



CRONTAB

- Objet : Lancer une tache automatiquement à l'aide de cron
- Niveau requis :
[avisé](#)
- Commentaires : *Fonctionnement et syntaxe*
- Suivi :
 - Création par  [smolki](#) le 13/05/2010
 - Testé par  [Kao](#) le 06/03/2014
- Commentaires sur le forum : [ici](#) ¹⁾

Le service cron

- Il est chargé de faire exécuter par le système toutes tâches (commandes et scripts) définies et planifiées à l'avance (cron vient de chronos, le dieu du temps !)
- Ces tâches, allant de la simple commande aux scripts complexes, peuvent ainsi être exécutées à heure fixe et même de façon périodique, et fait l'objet de messages de compte-rendu.
- La commande `/usr/bin/crontab` permet cette programmation. Son usage est en principe réservée à root. On peut toutefois autoriser certains utilisateurs. Pour cela on en dresse la liste sur des lignes successives dans le fichier `/etc/crontab.allow`, et de façon symétrique, on peut mettre dans `/etc/crontab.deny` la liste des utilisateurs non autorisés.

Exemple : seuls jean et toto ont la permission d'utiliser le service crontab

```
cat /etc/crontab.allow
```

[/etc/crontab.allow](#)

```
jean  
toto
```

Fonctionnement

Le processus crond est normalement lancé au démarrage.

On peut le lancer ou l'arrêter avec :

```
service cron stop
```

```
service cron start
```

Il lit toutes les minutes les fichiers présents dans le répertoire `/var/spool/cron` ainsi que le fichier `/etc/crontab`, pour voir si des tâches doivent être exécutées.

Chaque action de crond ajoute une ligne de message dans le fichier `/var/log/cron`, qu'il faut

purger de temps en temps ... par une commande crontab !
Par défaut si une commande produit un affichage, il est dirigé vers la messagerie.

Commande crontab

```
[-u user] {-l | -r | -e }
```

Options :

```
crontab -l
```

affiche le fichier crontab de l'utilisateur

```
crontab -r
```

efface ce fichier ; attention aucune demande de confirmation

```
crontab -l -u jean
```

root examine le fichier crontab de l'user jean

```
crontab -e
```

crée ou édite (pour modification) un fichier temporaire dans /tmp ouvert dans votre éditeur en ligne de commande par défaut.

Lors de la sauvegarde, le fichier est écrit dans /var/spool/cron/\$USER, où \$USER est le nom de login de l'utilisateur.

Pour éditer ce fichier avec votre éditeur préféré, par exemple vim, il faut éditer la variable EDITOR, à mettre cette ligne dans votre *shell* tel que bashrc, zshrc etc.

```
export EDITOR=/usr/bin/vim
```

Syntaxe des lignes des tables crontab

Chaque ligne du fichier contient 6 champs.

Les 5 premières déterminent, indiqués par une étoile les moments d'exécution de la tâche décrite au 6ème champ.

les 5 premiers, indiqués par des étoiles (*) séparés par des espaces, sont appelés champs temporels, décrivent la périodicité :

1. minutes (0-59),

2. heures (0-23),
3. jour du mois (1-31),
4. mois de l'année (1-12),
5. jour de la semaine (0-6, 0= dimanche)
6. l'espace finale est libre car destiné soit à la *rédaction* ... de la commande , soit au */chemin/...* du script à exécuter.

Un champ temporel peut contenir :

1. une valeur précise et valide pour le champ (par exemple 15 sur le champ minute)
2. une liste de valeurs valides, séparées par des virgules (1,3,5 dans le champ mois : janvier, mars, mai)
3. un intervalle valide (1-5 dans le champ jour : du lundi au vendredi)
4. * pour signifier toutes les valeurs possibles du champ (* dans le champ minute : toutes les minutes)
5. */5 (dans le champ minutes : tous les 5 minutes), 0-23/3 (dans le champ heures : toutes les 3 heures)

Exemples

Exécution chaque 1er et 15 de chaque mois à minuit :

`crontab`

```
0 0 1,15 * * <commande>
```

Provoquer un reboot la machine chaque 1er et 15 du mois à 2h 30 du matin :

`crontab`

```
30 2 1,15 * * /sbin/shutdown -r
```

Appeler un script de sauvegarde tous les lundis a 3 h 15 du matin :

`crontab`

```
15 3 * * 1 /usr/bin/backup
```

Appeler un script de sauvegarde tous les mardis à 3 h du matin :

`crontab`

```
0 3 * * 2 /usr/bin/backup
```

Si on laisse l'étoile des minutes, : * 3 * * 1 le script va se lancer toutes les minutes de l'heure 3...

Merci à l'amicale vigilance du **captfnfab**, le Zorro des redresseur d'erreurs du wiki qui m'a indiqué cela ! 😊

Exécution toutes les 15 minutes de la <commande> :

[crontab](#)

```
*/15 * * * * <commande>
```

Exécution toutes les heures passées 15 minutes :

[crontab](#)

```
15 * * * * <commande>
```

Exécution tous les matins du lundi au vendredi à 7 h 30 :

[crontab](#)

```
30 7 * * 1-5 <commande>
```

Exécution tous les quarts d'heure de 15h à 19h du lundi au vendredi seulement en 1ère quinzaine du troisième trimestre :

[crontab](#)

```
0,15,30,45 15-19 1-15 7-9 1-5 <commande>
```

Trouver puis nettoyer le répertoire /tmp des vieux fichiers (non modifiés depuis 31 jours) tous les 1er jour de chaque mois, à 2 heures du matin :

[crontab](#)

```
0 2 1 * * find /tmp -atime 31 -exec rm -f {} \;
```

Faire apparaitre un message dans la GUI d'un utilisateur à une heure précise

[crontab](#)

```
50 18 * * * export DISPLAY=localhost:0.0; xhost + | zenity --info --  
text="Reboot de l'ordinateur dans 10min, penser à faire vos  
sauvegardes"
```

Liens vers le forum df

- [Crontab - lancer une tache une seule fois](#)

Liens:

- Article Wikipedia avec de nombreux exemples (fr) <http://fr.wikipedia.org/wiki/Crontab>

1)

N'hésitez pas à y faire part de vos remarques, succès, améliorations ou échecs !

From:

<http://debian-facile.org/> - **Documentation - Wiki**

Permanent link:

<http://debian-facile.org/doc/systeme:crontab>

Last update: **13/10/2018 10:02**

