

Planificando Tareas con cron y at

¿ Que es cron ?

Cron se despierta una vez por minuto, lee las entradas conocidas y ejecuta las acciones que se le han indicado, de lo cual deducimos que no es necesario reiniciar el servicio **cron** cada vez que se añade una nueva entrada, ya que al minuto siguiente será el mismo el que comprueba si ha habido cambios. El demonio **cron** lee el archivo **/etc/crontab** y debe modificarse a mano. Cada usuario tiene su propio archivo y por tanto sus propias tareas. La sintaxis es la siguiente :

```
crontab [ -u user ] archivo
crontab [ -u user ] -l -e -r
```

La opción **-u** indica el usuario cuyo archivo **crontab** está apunto de utilizar. Si se omite la opción **-u** se asumirá que se está modificando el archivo **crontab**.

La opción **-l** le indica a **crontab** que enumere el archivo a la salida estándar.

La opción **-e** le indica a **crontab** que modifique el archivo, siendo el editor utilizado el definido en la variable **EDITOR** o **VISUAL**, si no existieran, se utilizaría **vi**.

La opción **-r** elimina el archivo **crontab** especificado. Si no se especifica el archivo, se eliminará el archivo **crontab** de ese usuario.

Los **/etc/cron.hourly**, **/etc/cron.daily**, **/etc/cron.monthly** y **/etc/cron.weekly** son ficheros que usa **cron** para su funcionamiento interno y los lanza cada día.

Ejemplo:

\$ vi /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 )
#
```

Los campos del **contrab** son los siguientes :

- Los minutos (0-59).
- Las horas (0-23).

- Día (1-31).
- Mes (1-12).
- Día de la semana (1-7).
- La orden a ejecutar con su respectivo path.

El símbolo / nos va a permitir afinar mucho mas el momento exacto de la ejecución de la tarea, por ejemplo, si deseamos que una determinada tarea se ejecuta durante todos los meses excepto junio, debemos indicarlo de la siguiente forma :

```
1-12/6
```

Si queremos que solo sea junio :

```
*/6
```

Vamos a comenzar podemos exportar la variable EDITOR para usar nuestro editor preferido :

```
$ export EDITOR=vim  
$ crontab -e
```

```
30 17 * * 1,7 tar cvfj /backup/etc.tar.bz2 /etc
```

Esto significa que se ejecute a las 17:30 de lunes a domingo todos los días, y todos los meses.

Mediante alguno de los siguiente archivos **/etc/cron.allow** y **/etc/cron.deny** podemos decir a quien le permitimos utilizar **cron** y a quien no.

Existe dentro del directorio **/var/spool/cron/crontabs** los usuarios que tiene tareas de ejecución.

¿ Que es at ?

A diferencia de **cron** con **at** vamos a poder realizar la ejecución de trabajos para una determinada hora. También existe el archivo **/etc/at.allow** y **/etc/at.deny** que es la misma idea de **cron**.

Ejemplo:

```
$ at 22:08  
at> ls -l /
```

Una vez que terminamos de escribir presionamos **CTRL+D** para salir y que quede grabado.

Para ver los programas que vamos a ejecutar :

```
$ atq
```

Para borrar un trabajo ejecutamos :

```
$ atrm numero_de_trabajo
```

Existe dentro del directorio **/var/spool/cron/atjobs** los usuarios que tiene tareas de ejecución.