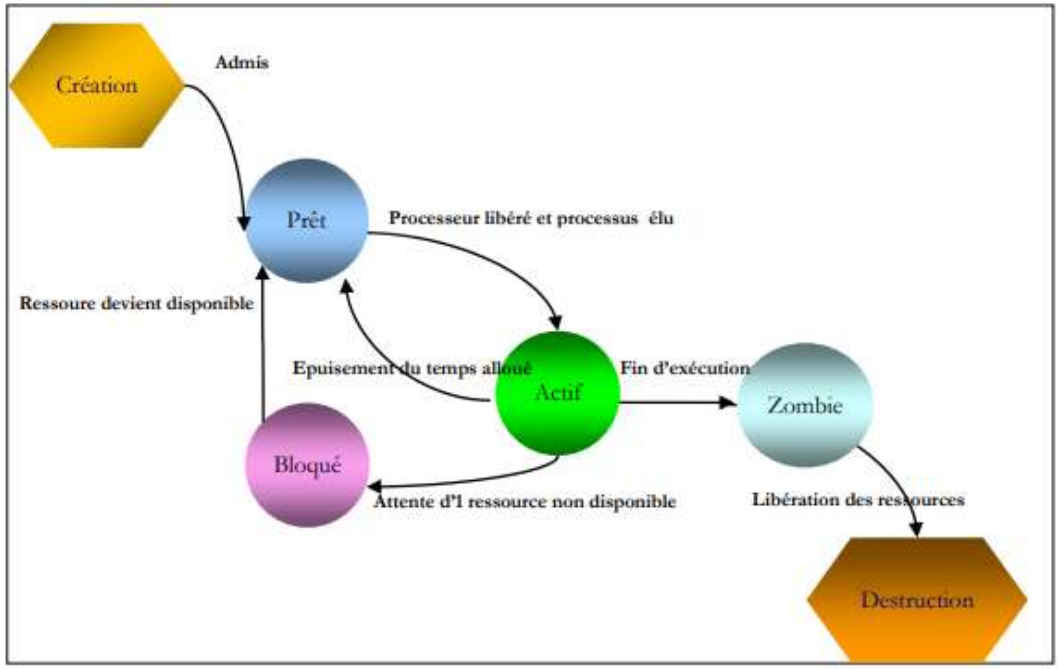
Le père de tous les processus lancés sous le système Ubuntu est le processus systemd ayant le PID 1

Définition d'un processus

- Le processus est caractérisé par l'identifiant de son propriétaire, en général (mais pas toujours) l'utilisateur qui l'a lancé : identifié sous le système par le terme: **user**
- Le processus possède un répertoire de travail, une priorité et un temps d'exécution ;etc.

Cycle de vie du processus

- Un processus possède un cycle de vie allant de sa naissance à sa mort passant par divers phases d'activité et d'attente. La figure suivante illustre le cycle de vie d'un processus dans le cadre d'un système mono processeur avec stratégie de partage de temps entre les processus



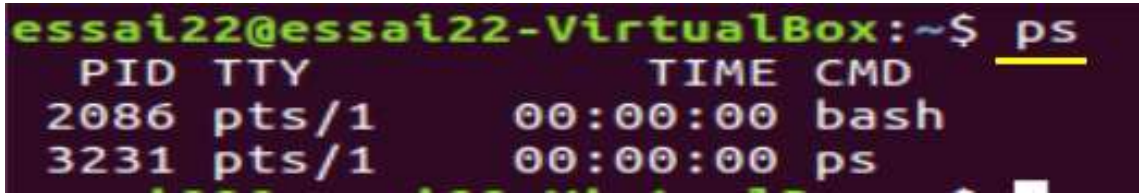
Cycle de vie du processus

- Lors de sa création, le processus passe par plusieurs étape:
- Actif sous le système, s'il s'exécute sur le processeur. Le terme qui représente cet état sous le système est: Runnable
- Prêt, il attend son tour pour s'exécuter sur le processeur. L'état actif et prêt sont transparents par à port à l'utilisateur, ils représente la gestion du système d'exploitation pour assurer le multitâche
- Bloqué, le processus est endormi. Il attend un evenement ou une ressource pour se réveiller et passe à l'état actif ou Prêt
- Zombie, un processus qui a terminé sa tâche et attend sa destruction

- Les commandes

Commande Shell appliquées sur les processus

- Pour avoir des informations sur les processus lancés dans la session de l'utilisateur connecté actuellement et qui sont en cours d'exécution, taper la commande ps. Voir l'exemple suivant:



- Les informations retournées par cette commande sont:

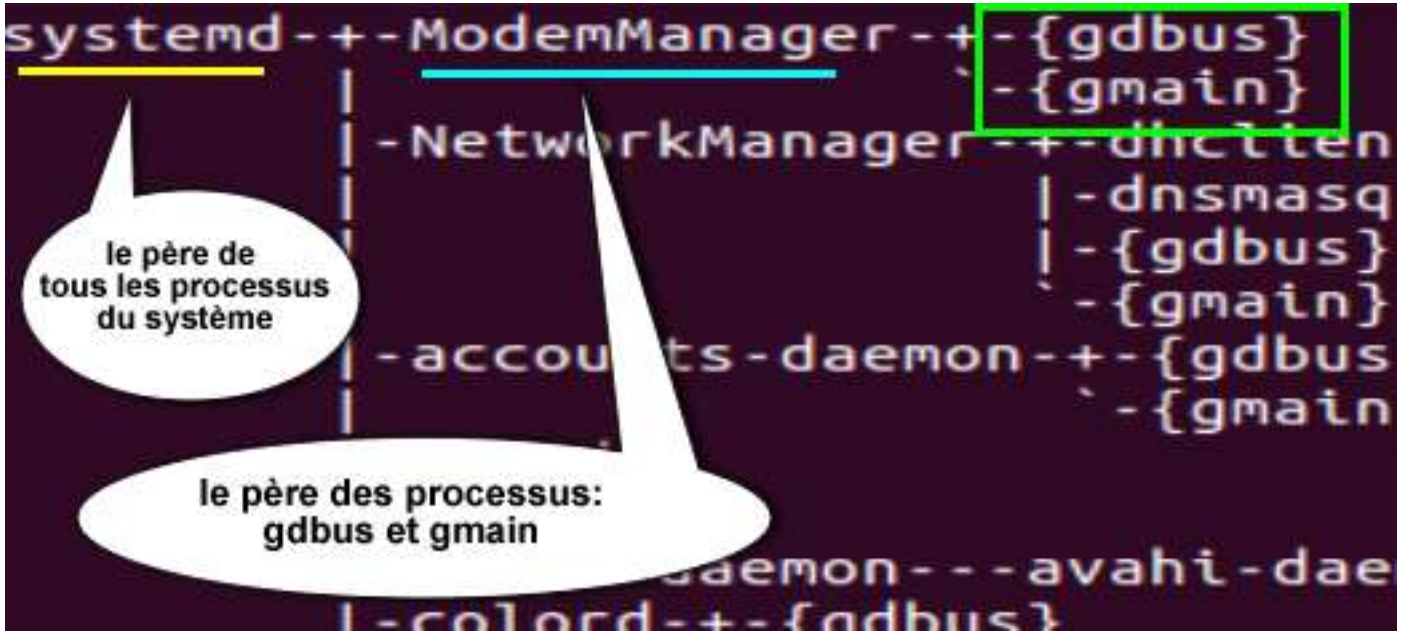
PID l'identifiant du processus , TTY: le terminal sur lequel est lancé la commande, TIME: le temps pris pour l'exécution et CMD est le nom de la commande qui a lancé la processus

Commande Shell appliquées sur les processus

- Pour avoir sur l'arborescence de parenté entre les processus, taper la commande suivante:

```
essai22@essai22-VirtualBox:~$ pstree|more
```

- Le résultat de cette commande est :



Commande Shell appliquées sur les processus

Pour afficher tous les processus lancés sous le système, taper la commande:

ps –ef

- L'option -e affiche tous les processus en cours d'exécution sous le système
- les options –f affichent des informations détaillées.

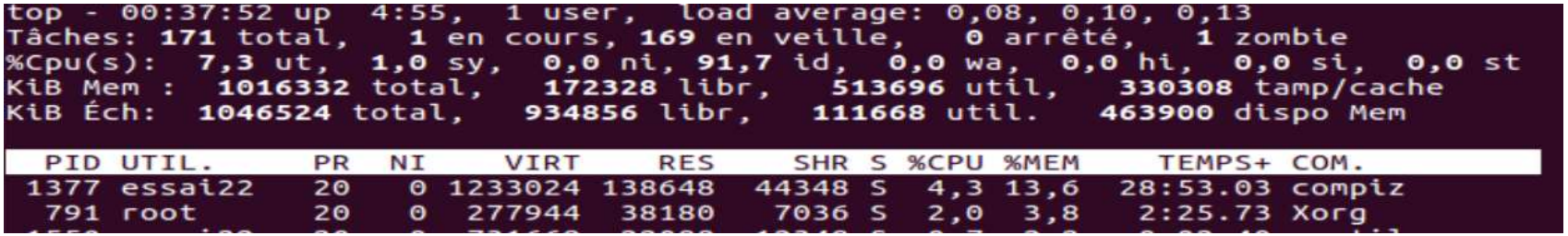
```
UID          PID    PPID  C STIME TTY          TIME CMD
root           1       0  0 oct.22 ?        00:00:03 /sbin/init spla
root           2       0  0 oct.22 ?        00:00:00 [kthreadd]
root           3       2  0 oct.22 ?        00:00:00 [ksoftirqd/0]
root           5       2  0 oct.22 ?        00:00:00 [kworker/0:0H]
root           7       2  0 oct.22 ?        00:00:02 [rcu_sched]
root           8       2  0 oct.22 ?        00:00:00 [rcu_bh]
root           9       2  0 oct.22 ?        00:00:00 [migration/0]
root          10       2  0 oct.22 ?        00:00:00 [watchdog/0]
root          11       2  0 oct.22 ?        00:00:00 [kdevtmpfs]
root          12       2  0 oct.22 ?        00:00:00 [netns]
root          13       2  0 oct.22 ?        00:00:00 [perf]
```

Commande Shell appliquées sur les processus

Pour afficher l'activité des processus sous le système en temps réél:

top

- Pour sortir de l'environnement top, taper la lettre q

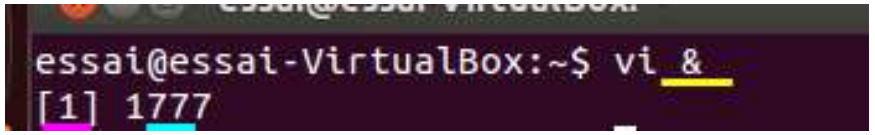


- Lancer les processus en arrière plan / avant plan

Lancer un processus en arrière plan/ avant plan

- Sous Shell, vous pouvez lancer des commandes en arrière plan si vous voulez libérer la console et faire un autre traitement en parallèle. Pour ce faire: lancer votre commande avec le symbole **&**

exemple:



- Lorsque vous lancez une commande en arrière plan: Shell vous retourne deux informations:

[1]:c'est le numéro du processus lancé en arrière-plan dans cette console on peut le nommer job. Il a pris le numéro 1 puisque il est le premier lancé en arrière plan

1777: le pid du processus

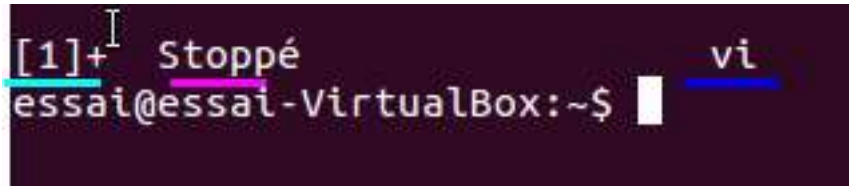
Mettre en pause l'exécution d'une commande

- Pour suspendre le traitement un processus, utiliser le raccourcis clavier

Ctrl+z

Exemple:

Après l'utilisation de ce raccourcis, le Shell vous envoie les informations suivantes:



Mettre en pause l'exécution d'une commande

[1]:c'est le numéro du processus en arrière-plan lancé dans cette console.

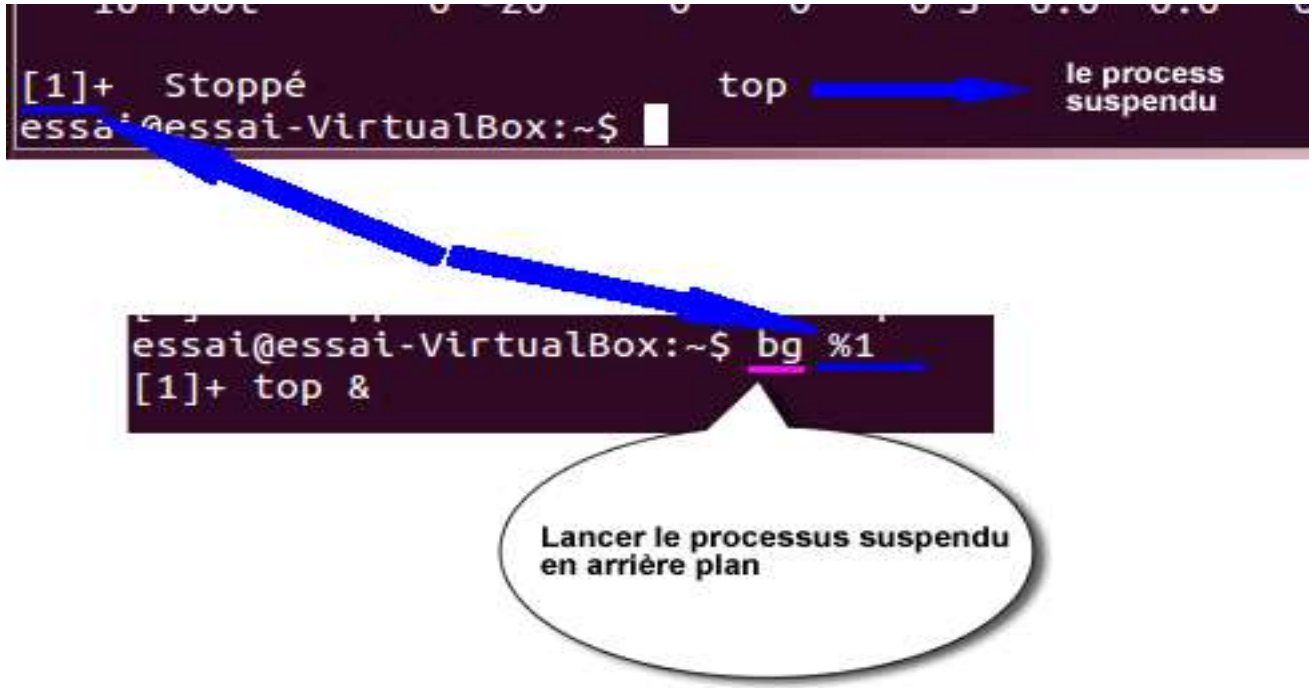
stoppé: son état sous le système

vi: le nom de la commande qui a lancé le processus

Remarque: Le processus de la commande vi est maintenant dans un état de pause. Il ne s'exécute pas mais reste en mémoire.

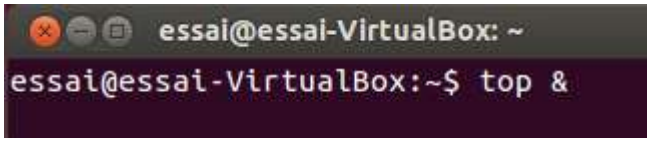
Passer le processus en arrière-plan

- Pour rendre un processus suspendu actif en arrière plan, vous pouvez utiliser la commande bg



Résumé

- Pour lancer un processus en arrière plan sous Shell, on peut utiliser deux manières différentes:
  - Lancer la commande avec le symbole **&**

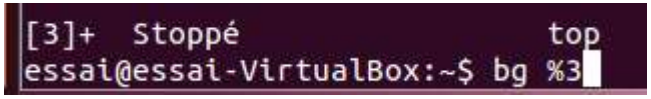


ou bien

  - Lancer la commande en avant plan

  - Suspendre son traitement en utilisant le raccourcis clavier **ctrl +z**

  - Puis lancer le processus suspendu en arrière plan en utilisant la commande bg avec le numéro de job:



Connaître les processus qui tournent en arrière plan

- Pour afficher tous les processus qui tournent en arrière plan, vous pouvez utiliser la commande **jobs.** Le graphique ci-dessous, Vous explique cette commande

Reprendre un processus au premier plan

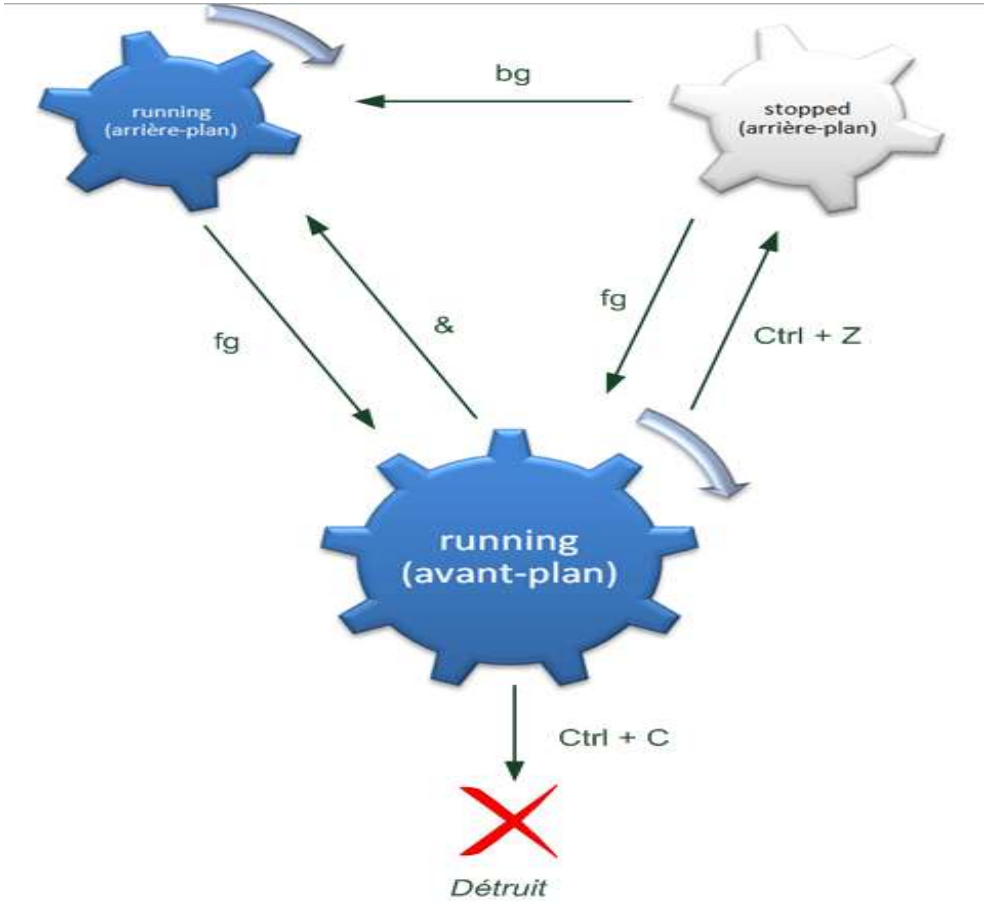
- Si vous avez un processus qui tourne en arrière plan et vous voulez rendre son exécution en premier plan. Utiliser la commande fg. Voir l'exemple ci-dessous

Resultat

```
top - 18:43:34 up 11 min,  1 user,  load average: 1.51, 1.08, 0.68
Tasks: 133 total,   4 running, 129 sleeping,   0 stopped,   0 zombie
Cpu(s):  7.6%us, 20.6%sy,  0.0%ni, 71.8%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:    507536k total,   489392k used,    18144k free,    50596k buffers
Swap:   522236k total,     1676k used,   520560k free,   200960k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 1401 essai     20   0 81448  16m  10m S  6.2  3.2   0:19.95 unity-panel-ser
 1487 essai     20   0 63100 4916 4076 R  5.9  1.0   0:17.29 hud-service
  925 root      20   0 94980  49m  10m S  3.6  9.9   0:12.50 Xorg
 1304 essai     20   0  6260 2564  616 S  3.0  0.5   0:09.35 dbus-daemon
 1347 essai     20   0  107m  23m  17m R  2.3  4.7   0:07.49 unity-2d-panel
 1412 essai     20   0 52312 4432 3612 S  2.0  0.9   0:07.46 indicator-appli
```

Les états de passage d'un processus du premier plan vers arrière plan et vice-versa



- Envoyer des signaux à un processus

envoyer des signaux aux processus

- La commande kill, Contrairement à ce que son nom semble indiquer, le rôle de cette commande n'est pas forcément de détruire ou de terminer un processus mais d'envoyer des signaux aux processus.
- Syntaxe : **kill [-s] -Num_signal PID [PID2...]**
- Un signal sera désigné soit par sa valeur numérique soit par son nom. Etant donné que les valeurs numériques peuvent varier d'un système à un autre, il vaut mieux utiliser le nom.
- Le tableau suivant liste les signaux les plus utilisés avec leur valeur sur un système Linux.

Exemple : $ kill -9 8 82

envoyer des signaux aux processus

Principaux signaux			
Signal	**Valeur numérique**	**Comportement par défaut**	**Description**
SIGINT	2	Terminer le *processus*	Il s'agit d'une demande venant du clavier, le plus souvent à l'aide la combinaison de touches Ctrl-C.
SIGKILL	9	Terminer le *processus*	Ce signal permet d'arrêter tout programme car il ne peut être géré différemment que le comportement par défaut. L'arrêt du programme est brutal.
SIGTERM	15	Terminer le *processus*	Si le programme n'a pas prévu de gérer ce signal, l'arrêt sera aussi brutal que pour SIGKILL. Mais comme le comportement par défaut peut être changé, le programme a la possibilité de réaliser des opérations avant de se terminer.
SIGCONT	18	Reprendre le *processus*	Permet de faire se continuer un *processus* qui avait été arrêté par exemple à l'aide de SIGSTOP (voir ci-après).
SIGSTOP	19	Arrêter le *processus*	Ce signal demande au *processus* de suspendre son exécution. Comme SIGKILL, ce signal ne peut être géré différemment.

9	SIGKILL	Terminaison immédiate du processus
15	SIGTERM	Terminaison propre du processus
18	SIGCONT	Reprise du processus
19	SIGSTOP	Suspension du processus

Exemple:

```
$ kill -9 1664
```

MERCI POUR VOTRE ATTENTION

Ens: MANSOUR Sihem