Quieres programar una tarea para que se haga ciertos días a ciertas horas, todos los dias...?? este simple comando de consola puede hacer todas aquellas tareas que le ordenemos :)

fuente: http://www.elrincondelprogramador.com/ Autor: Daniel Fernández Garrido

crontab: Programar tareas para que se ejecuten solas cuando queramos (Linux) *Daniel Fernández Garrido*

Existe en Linux una utilidad que no muchos conocen y que resulta a veces imprescindible: **cro ntab**
.

crontab permite programar lo que se llaman *crones*, esto es, tareas que se ejecutarán en un momento determinado del tiempo de manera automática. El sistema Linux (y cualquier UNIX en general) comprueba regularmente, guiándose por el
evento de reloj
del sistema, si existe alguna tarea programada para ejecutarse y, en caso afirmativo, la ejecuta sin necesidad de que nadie (ningún usuario) lo haga explícitamente.

En primera instancia, algunas aplicaciones que se nos pueden ocurrir para algo así podrían ser:

- *Apagar un equipo a la hora que nosotros queramos*: Por ejemplo, mientras dormimos podremos dejarlo bajando cosas de internet haciendo que se apague solo por la mañana.
- *Crear backups*: Podríamos programar un *cron* para que, a cierta hora de ciertos días de la semana o del mes, realizase un backup de nuestros datos de manera automática, sin tener que recordar nosotros el hacerlo.
- *Poner mensajes recordatorios*: Para que nos salga en pantalla una ventana recordándonos que hagamos algo que podría olvidársenos.
- Y muchísimas otras...

La herramienta **crontab** nos permite programar una tarea para que se ejecute especificando:

- La **hora** (0-23)
- Los **minutos** (0-59)
- El **día del mes** (1-31)

- El **día de la semana** (0-7: tanto *0* como *7* representan al domingo)
- El **mes** (1-12)

Y además en cualquiera de estos parámetros nos permitirá especificar, mediante un *, que deseamos que ocurra **todos** los minutos, las horas, los días del mes, los meses, o los días de la semana.

También podremos especificar **listas** separando mediante una coma, por ejemplo, podremos poner 1,15 en la columna del día del mes para que la tarea se ejecute cada dia 1 y 15 de mes.

Al igual que con las listas, podremos también especificar **rangos** mediante un guión, de manera que 1-4 en la columa del día de la semana significaría
de lunes a jueves
.

Los *crontab*'s de usuario.

En un sistema Linux, cada usuario tendrá su propio *crontab*, incluido root. Por supuesto, en el crontab de cada usuario sólo se permitirá ejecutar tareas para las que ese usuario tenga permisos (por ejemplo, un usuario que no sea root no podrá apagar el sistema).

Manejo del comando crontab.

Para usar la funcionalidad de *crontab*, el comando nos ofrece, de manera básica, las siguientes posibilidades:

Ver nuestra tabla de tareas programadas

Esto lo haremos con:

```
$ crontab -l
```

Editar nuestra tabla de tareas programas

Para editar nuestras tareas *cron*, ejecutaremos:

```
$ crontab -e
```

...y obtendremos un editor (el que hayamos definido en la variable de entorno $EDITOR) en el que podremos añadir líneas a nuestro fichero de crontab, una por cada tarea que deseemos programar, en el formato que se explicará más abajo.

Borrar nuestra tabla de tareas programas

Podremos borrar toda nuestra programación de tareas con:

```
$ crontab -r
```

El formato de las entradas de crontab.

Cuando añadamos una entrada a nuestro crontab, lo haremos en una sola línea y con el siguiente formato:

```
[minutos] [hora] [día] [mes] [dia_de_semana]        [comando]
```

Pongamos algunos ejemplos:

Para apagar el equipo a las siete y media de la mañana del 15 de febrero (ojo, hay que ser root para ejecutar esto):

```
30  7  15  2  *          /sbin/shutdown -h now
```

Para hacer un .tar.gz de un directorio (como backup, por ejemplo), todos los días laborables a las 14:00:

```
0  14  *  *  1-5          tar -czf /home/backups/backup.tar.gz /usr/importante/
```

Para que, cada hora, los sábados y domingos, se guarde en un fichero de log las conexiones de red abiertas en el sistema:

```
0  *  *  *  6,7        date >> /var/log/net_con.log; netstat -t >> /var/log/net_con.log
```

...Como se puede ver, es súmamente fácil de manejar y muy potente.