

Curso de administración de sistemas GNU/linux  
**CRON: El planificador de tareas**

Por Jorge Fuertes  
<http://jorgefuertes.com>

©2009 Jorge Fuertes Alfranca  
Revisada al 15 de mayo de 2009

# Índice

<b>1. Introducción</b>	<b>3</b>
<b>2. El crontab de root</b>	<b>3</b>
2.1. Listando el crontab . . . . .	4
2.2. Programando tareas . . . . .	4
<b>3. Directorios predefinidos: hourly, daily, weekly y monthly</b>	<b>5</b>
<b>4. Crontab del sistema</b>	<b>5</b>
<b>5. Log de cron</b>	<b>6</b>
<b>6. Ejercicios sobre cron</b>	<b>7</b>
<b>7. Sobre esta unidad didáctica</b>	<b>9</b>
7.1. Notas y advertencias . . . . .	9
7.2. Derechos . . . . .	9
7.3. Agradecimientos . . . . .	9
7.4. Revisiones . . . . .	9

## 1. Introducción

**Cron**<sup>1</sup> es un administrador o planificador de tareas de segundo plano. Es un *demonio* que ejecuta procesos, programas o scripts, a intervalos regulares programados. Por ejemplo podemos instruirlo para ejecutar un script cada cinco minutos, o una copia de seguridad diaria, o una exportación semanal de la base de datos.

Las horas a las que deben ejecutarse dichos procesos se indican en el fichero *crontab* correspondiente al usuario.

En el sistema operativo Unix, *cron* es un administrador regular de procesos en segundo plano (*demonio*) que ejecuta procesos o scripts a intervalos regulares (por ejemplo, cada minuto, día, semana o mes). Los procesos que deben ejecutarse y la hora en la que deben hacerlo se especifican en el archivo *crontab* correspondiente.

En este módulo veremos como:

- Añadir tareas al *crontab* de root.
- Añadir tareas enlazadas cada hora, diarias, semanales y mensuales.
- Añadir tareas al *crontab* del sistema.

La guía está escrita basándose en la distribución *Debian 5.0/Lenny de GNU/Linux* aunque todo lo relacionado con *cron* debería funcionar sin cambios en cualquier distro. No se pretende que esto sea un manual exhaustivo de *cron*, al contrario es una iniciación que nos permitirá empezar a manejarnos sin problemas con este planificador.

**NOTA:** En esta guía aparecerá en repetidas ocasiones el conjunto de símbolos "#>". Son los símbolos que indican que se debe de estar en la línea de comandos de su consola, el *prompt* de su sistema, que puede ser diferente. No debe teclear esos símbolos, son sólo una indicación.

## 2. El crontab de root

El fichero *crontab* de root se guarda en */var/spool/cron/crontabs/root*, pero nunca editaremos los ficheros de *cron* directamente. Para editar el *crontab* (fichero donde se guardan las tareas programadas) de root, utilizaremos siempre la utilidad *crontab*. Concretamente utilizaremos la orden ***crontab -e***:

```
#> crontab -e
```

Utilizaremos siempre esta orden en lugar de ir a buscar el fichero con un editor de texto ya que así nos garantizamos que estamos editando un fichero auxiliar en exclusiva, y evitaremos problemas de concurrencia con otros administradores.

Probablemente *crontab* nos presentará un fichero vacío o bien con la siguiente línea de cabeceras:

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

---

<sup>1</sup>[http://es.wikipedia.org/wiki/Cron\\_\(unix\)](http://es.wikipedia.org/wiki/Cron_(unix))

Esta línea simplemente nos servirá de guión para escribir debajo de ella nuestras tareas. Como se puede ver nos está indicando el formato, primero los minutos, luego la hora, el día del mes, el mes y el día de la semana, es decir:

```
# minutos horas día_del_mes mes_1-12 día_semana_1-7 comando_a_ejecutar
```

No obstante añadiremos unas líneas al principio, dándole algunas variables de entorno de nuestro interés, por ejemplo:

```
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=sistemas@conectahosting.es
```

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

En la primera variable, "*SHELL*" indicaremos con qué intérprete deberemos ejecutar los comandos, con "*PATH*" estableceremos el camino de búsqueda de comandos que deberá seguir el sistema, y con la variable "*MAILTO*" definiremos a que buzón irán los avisos en caso de algo vaya mal al ejecutar una de las tareas. Estas variables no son estrictamente necesarias, sólo son para ajustarlas si queremos.

## 2.1. Listando el crontab

Para obtener un listado de las tareas programadas de *root*, ejecutaremos:

```
#> crontab -l

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=sistemas@conectahosting.es

# m h dom mon dow command
```

Evidentemente no saldrán tareas si todavía no hemos programado ninguna.

## 2.2. Programando tareas

Para programar una tarea utilizaremos la orden *crontab -e*. Por ejemplo para programar una tarea que a las 19h genere un mensaje en un log, escribiremos lo siguiente:

```
#> crontab -e

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root

# m h dom mon dow command
0 19 * * * echo "Son las 19h." >> /var/log/mi-cron.log
```

Si quisiéramos que se ejecutase a las 19:30h pero sólo los lunes pondríamos:

```
# m h dom mon dow command
30 19 * * 1 echo "Son las 19:30h. del lunes." >> /var/log/mi-cron.log
```

Y si quisiéramos que se ejecute a las 19:30 del día 1 de diciembre:

```
# m h dom mon dow command
30 19 1 12 * echo "Son las 19:30h. del 1 de diciembre." >> /var/log/mi-cron.log
```

¿Pero que ocurre si queremos que se ejecute a las 19 y a las 19:30h? Podemos poner más de un valor en cada campo, separándolos por comas:

```
# m h dom mon dow command
0,30 19 * * * echo "Son las 19 o las 19:30h. No estoy seguro." >> /var/log/mi-cron.log
```

¿Y para ejecutarlo cada 5 minutos? Podríamos poner "5,10,15,20,25,..." pero hay una abreviatura, así "*\*/min*":

```
# m h dom mon dow command
*/5 * * * * echo "Han pasado cinco minutos." >> /var/log/mi-cron.log
```

### 3. Directorios predefinidos: hourly, daily, weekly y monthly

Los sistemas UNIX modernos vienen con unos directorios predefinidos para que cron los lee y ejecute lo que hay dentro en los intervalos que su nombre indica:

- /etc/cron.daily
- /etc/cron.hourly
- /etc/cron.weekly
- /etc/cron.monthly

Dichos directorios se suelen utilizar para enlazar guiones que deban ser llamados en el intervalo correspondiente al directorio, sin argumentos. Por ejemplo, si creamos un guión de *bash* y lo guardamos en */root/bin*, le damos permisos de ejecución, y lo enlazamos en */etc/cron.hourly*, cron lo ejecutará cada hora:

```
#> chmod +x /root/bin/mi-script.sh
#> ln -s /root/bin/mi-script.sh /etc/cron.hourly/.
```

Fácil, ¿No?

### 4. Crontab del sistema

El archivo de cron de sistema es */etc/crontab*. Si ponemos tareas en dicho archivo serán ejecutadas igualmente, pero no es recomendable. Este crontab se deja para que lo maneje la distribución y sus programas.

Es igual que el crontab de root, salvo que en este podemos especificar con qué usuario se ejecuta cada cosa, y cron hará una suplantación previa a la ejecución.

Un */etc/crontab* típico es:

```
#> cat /etc/crontab

# etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the 'crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 * * * * root    cd / && run-parts --report /etc/cron.hourly
25 6 * * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6 * * 7 root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6 1 * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
```

Es desde este archivo desde el que nuestro cron ejecutará las tareas enlazadas descritas en la sección 3.

## 5. Log de cron

En *Debian 5 Lenny* el log de cron viene desactivado por defecto. Para activarlo tendremos que editar el fichero `/etc/default/cron` y eliminar la almohadilla (`#`) del principio de la línea `EXTRA_OPTS="-L 2"` dejando el fichero así:

```
# cat /etc/default/cron
# Cron configuration options

# Uncomment this option for LSB name support in /etc/cron.d/
#LSBNames='-l'

# Whether to read the system's default environment files (if present)
# This will make cron set a proper charset for the mails it sends
# Comment this or set it to something other than 'yes' to prevent
# cron from reading it.
READ_ENV="yes"

# Extra options for cron, see cron(8)
# For example, set a higher log level to audit cron's work
EXTRA_OPTS="-L 2"
```

De esta forma conseguiremos que cron vaya dejando un rastro en `/var/log/syslog` que podremos observar, por ejemplo, así:

```
tail -f /var/log/syslog
```

Si quisiéramos separar el log de cron en otro fichero, como `/var/log/cron.log`, tendremos que configurar el demonio de log para ello. Como eso excede del ámbito de este módulo simplemente diremos que se puede hacer y que el autor lo ha probado con *"syslog-ng"* que es un daemon de log avanzado.

## 6. Ejercicios sobre cron

1. Programe una tarea que cada 5 minutos escriba la fecha y hora del sistema en un fichero */var/log/mi-cron.log*.
2. Programe una tarea para que a las 20h. escriba " *Descanso de 5 minutos.*" en el terminal.
3. Programe una tarea que cada hora haga un tar, comprimido con gzip, de su */etc* y que lo guarde en */backup*.
4. Haga una tarea que borre su */var/log/mi-cron.log* cada día.
5. Haga una tarea a una hora determinada, la que usted quiera, que capture las últimas líneas del log del sistema y se las deje en */root/ultimolog.txt*.
6. Cree un script que haga un tar de su */root*, que lo guarde en */backup*, el nombre de dicho tar estará formado por la fecha y la hora en formato "ddmmaaaahhmmss", un guión y la palabra root-bk, por ejemplo:

`23032009194520-root-bk.tar.gz`

Deben guardarse en el directorio */backup*. Programe cron para que ejecute este script cada media hora.

7. Programe un script que borre los backups, correspondientes al script del ejercicio anterior, que sean más viejos de 24h. Este script de limpieza debe ejecutarse cada día.
  
8. Localize el log de cron, examínelo y anote aquí donde está.
  
9. Defina una tarea que apague el equipo todos los días a las 23:59h.
  
10. Defina una tarea que encienda el equipo todos los días a las 9h. (;-D).
  
11. **Medio:** Haga lo mismo que en el ejercicio 9, pero comprobando antes si hay usuarios trabajando en la máquina y, si es así, deberá avisarles de que va a apagar y darles 15 minutos de margen para salir antes de apagar de verdad.
12. **Medio:** Haga un script que mire la ocupación de disco de cada partición montada en el sistema, y que deje un informe de dicha ocupación en */var/log/disco-ocupado.log*.
13. **Avanzado:** Haga un script que mire la ocupación de disco de cada partición montada en el sistema, cada 10 minutos, y que le envíe un email a *root* en caso de que alguna de ellas esté a más del 90 % de ocupación. El email debe incluir la fecha y la hora, el nombre de la máquina, el punto de montaje de cada partición y el porcentaje de ocupación.

## 7. Sobre esta unidad didáctica

### 7.1. Notas y advertencias

**Debian:** Esta guía está basada en el sistema *Debian GNU/Linux*, podría haber pequeños cambios si se aplica a otras distribuciones de *GNU*, pero en su mayor parte funcionará bien con la excepción de lo referido al sistema de paquetería de programas, los comandos que empiezan por *apt*, ya que otras *distros* no basadas en *Debian* podrían incorporar sistemas diferentes para el manejo de sus paquetes.

### 7.2. Derechos

Esta guía se cede bajo contrato Coloriuris. Sólo puede ser utilizada previa aceptación del contrato de cesión sito en:

- <http://www.coloriuris.net/contratos/ef5af6aaa441ab9c213273fade56dca1>

Dicho contrato garantiza que estoy cediendo los derechos de uso y modificación sin ánimo de lucro.

### 7.3. Agradecimientos

El autor quiere reflejar su agradecimiento a todas las páginas de Internet que ponen a disposición de todo el mundo sus contenidos, así como a todo aquél que publica artículos, manuales y experiencias en Internet, ya que eso favorece a la difusión del conocimiento y al desarrollo humano. *La información quiere ser libre.*

Un agradecimiento muy especial a toda la comunidad del Software Libre. Sin ellos el autor viviría en la oscuridad: Programadores, traductores, asociaciones, hacktivistas, webmasters, etc...

También quiero agradecer muy especialmente su ayuda a mis alumnos y lectores, por tomarse la molestia de comunicarme las erratas y por darme ideas para mejorar los ejercicios.

### 7.4. Revisiones

El autor irá eventualmente publicando revisiones de esta unidad en su página personal, y estará encantado de recibir sugerencias y dudas en la misma o en su email:

- <http://jorgefuertes.com>.
- [cursos@jorgefuertes.com](mailto:cursos@jorgefuertes.com).

Por supuesto se puede contactar con el autor para contratarle para hacer nuevas unidades, adaptaciones, modificaciones, cursos, etc...