ADMINISTRATION SYSTEME LINUX
DISTRIBUTION UBUNTU

GESTION DES PROCESSUS :

I. NOTIONS GENERALES :

1. Rappel :

Un processus est un programme en cours d'exécution.
(Ça peut-être une ou un ensemble de commande).

Exemple :

`ls- al |grep cisse` → lance deux processus, un par commande.

A savoir ! Les processus sont gérés en mémoire.
Le fonctionnement d'un système Unix (linux) met donc en jeu un ensemble, un ensemble de processus.

2. Attributs d'un processus :

Un processus est représenté par un ensemble d'attributs qui forment ce qu'on appelle son environnement.
Les principaux attributs d'un processus sont :

1- le nom du processus,
2- le **PID (Process Id**entifier) **:** nombre unique affecté au processus pour l'identifier,
3- le **PPID (**Parent **PID**) **:** le PID du processus père,

> **NB :**
> Sous Unix, tout processus sauf **init** est crée par un autre processus alors appelé père du processus.

4- **UID (User Id**entifier**) :** UID de l'utilisateur propriétaire du processus,
5- **GID :** GID du groupe propriétaire du processus,
6- **l'état actuel du processus :** running (r) → actif
sleeping (s) → en veille
stopped → arrêté
7- **télétype (tty) d'exécution :** Le terminal depuis lequel le processus a été lancé.

NB :
Certain processus n'ont pas besoin de terminal : les démons.

8- **Les fichiers ouverts :** Les fichiers utilisés par le processus dans son exécution.
9- **La mémoire utilisée :** Nombre (ensemble) de page mémoire que le processus utilise ou a réservé pour son exécution.
10 **- TIME :** C'est le cumule du temps cpu pour l'exécution du processus.

3. Classification des processus :

On distingue deux types de processus :
- Les processus système ou démons (daemons en Anglais).
- Les processus utilisateurs.

Les processus système sont en général lancés au démarrage du système et attendent d'être sollicités. Ils assument des tâches d'ordres généraux. Ils ne sont pas attachés à des terminaux.

Exemple :

`init, ftpd` (file transfer protocol daemons)

Les processus utilisateurs correspondent à l'exécution de programmes utilisateur. Ils sont généralement rattachés à un terminal.

Exemple :

Le Shell de connexion.

4. Modes d'exécution d'un processus (Commande)

Il existe cinq (5) modes d'une commande (processus) :

- **Mode interactif :**
  Commande lancée à partir d'un interpréteur de commandes.
- **Le mode arrière-plan : (background)**
  L'utilisateur reprend immédiatement la main, après avoir lancé le processus. L'exécution d'une commande en arrière-plan se fait avec le signe **&** qui suit la commande. La commande lancé est alors appelée **job**.
- **Le mode différé :**
  L'exécution de la commande est déclenchée à une date fixée. On utilise la commande **at**.
- **Le mode batch :**
  La commande à exécuter est placée dans une file d'attente. Les commandes de la file sont exécutées séquentiellement. Ici c'est le

système d'exploitation qui décide du moment de l'exécution de la commande. (Efficace pour les systèmes chargés).

- **Le mode cyclique :**

La commande à exécuter est placée dans un fichier appelé crontab à l'aide de la commande crontab et son exécution se fait de façon cyclique.
Un démon appelé cron scrute le fichier crontab pour l'exécution des commandes qui s'y trouvent aux dates fixées.

Remarque :

- l'utilisation de la commande at nécessite une autorisation gérée par les fichiers **/etc/at.allow et/etc/at.deny**
- l'utilisation de crontab nécessite une autorisation gérée par les fichiers **/etc/cron. Allow et /etc/cron .deny**
- Ces fichiers sont généralement dans les distributions RedHat, donc à vérifier dans les autres.

II. QUELQUES COMMANDES DE GESTION (CONTROLE) DES PROCESSUS :

NB :
Un processus meurt lors que son père meurt (Fonctionnement normal dans le contexte Unix).

1. Commande nohup :

Elle permet de maintenir en vie un processus malgré la fin (mort)de son père

Exemple :

`nohup ls – R / &` (sortie dans nohup .out qui est créé)
`nohup ls – R / toto &` (sortie dans le fichiers toto)

Remarque :

En utilisant nohup dans une pipe (tube), il faut l'utiliser dans chaque commande de la pipe.

2. Commandes jobs, fg, bg :

Jobs :
Affiche la liste des processus en arrière plan pour la session actuelle.
Un numéro est alloué à chaque job (de 1 à N) et l'état du processus est précisé.

Ce numéro est souvent manipulé à la place du pu du processus.

fg (forground) : Amène un jobs au premier plan

Exemple :

fg % 3 amène le jobs numéro 3 au 1er plan.

bg (background) : Envoie un processus en background

3. Commande ps : (process status)

Elle fournit les informations sur les processus. Elle possède beaucoup d'option et est accessible à tous les utilisateurs. Sans option, ps fournit :

PID TTY TIME COMMANDE

Quelques options utiles :
- U : Présente la liste des utilisateurs ps u
- E : Affiche les renseignements sur tous les processus en cours.
- L : Format long (beaucoup d'informations)
- F : Affiche la généalogie (affiche du PPID)

4. Commande Kill :

Elle sert à envoyer un signal à un processus, qui agit en fonction de celui-ci.
Il existe une liste de signaux, fonction du système (kill -l) affiche cette liste.
Chaque signal a un nom et un numéro qui sont utilisés (l'un ou l'autre) par kill.
Kill utilise le PIP du processus ou son numéro job.

Exemples de signaux :

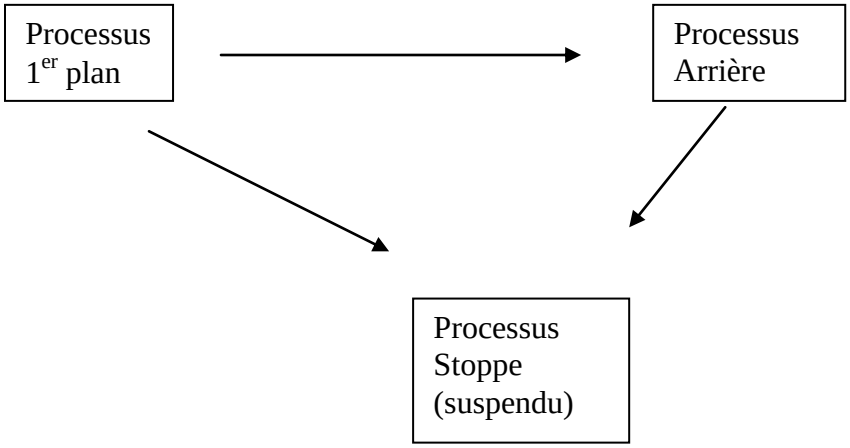
NOM	NUMERO	FONCTION
SIGHUP	1	Relecture de la configuration
SIGKILL	9	Tuer le processus
SIGSTOP	19	Stopper (suspender)
SIGCONT	18	Continuer
SIGTERM	15	Terminer

Remarque :

- les noms ou numéro des signaux sont à utiliser comme option de la commande kill.
- on peut se passer du préfixe SIG dans le nom du signal.

- kill sans option (ie sans précision du signal) envoie le signal TERM
- il est possible de définir le comportement d'un processus par rapport à un signal (prendre en compte ou non) sauf pour le signal KILL.
- seul le processus init peut ignorer le signal KILL. En d'autres termes, on ne peut pas tuer le processus init pour un processus tournant au premier plan, on a la possibilité de lui envoyer les 2 signaux suivants :

KILL (avec CTRC-C) (tuer)
STOP (avec CTRL-Z) (suspendre)



Exemple :

```
kill –1 1          (init relit la configuration)
Kill –9 1          (sans effet : on ne peut tuer init)
Kill –kill 2130    (tue le processus de PID 2130
Kill – 9   % 3     (tue le processus en bg dont le numéro de job est 3)
fg % 2             (envoie au premier plan le processus)
```

Il est possible de changer les fonctions de CTRL- 2 et CTRL – C à l'aide de la commande stty.

5. Commande at :

La commande at permet de lancer des actions à une heure donnée une date donnée, un mois donné et une année donnée mais sans répétition.

Syntaxe :

at heure : minute jour/mois/année

Exemple :

```
at 11 :00
```

`at now +1hour` : exécuter la tâche une heure à partir de cet instant
`at 15 :00 +1day` : exécuter une tâche un jour à compter de cette heure

REMARQUE :

L'exécution de la commande at affiche l'invité at> et on saisie les actions à exécutées et on termine par Ctrl+d et le système renvoi alors un numéro de jobs.

Quelques Options de at :

- **-l :** permet de donner la liste des taches programmées
- **-d :** supprime une tache spécifiée
- **atq :** qui affiche la file d'attente ou la liste des taches
- **atm :** supprime aussi une tache programmée
- **-c** : permet de voir ce que fait l'action

NB :
Certaines options de at marchent avec le numéro de job

Exemple :
`at –d 100`

Les commandes programmées par at sont enregistrées dans le répertoire **/var/spool/at.**
Les tâches programmées par at sont exécutées par le demon atd.
Avec la commande at il est possible d'ajouter des utilisateurs dans la liste de ceux qui peuvent exécuter la commande at.
nano /etc/at.deny.

6. Commande batch :

La commande batch permet d'exécuter les taches quand la charge du système le permet, c'est au système d'exploitation de décider du moment précis d'exécution, les travaux exécutés sous batch fonctionnent en mode background(en arrière plan)

Syntaxe :

```
batch
        Commande1…….
        Commande2…….
        Commande3……
Ctrl+d
```

REMARQUE :
Il est conseiller de mettre les commandes exécutées avec batch dans un fichier.

7. Commande crontab :

Par défaut chaque utilisateur peut se servir de la commande crontab pour définir des actions cycliques (régulières).
Le système **cron** permet la configuration des tables d'exécutions des taches périodiques (crontab) pour chaque utilisateurs, les tables des utilisateurs sont enregistrées dans le fichier **/var/spool/cron/nom_utilisateur.**

Ces fichiers sont gérés par l'utilisateur a l'aide de l'outil **crontab,** la table **cron** du système lui-même se trouve dans le fichier **/etc/crontab.**

- **Quelques options de crontab :**

  crontab –e :
  Permet de créer une **crontab,** c'est-à-dire ouvrir l'éditeur par défaut avec la table du **cron.**

  crontab -l :
  Permet d'afficher la liste de tous les fichiers **crontab.**

  crontab -u :
  Permet de voir les **crontab** des utilisateurs donnés.

  crontab -r :
  Supprime la table **cron**

Format d'un fichier crontab :

Un fichier crontab est au format textuel. Chaque ligne est formée de 6 champs avec l'espace comme séparateur de champ.
On a :

Minute	Heure	Jour du mois	mois	Jour de la semaine	Action à exécutée
0 - 59	0 - 23	1 - 31	1 - 12	0 - 6	cmd

Remarque :

- un champ peut comporter plusieurs valeurs à séparer par des virgules (,)
- le signe (-) permet de définir un intervalle

- les mois et les jours peuvent être donnés avec les abréviations anglaise jan (1) feb (2) /mon(1) sun(0)
- le signe * dans un champ sauf le dernier indique toute les valeurs du champ
- le / permet de spécifier une répétition par rapport à un champ.

Exemple : dans le champ minute (/5) signifie toutes les 5 minutes

Remarque :

La commande anacron permet de vérifier les crontab non exécutées et les exécute après la remise en route d'un système arrête

Exercice : (exemples de planification)

1- une action le 1er jour du mois à 1h

0	1	1	*	*	Action

2-Une action une fois par semaine le mardi à 12h 30

30	12	*	*	2	Action

3-Une action tous les 1er et 15 du mois à 0h

0	0	1,15	*	*	Action

4-Une action tous les 5 jours à 16h00

0	16	/5	*	*	Action

5- Une action tous les quinze 1er jour du mois à 4h 30

30	4	1-15	*	*	Action