

Gestion des processus et automatisation de tâches

Abdelali SAIDI

abdelali.saidi@gmail.com

Plan

- 1 Gestion de processus
- 2 Automatisation de tâches

Plan

- 1 Gestion de processus
- 2 Automatisation de tâches

Présentation

Définition

Un processus est une instance d'un programme en cours d'exécution.

Les états

Un processus peut avoir différents états pour définir son comportement. Les principaux états sont:

- “running” : le processus occupe le CPU.
- “sleeping” : le processus est en attente du CPU.
- “idle” : le processus est à l'écoute d'un signal pour basculer à l'état “sleeping”

Présentation

Le Processus Identifier

Quand un processus démarre, le SE lui affecte un identifiant numérique unique. Le PID.

Commandes

- ps : affiche des informations sur une sélection des processus actifs.
- top : offre une vue dynamique en temps réel d'un système démarré.
- pstree : affiche les processus en cours d'exécution sous forme d'un arbre
- jobs : affiche tous les processus lancés dans le shell courant
- nohup : permet au processus d'ignorer le signal SIGHUP

Présentation

Les champs de top et ps -aux

- PID : L'identifiant du processus
- USER : L'utilisateur propriétaire du processus
- PR : La priorité du processus (plus petit est le nombre, plus élevé est la priorité du processus)
- SIZE : La taille du processus dans la mémoire comprenant les données et le fichier
- RSS : La taille totale de mémoire utilisée par le processus
- SHARE : La taille de la mémoire partagée

Présentation

Les champs de top et ps -aux

- STAT :
 - D : deep sleep
 - S : sleep
 - R : running
 - T : stoped
 - Z : zombie (un processus qui s'est arrêté sans attendre l'arrêt de ses fils)
- LIB : Nombre de pages de mémoire prises par les bibliothèques
- %CPU : Le pourcentage occupé du processus au niveau du CPU
- %MEM : La taille de la mémoire physique prise par le processus
- TIME : La durée de vie du processus depuis son lancement
- COMMAND : La commande qui a lancé le processus

Les signaux

La commande Kill

Pour communiquer avec les processus, l'utilisateur ou le système utilise des signaux. Cela est faisable avec la commande kill

Options

kill -l affiche une liste des signaux disponibles (ils peuvent aussi être représentés par des nombres). On en trouve:

- SIGHUP : Le signal SIGHUP est envoyé à un processus quand le terminal qui l'a lancé est fermé
- SIGINT : ctrl-C (envoie une interruption)
- SIGKILL : Tuer le processus sans conditions
- SIGTERM : Termine l'exécution du processus d'une façon clean
- SIGSTP : ctrl-Z (arrête l'exécution et met le processus en état prêt)

background, foreground et détachement

Présentation

Quand un programme est exécuté, le shell maintient l'exécution de son processus. Si on doit attendre la fin de cette exécution pour avoir le prompt, on est dans le *mode foreground* sinon, le programme est lancé en *mode background*

Commandes

Pour lancer une commande en backgroun, on peut utiliser le caractère &

- Exemple : `$updatedb &`

background, foreground et détachement

Commandes

Pour passer l'exécution d'un processus d'un mode à un autre, on peut utiliser les commandes:

- fg N° du job : faire passer le processus au mode foreground
- bg N° du job : faire passer le processus au mode background
- ctrl-Z : faire passer le processus à l'état sleep

Le job

Le numéro du job est un identifiant que le shell affecte à un processus qu'il maintient. On peut citer les jobs d'un shell grâce à la commande *jobs*

background, foreground et détachement

Présentation

Un processus exécuté dans un shell se voit attaché à ce shell. Cela a pour conséquence l'interruption du processus en cas de fermeture du shell.

La commande disown

Pour que la fermeture d'un shell n'interrompt pas un processus, on peut utiliser la commande `disown` pour son détachement.

- La syntaxe : `disown PID`

La priorité

Présentation

La priorité d'un processus varie de -20 jusqu'à +19. Plus cette valeur est petite, plus le processus est prioritaire

Caractéristiques

- Seul le root peut affecter des priorités négatives
- Par default, un programme est d'une priorité 10

La priorité

Commandes

- nice : lance un processus avec une priorité ajusté (ce qui affecte l'ordonnancement des processus)
 - Option : -n (pour définir la valeur de la priorité à mettre)
- renice : Autorise la modification de la priorité d'un processus en cours d'exécution
 - Options : -u/-g/-p (pour définir l'utilisateur/groupe/PID concerné)

Exemples

- nice +12 cmd
- renice -10 -u ahmad

Plan

- 1 Gestion de processus
- 2 Automatisation de tâches

Présentation

Planification de tâches

Dans un système Linux, il est possible de planifier une tâche de façon à ce qu'elle se lance dans un moment précis, ou bien, d'une façon répétitive.

Cela est possible grâce à deux démons:

- atd : pour les tâches planifiées
- crond : pour les tâches récurrentes

Présentation

Utilisation

L'administrateur gère l'utilisation des ces deux précédant démons grâce aux fichiers :

- /etc/cron.allow
- /etc/cron.deny
- /etc/at.allow
- /etc/at.deny

règles

- Si aucun des fichiers précédant n'existe, seul le root a le droit d'utiliser ces services
- Si /etc/at.allow et /etc/cron.allow existent, seuls les utilisateurs qui s'y trouvent pourront utiliser les deux services
- Si /etc/at.deny et /etc/cron.deny existent, les utilisateurs qui s'y trouvent ne pourront pas utiliser les deux services

La commande crontab

Présentation

La commande crontab (chrono table) permet de créer (ou modifier) un fichier crontab ayant plusieurs demandes que l'utilisateur veut être exécuté. Ce fichier a le nom de l'utilisateur et il est enregistré dans le répertoire `/var/spool/cron` avant l'exécution par crond.

Options

- `-u` : Installation du fichier crontab de l'utilisateur spécifié
- `-e` : Modification du fichier crontab avec l'éditeur spécifié par la variable `EDITOR`
- `-l` : Afficher le contenu du fichier crontab courant

Le fichier crontab

Les champs

Le fichier crontab créé par l'utilisateur dispose de cinq champs pour préciser la période d'exécution et un sixième pour la commande à exécuter:

- minute (0 - 59)
- heure (0 - 23)
- date (1 - 31)
- mois (1 - 12)
- jours de la semaine (0 - 6) : le 0 définit dimanche

On peut raffiner la période en utilisant:

- , : pour séparer plusieurs valeurs
- - : pour spécifier un intervalle
- * : remplace toutes les valeurs possibles
- / : pour spécifier un cycle de temps (* / 5 toute les 5 minutes si cette valeur est placé dans le champs minute)

Le fichier système crontab

/etc/crontab

Le fichier `/etc/crontab` est également lu par le démon `crond` et a une syntaxe différente par rapport à un fichier ordinaire `crontab`.

```
$ cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 * * * * root    cd / && run-parts --report /etc/cron.hourly
25 6 * * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily
47 6 * * 7 root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly
52 6 1 * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthl
#
$ █
```

Le fichier système crontab

Caractéristiques

- il contient des informations sur l'environnement avant les entrées
- après les cinq premiers champs il y a le nom d'utilisateur propriétaire de la commande
- il est souvent utilisé pour démarrer les scripts qui sont placés dans les répertoires `/etc/cron.(hourly|daily|weekly|*)`
- la commande `run-part` prend en argument un répertoire et exécute tous les scripts qui s'y trouvent

La commande at

Présentation

L'outil at lit des commandes afin de pouvoir les lancer à un moment ultérieur. Ces commandes peuvent être spécifiées depuis l'entrée standard, comme ils peuvent être récupérée depuis un fichier.

Syntaxe

at *[spécification du temps]*

La commande at

Caractéristiques

- La spécification du temps : “now”, “3am + 2days”, “midnight”, “10:15 Apr 12”
- Les commandes programmées par at sont enregistrées dans /var/spool/at/
- Options:
 - at -l : donne la liste des tâches programmées (atq)
 - at -d “numéro de la tâche” : supprime la tâche spécifiée (atrm)

Exemple

```
$ at 6:00
warning: commands will be executed using /bin/sh
at> mail ahmad < lettre.txt
at> █
```