



EL PLANIFICADOR DE TAREAS DE LINUX

Herramienta "CRON y CRONTAB"

Manual creado por Isabel M^a Cano Jordán y Juan Gabriel Merlos Casanova

INDICE:

- 1. ¿Qué es Cron?**
- 2. ¿Cómo funciona Cron?**
- 3. ¿Qué es Crontab?**
- 4. Arquitectura de Crontab**
- 5. ¿Cómo se usa Crontab?**
- 6. Enlaces de interés**

1. ¿Qué es Cron?

Cron es un equivalente del Programador de Tareas de Windows. Se trata de una herramienta administrativa de procesos en segundo plano de Linux/Unix, que ejecuta procesos en intervalos programados. Los procesos que deben ejecutarse y la hora en la que deben de hacerlo se encuentran registrados en el fichero Crontab.



CRONTAB

2. ¿Cómo funciona Cron?

Cron se inicia desde el directorio `"/etc/rc.d/"` o `"/etc/init.d"`, dependiendo de la distribución. Este demonio lee periódicamente los procesos guardados en el fichero Crontab y ejecuta los comandos guardados en el horario que fueron programados.

3. ¿Qué es Crontab?

Crontab es un fichero de texto que guarda una lista de comandos a ejecutar en el tiempo especificado por el usuario. Crontab verificará la fecha y hora a la que se debe ejecutar el comando y los permisos de ejecución. Esta es la manera más sencilla de administrar tareas de cron en sistemas multiusuario ya sea como un usuario normal o como root.

4. Arquitectura de Crontab.

La interfaz de Crontab viene distribuida en seis columnas las cuales no siempre se muestran en el Crontab del usuario.

m h dom mon dow command

Dónde:

- “**m**” Son los minutos desde el 0 hasta el 59.
- “**h**” Son las horas desde 0 hasta 23.
- “**dom**” Es el día del mes desde 1 hasta 31.
- “**mon**” Es el mes desde el 1 hasta el 12.
- “**dow**” Es el día de la semana (0= domingo, 1=lunes, 2=martes, 3=miércoles, 4=jueves, 5=viernes, 6=sábado).
- “**command**” Es la ruta absoluta del script a ejecutar o el comando.



5. ¿Cómo se usa Crontab?

En primer lugar editamos el fichero Crontab con el comando “**Crontab -e**”. La primera vez que lo ejecutemos nos dará antes de abrir el fichero una lista con 4 editores para elegir. El editor más recomendable es el nano.

A continuación, nos saldrá la siguiente pantalla:

```
GNU nano 2.2.6 Archivo: /tmp/crontab.kNdIEq/crontab
# 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
[ 22 líneas leídas ]
^G Ver ayuda ^O Guardar ^R Leer Fich ^Y RePág. ^K Cortar Tex ^C Pos actual
^X Salir ^J Justificar ^W Buscar ^U Pág. Sig. ^U PegarTxt ^T Ortografía
```

Después de todos los comentarios podemos añadir los ficheros para que se ejecuten a la hora en la que estén programados

```
10 21 * * 5 /home/isabel/h_salir.sh
```

Fichero que se ejecutara todos los días a las 21:10

```
10 21 * * * /home/isabel/h_salir.sh
```

Fichero que se ejecutará el 20 de noviembre a las 7:30

```
30 7 20 11 * /home/isabel/h_salir.sh
```

Fichero que se ejecutará el domingo 11 de noviembre a las 9:00

```
0 9 11 11 0 /home/isabel/h_salir.sh
```

Fichero que se ejecutará todos los días menos los domingos a las 17:30

```
30 17 * * 1,2,3,4,5_ /home/isabel/h_salir.sh
```

Fichero que se ejecutará todos los meses los días 1, 15 y 18 a las 12:00

```
0 12 1,15,28 * * /home/isabel/h_salir.sh_
```

También Crontab maneja otras cadenas especiales para definir algunos rangos.

@reboot Ejecuta una vez al inicio

@yearly Ejecuta solo una vez al año (0 0 1 1 *)

@annually Es igual que **@yearly**

@monthly Ejecuta una vez al mes, el día primero (0 0 1 **)

@weekly Semanal el primer minuto de la primera hora de la semana (0 0 * * * 1)

@daily Se ejecuta diariamente, por defecto a las 00:00 (0 0 * * *)

@midnight Igual que **@daily**

@hourly al primer minuto de cada hora (0 * * * *)

También en Crontab podemos encontrar otros comandos para manipular ficheros.

<code>crontab archivo</code>	Remplaza el existente archivo de Crontab con un archivo definido por el usuario
<code>crontab -e</code>	Edita el archivo de Crontab.
<code>crontab -l</code>	Lista todas las tareas de Crontab del usuario
<code>crontab -d</code>	Borra el fichero Crontab del usuario
<code>crontab -c dir</code>	Define el directorio de Crontab del usuario

6. Enlaces de interés

Puede consultar más información sobre el uso de la herramienta Cron en las siguientes páginas:

- http://www.linuxtotal.com.mx/?cont=info_admon_006
- <http://blog.desdelinux.net/cron-crontab-explicados/>