

 eBook Gratuit

APPRENEZ

crontab

eBook gratuit non affilié créé à partir des
contributeurs de Stack Overflow.

#crontab

Table des matières

À propos	1
Chapitre 1: Démarrer avec crontab	2
Remarques.....	2
Exemples.....	2
Installer crontab sur Linux.....	2
Créer un nouveau cron sur Ubuntu (ou la plupart des autres systèmes Linux).....	2
Crédits	5

À propos

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [crontab](#)

It is an unofficial and free crontab ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official crontab.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapitre 1: Démarrer avec crontab

Remarques

Crontab signifie table cron. C'est un utilitaire utilisé pour ajouter, éditer et supprimer des crons. Un cron est une commande ou un programme à exécuter régulièrement. Par exemple: un script de sauvegarde qui s'exécute tous les jours ou toutes les heures.

Exemples

Installer crontab sur Linux

Debian / Ubuntu

```
# apt-get update & apt-get -y upgrade
# apt-get install cron
```

Fedora / CentOS

```
# yum -y update
# yum install vixie-cron
```

Cambre

```
# pacman --noconfirm -Syu
# pacman -S cronie
```

Créer un nouveau cron sur Ubuntu (ou la plupart des autres systèmes Linux)

Vous pouvez créer une nouvelle entrée cron en tapant simplement

```
crontab -e
```

sur la ligne de commande. Si vous souhaitez modifier votre fichier crontab pour la première fois, vous serez invité à choisir un éditeur:

```
no crontab for <user> - using an empty one

Select an editor. To change later, run 'select-editor'.
 1. /bin/ed
 2. /bin/nano      <---- easiest
 3. /usr/bin/vim.basic
 4. /usr/bin/vim.tiny

Choose 1-4 [2]:
```

Choisissez simplement votre éditeur en suivant l'invite, et votre crontab s'ouvrira avec un fichier

vide (contenant uniquement des lignes d'explication commentées):

```
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow  command
```

Comme vous pouvez le voir, il existe déjà un exemple d'entrée dans le texte:

```
0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
```

Cela créerait un fichier de sauvegarde appelé `home.tgz` dans `/var/backups/`. Le timing pour cette crontab serait

```
every monday (first day of week) at 5:00 A.M.
```

Si vous aviez entré cette ligne en tant que crontab, il ne restait plus qu'à enregistrer le fichier crontab. Par exemple, avec l'éditeur `nano`, cela se fait avec `<Ctrl> + <X>` - puis confirmez pour enregistrer avec `Y`

Pour vérifier votre crontab, tapez simplement

```
crontab -l
```

dans la console.

Plus d'informations sur les crontimings que vous pouvez choisir:

```
# * * * * *  command to execute
# | | | | |
# | | | | |
# | | | | | _____ day of week (0 - 6) (0 to 6 are Sunday to Saturday, or use names; 7 is
Sunday, the same as 0)
# | | | | | _____ month (1 - 12)
# | | | | | _____ day of month (1 - 31)
# | | | | | _____ hour (0 - 23)
```

Les caractères spéciaux dans les cronjobs sont:

Astérisque (*)

L'astérisque indique que l'expression cron correspond à toutes les valeurs du champ. Par exemple, utiliser un astérisque dans le 4ème champ (mois) indique chaque mois.

Slash (/)

Les barres obliques décrivent les incréments de plages. Par exemple, 3-59 / 15 dans le premier champ (minutes) indique la troisième minute de l'heure et toutes les 15 minutes par la suite. La forme "*" / ..." est équivalente à la forme "premier-dernier / ...", c'est-à-dire un incrément sur la plus grande plage possible du champ.

Virgule (,)

Les virgules sont utilisées pour séparer les éléments d'une liste. Par exemple, utiliser "MON, WED, FRI" dans le 5ème champ (jour de la semaine) signifie les lundis, mercredis et vendredis.

Trait d'union (-)

Les traits d'union définissent des plages. Par exemple, 2000-2010 indique chaque année entre 2000 et 2010, inclusivement.

Pourcentage (%)

Les signes de pourcentage (%) de la commande, à moins qu'ils ne soient échappés avec une barre oblique inverse (), sont transformés en caractères de nouvelle ligne et toutes les données après le premier% sont envoyées à la commande en tant qu'entrée standard.

Lire Démarrer avec crontab en ligne: <https://riptutorial.com/fr/crontab/topic/5750/demarrer-avec-crontab>

Crédits

S. No	Chapitres	Contributeurs
1	Démarrer avec crontab	baao , Community , Harikrishnan , jesussegado , Tobias F.