

# cron

## Conceptos básicos

comprobar si está en ejecución:

```
$ ps -ef | grep crond
```

Añadir el servicio si no lo estuviera:

```
$ chkconfig --level 35 crond on
```

## variables de entorno

```
$ export EDITOR=nano
```

## Uso sencillo

añadir el script a ejecutar en alguno de estos directorios (o un link):

- /etc/cron.hourly → se ejecuta cada hora
- /etc/cron.daily → se ejecuta cada día
- /etc/cron.weekly → se ejecuta cada semana
- /etc/cron.monthly → se ejecuta cada mes

## Uso "elaborado"

modificar /etc/crontab las líneas que realmente ejecutan los «trabajos» son del estilo:

```
* * * * * <usuario> <comando> [ > /dev/null 2>&1 ]
```

Donde los primeros asteriscos corresponden, de izquierda a derecha:

- minuto
- hora
- día del mes
- mes
- día de la semana

Un asterisco indica «cualquiera»,

- Se puede indicar rangos: 1-3
- Se pueden indicar listas: 1,3,5,7
- Se pueden indicar fracciones: \*/15 ⇒ cada 15 minutos, 1-10/2 ⇒ los 1,3,5,7,9 (empieza en el 1, a saltos de 2), 2/4 ⇒ 2,6,10,14

```
----- minutos (0 - 59)
| ----- horas (0 - 23)
| | ----- día del mes (1 - 31)
```

```

| | | - - - - - mes (1 - 12)
| | | | - - - - - dia de la semana (0 - 6) (domingo=0, lunes=1, ... sabado=6)
| | | | |
* * * * * comando a ejecutar

* significa todos los valores validos
/ permite definir una repeticion
- permite definir un rango
, permite definir varios valores

```

otros formatos (special strings):

- @reboot → Run once, at startup.
- @yearly → Run once a year, «0 0 1 1 \*».
- @annually → (same as @yearly)
- @monthly → Run once a month, «0 0 1 \* \*».
- @weekly → Run once a week, «0 0 \* \* 0».
- @daily → Run once a day, «0 0 \* \* \*».
- @midnight → (same as @daily)
- @hourly → Run once an hour, «0 \* \* \* \*».

## personalizado

con los siguientes comandos podemos gestionar nuestros propios trabajos CRON (individual para cada usuario):

```

$ crontab <fichero> : añade el fichero con formato cron
$ crontab -e : edita el fichero
$ crontab -l : lista los trabajos añadidos a nuestros fichero
$ crontab -r : elimina los crontab

```

No sirve modificar el fichero original, hay que añadirlo cuando toque.

El formato del crontab es ligeramente diferente, no se ha de especificar el usuario.

permite una linea del tipo MAILTO=«direccion@correoelectronico.es»

El fichero se guarda, a buen recaudo, en /var/spool/cron/crontabs (en una debian, al menos)

## ejemplos

- ejecuta el trabajo a las 00:00, 06:00, 12:00, 18:00

```
0 0,6,12,18 * * * /comando
```

- ejecuta el trabajo a las 00:00, 06:00, 12:00, 18:00 pero solo los días laborables:

```
0 0,6,12,18 * * 1-5 /comando
```

- ejecuta el trabajo a las 00:00, 06:00, de 9:00 a 15:00 cada hora, 18:00 pero solo los días laborables:

```
0 0,6,9-15,18 * * 1-5 /comando
```

- ejecuta el trabajo cada 6 horas (basicamente, cuando la división da 0 de resto) a la hora en punto (el 0 de

los minutos):

```
0 */6 * * * /comando
```

- ejecuta el trabajo cada 5 minutos:

```
*/5 * * * * /comando
```

- ejecuta el trabajo todos los días pares a las 12:00:

```
0 12 */2 * * /comando
```

## seguridad

se puede controlar quien usa y quien no el servicio CRON en los archivos:

```
/etc/cron.allow  
/etc/cron.deny
```

en estos ficheros se añade la lista de usuarios con o sin permisos. Se puede utilizar ALL en cualquiera de los 2

Si no se crea el cron.deny y si el cron.allow, es como si hubiesemos creado un cron.deny con un ALL y tendremos que añadir a todos los usuarios que tengan que utilizar este servicio en el cron.allow

## más info

[Cron on Linux](#)

<https://help.ubuntu.com/community/CronHowto>

From:  
<https://miguelangel.torresegea.es/wiki/> - **miguel angel torres egea**

Permanent link:  
<https://miguelangel.torresegea.es/wiki/linux:cron:cron?rev=1543401415>

Last update: **28/11/2018 02:36**

