Faire un peu de crontab

Le démon **crond** est le processus qui permet d'exécuter des tâches planifiées automatiquement à des instants précis prévus à l'avance (date, heure, minute). Si par exemple, je veux que la machine de Thomas s'éteigne à 20h51mn le 15 décembre 2008, je peux le faire avec cron. Pour cela, il me faut une entrée dans le **cron** de la machine de Thomas qui dit que celle-ci doit s'étiendre à la date prévue. En général, cron est installé par défaut sur les systèmes, sinon pour l'installer utiliser le programme d'install de votre distro (aptitude ou yum).
La configuration des entrées de la cron se fait avec le programme **crontab** (tables de cron). C'est très facile à comprendre et à utiliser, mais il y'a une syntaxe à respecter et ce que nous allons décrire c'est juste cette petite syntaxe très simple à comprendre.

1- Fonctionnement Général :

Pour programmer l'exécution planifiée d'une tâche, il faut écrire une entrée dans la crontab. Cette entrée spécifiera la (ou les) date d'exécution de cette tâche, ainsi que la tâche à exécuter et d'autres paramètres si nécessaire.

Pour chaque utilisateur du système, on peut éditer un fichier de crontab relatif aux tâches planifiées de ce dernier. Ces fichiers se trouve en général dans « /var/spool/cron/crontabs/ » ou « /var/spool/cron/tabs » et portent le nom de l'utilisateur en question.
Pour éditer ces fichiers, il faut utiliser la commande **crontab**.
crontab -e // Permet d'éditer le fichier de crontab relatif à l'utilisateur connecté.
crontab -e -u root //Edite la crontab de l'utilisateur root.
crontab -l // Affiche juste la crontab le utilisateur connecté.
crontab -r // Efface la crontab de l'utilisateur
On peut éditer ces fichiers avec vi ou emacs, ... mais crontab qui connait la syntaxe correcte de ce genre de fichier fait des vérifications syntaxiques et signale les éventuelles erreurs qu'on peut faire.

En plus des fichiers de crontab des utilisateurs, le système dispose de ses propres fichiers de tâches planifiées dans les répertoires « /etc/cron.* » (* = d, daily, hourly, weekly, monthly).
Pour avoir une explication plus détaillée et sure du fonctionnement de ces répertoires, il faut visualiser le fichier « /etc/crontab ». Mais en général, tout ce qui se trouve dans :
- « /etc/cron.hourly » est exécuté une fois par heure,
- « /etc/cron.daily » une fois par jour,
- « /etc/cron.weekly » une fois la semaine,
- « /etc/cron.monthly » chaque mois.
- « /etc/cron.d » contient des fichiers au format crontab pour des choses plus spécifiques.

Je peux par exemple créer un fichier au format crontab que je place dans « /etc/cron.d » qui me permet d'exécuter des programmes tous les jours à 20h10mn ou tous les 1er du mois, ...
Donc la crontab, c'est l'ensemble de fichiers dans le répertoire « /etc/cron.d », plus les fichiers des utilisateurs de la forme « /var/spool/cron/crontabs/<nom_utilisateur> », plus le fichier « /etc/crontab » qui définit comment fonctionnent les répertoires « /etc/cron.* » (* != d).

2- Syntaxe de cron :

Chaque entrée de la crontab correspond à une tâche à exécuter. Et chaque entrée de la crontab est une ligne dans un des fichiers cités plus haut. Regardons donc ce que c'est qu'une ligne d'un de ces fichiers :

51 19 * * 0 root /usr/local/bin/backupbackup

Cette ligne signifie de lancer le programme « /usr/local/bin/backupbackup » tous les dimanches à 19h51mn.

La syntaxe générale d'un ligne est de cette forme :

mm hh jj MMM JDS commande

mm : codé sur 2 chiffres représente les minutes (de 0 à 59, mais 51mn dans notre exemple)

hh : codé sur 2 chiffres correspond aux heurs (de 0 à 23, mais 19h dans notre exemple)

jj : codé sur 2 chiffres correspond à la date du jour (de 1 à 31, mais * dans notre cas qui signifie tous les jours)

MMM : mois codé sur 2 chiffres ou 3 lettres (de 1 à 12 ou de jan à dec, mais le * dans notre cas signifie tous les mois)

JDS : Jour de la semaine codé par 1 chiffre ou 3 lettres (de 0 à 7 ou de sun à sun. Le 0 dans notre exemple signifie dimanche. 0 ou 7 représente dimanche).

commande : est la commande qui sera exécutée

Un autre exemple :

27 3 * * * list

la commande list est exécuté à 3h (**3**) 27mn (**27**) tous les jours (*****), tous les mois (*****), et tous les jours de la semaine (*****).

3- Quelques détails sur les champs et caractères d'une ligne :

Quand on a un nombre comme champ c'est bon. Par exemple, si on trouve 17 sur la position des jours, c'est qu'il s'agit du 17è jour du mois.

Mais il est possible d'avoir d'autres caractères :

***** : signifie tout ou chauqe (chaque unité de temps). Si * est positionné sur les minutes, il s'agit donc d'exécuter la commande toutes les minutes (mais cela dépendra aussi des heures et jours précisés). Par exemple, je peux exécuter une commande toutes les minutes à 12h le 1er du mois.

, : permet de définir une liste. Exemple : 3,5,17

- : permet de préciser un intervalle. Exemple : de 1 à 10 se note 1-10 ==> 1,2,3,4,5,6,7,8,9

***/** : sert à définir des intervalles avec des pas différents de 1. Par exemple : */10 signifie 0,10,20,30, ... donc toutes les 10 unités de temps. Si c'est position sur les minutes, ça signifie toutes les 10 minutes.

@reboot : permet de lancer la commande au démarrage de la machine

@yearly : tous les ans

@daily : tous les jours

Sources :

http://www.unixgeeks.org/security/newbie/unix/cron-1.html +++++

http://doc.ubuntu-fr.org/cron

Voir aussi **anacron** pour les tâches qui ont forcement besoin d'être exécutées même si la machine est arrêté au moment où la tâche est prévue. Evidemment, ça n'executera rien si la machine est arrêtée, mais ça pourra le faire au redémarrage de celle-ci.