



EBook Gratis

APRENDIZAJE crontab

Free unaffiliated eBook created from
Stack Overflow contributors.

#crontab

Tabla de contenido

Acerca de.....	1
Capítulo 1: Empezando con crontab.....	2
Observaciones.....	2
Examples.....	2
Instalar crontab en Linux.....	2
Creando un nuevo cron en ubuntu (o en la mayoría de los otros sistemas operativos de Linux.....	2
Creditos.....	5

Acerca de

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

Capítulo 1: Empezando con crontab

Observaciones

Crontab significa cron table. Es una utilidad utilizada para agregar, editar y borrar crons. Un cron es un comando o programa que debe ejecutarse en un horario regular. Ej .: Un script de copia de seguridad que se ejecuta todos los días o cada hora.

Examples

Instalar crontab en Linux

Debian / Ubuntu

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

Fedora / CentOS

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

Arco

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

Creando un nuevo cron en ubuntu (o en la mayoría de los otros sistemas operativos de Linux)

Puedes crear una nueva entrada cron simplemente escribiendo

```
crontab -e
```

en la línea de comando Si es la primera vez que desea editar su crontab (le), se le solicitará una selección de editor:

```
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]:
```

Simplemente elija su editor siguiendo el indicador, y su crontab se abrirá con un archivo vacío (que solo contiene algunas líneas de explicación comentadas):

```
# 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
```

Como puede ver, ya hay una entrada de ejemplo en el texto:

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

Esto crearía un archivo de copia de seguridad llamado `home.tgz` dentro de `/var/backups/`. El momento para este crontab sería

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

Si ingresó esa línea como su archivo crontab, todo lo que tenía que hacer ahora sería guardar el archivo crontab. Por ejemplo, con el editor `nano`, esto se hace con `<Ctrl> + <X>`, luego confirme para guardar con `Y`

Para comprobar su crontab, simplemente escriba

```
crontab -l
```

en la consola

Un poco más de información sobre crontimings puede elegir:

```
# * * * * *  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)
# | _____ min (0 - 59)
```

Los caracteres especiales en cronjobs son:

Asterisco (*)

El asterisco indica que la expresión cron coincide con todos los valores del campo. Por ejemplo, usar un asterisco en el 4to campo (mes) indica cada mes.

Barra (/)

Las barras inclinadas describen incrementos de rangos. Por ejemplo, 3-59 / 15 en el primer campo (minutos) indica el tercer minuto de la hora y cada 15 minutos a partir de entonces. La forma "* / ..." es equivalente a la forma "primera-última / ...", es decir, un incremento sobre el mayor rango posible del campo.

Coma (,)

Las comas se utilizan para separar elementos de una lista. Por ejemplo, usar "MON, WED, FRI" en el quinto campo (día de la semana) significa lunes, miércoles y viernes.

Guión (-)

Los guiones definen rangos. Por ejemplo, 2000-2010 indica todos los años entre 2000 y 2010 AD, inclusive.

Porcentaje (%)

Los signos de porcentaje (%) en el comando, a menos que se escapen con barra diagonal inversa (), se cambian a caracteres de nueva línea, y todos los datos después del primer% se envían al comando como entrada estándar.

Lea Empezando con crontab en línea: <https://riptutorial.com/es/crontab/topic/5750/empezando-con-crontab>

Creditos

S. No	Capítulos	Contributors
1	Empezando con crontab	baao , Community , Harikrishnan , jesussegado , Tobias F.