

LINUX: Exécuter une commande régulièrement avec crontab

Table of Contents

- [La « crontab », qu'est-ce que c'est ?](#)
- [Comment utilise-t-on crontab ?](#)
- [Pour exécuter une commande tous les jours à une heure précise](#)
 - [Les différentes notations possibles](#)
- [Autres exemples de programmation](#)

La « crontab » constitue un incontournable sous Linux : cet outil nous permet de programmer l'exécution régulière d'un programme.

Contrairement à `at` qui n'exécutera le programme qu'une seule fois, `crontab` permet de faire en sorte que l'exécution soit répétée : toutes les heures, toutes les minutes, tous les jours, tous les trois jours, etc. Un peu de configuration...

La « crontab », qu'est-ce que c'est ?

`crontab` est en fait une commande qui permet de lire et de modifier un fichier appelé la « crontab ». Ce fichier contient la liste des programmes que vous souhaitez exécuter régulièrement, et à quelle heure vous souhaitez qu'ils soient exécutés.

`crontab` permet donc de changer la liste des programmes régulièrement exécutés. C'est toutefois le programme `cron` qui se charge d'exécuter ces programmes aux heures demandées.

Ne confondez donc pas `crontab` et `cron` : le premier permet de modifier la liste des programmes à exécuter, le second les exécute.

Comment utilise-t-on crontab ?

Il y a trois paramètres différents à connaître, pas plus :

```
-e : modifier la crontab ;  
-l : afficher la crontab actuelle ;  
-r : supprimer votre crontab. Attention, la suppression est immédiate et  
sans confirmation !
```

Commençons par afficher la crontab actuelle :

```
$ crontab -l
```

```
no crontab for <nom_user>
```

Il y a une crontab par utilisateur et root a aussi sa propre crontab :

```
$ sudo crontab -l  
no crontab for root
```

Pour modifier de la crontab. Taper :

```
$ crontab -e
```

Si la crontab est vide, le fichier ne contient qu'une seule ligne :

```
# m h dom mon dow  command
```

Comme cette ligne est précédée d'un #, il s'agit d'un commentaire (qui sera donc ignoré). Cette ligne vous donne quelques indications sur la syntaxe du fichier :

- m : minutes (0 - 59) ;
- h : heures (0 - 23) ;
- dom (day of month) : jour du mois (1 - 31) ;
- mon (month) : mois (1 - 12) ;
- dow (day of week) : jour de la semaine (0 - 6, 0 étant le dimanche) ;
- command : c'est la commande à exécuter.

Il faut d'abord indiquer à quel moment la commande doit être exécutée, puis ensuite écrire à la fin la commande à exécuter.

C'est un peu comme un tableau. Chaque champ est séparé par un espace.

Chaque champ numérique peut être remplacé par une étoile qui signifie « tous les nombres sont valables ».

Bien comprendre la crontab n'est pas si simple, pour cela voici quelques exemples pour voir comment ça fonctionne.

Pour exécuter une commande tous les jours à une heure précise

```
47 15 * * * touch /home/fichier.txt
```

Seules les deux premières valeurs sont précisées : les minutes et les heures. Chaque fois qu'il est 15 h 47, la commande indiquée à la fin sera exécutée.

Il est préférable d'utiliser le chemin du fichier en entier, car on ne peut pas être sûrs que le cron s'exécutera dans le bon répertoire .

Désormais, fichier.txt sera créé dans le répertoire /home tous les jours à 15 h 47 (s'il n'existe pas déjà).

Les différentes notations possibles

Pour chaque champ, on a le droit à différentes notations :

- 5 (un nombre) : exécuté lorsque le champ prend la valeur 5 ;
- * : exécuté tout le temps (toutes les valeurs sont bonnes) ;
- 3,5,10 : exécuté lorsque le champ prend la valeur 3, 5 ou 10. Ne pas mettre d'espace après la virgule ;
- 3-7 : exécuté pour les valeurs 3 à 7 ;
- */3 : exécuté tous les multiples de 3 (par exemple à 0 h, 3 h, 6 h, 9 h...).

Autres exemples de programmation

Crontab	Signification
47 * * * * commande	Toutes les heures à 47 minutes exactement. Donc à 00 h 47, 01 h 47, 02 h 47, etc.
0 0 * * 1 commande	Tous les lundis à minuit (dans la nuit de dimanche à lundi).
0 4 1 * * commande	Tous les premiers du mois à 4 h du matin.
0 4 * 12 * commande	Tous les jours du mois de décembre à 4 h du matin.
0 * 4 12 * commande	Toutes les heures les 4 décembre.
* * * * * commande	Toutes les minutes, il n'est pas possible d'exécuter une commande plus fréquemment que toutes les minutes
30 5 1-15 * * commande	À 5 h 30 du matin du 1er au 15 de chaque mois.
0 0 * * 1,3,4 commande	À minuit le lundi, le mercredi et le jeudi.
0 */2 * * * commande	Toutes les 2 heures (00 h 00, 02 h 00, 04 h 00...)
*/10 * * * 1-5 commande	Toutes les 10 minutes du lundi au vendredi.

From:

<http://www.ouarte.garden/> - dwndoc

Permanent link:

<http://www.ouarte.garden/doku.php?id=debian:linux-programmer-commande-cron>

Last update: 2025/02/19 10:59

