Como usar Cron y Crontab en Linux

Es muy probable que todos hayan escuchado o utilizado en algún momento las tareas programadas de Windows, las cuales no son más que la configuración para que un archivo, programa o proceso, se ejecute en un tiempo determinado y bajo condiciones especificadas por nosotros los usuarios. Este mismo concepto aplica para Linux, sin embargo, el proceso no se realiza a través del entorno gráfico, para ello usamos la terminal. Por este motivo hoy traemos para ti la explicación de **cron** y **crontab** que son unos comandos esenciales para llevar a cabo estas tareas.

¿Qué es Cron?

Su nombre proviene de la expresión griega **chronos y su significado es tiempo**. Es uno de los **demonios** o "daemon" (proceso en segundo plano) más importantes y habituales en el sistema. Su ejecución comienza desde el primer instante de arranque.

Su función principal es encargarse de lanzar las **tareas programadas en fechas específicas** y de **forma automática y repetitiva**. La definición de las tareas se localiza en el archivo **/etc/crontab**. El funcionamiento es simple, verifica si existen tareas (jobs) para su ejecución de acuerdo al horario del sistema. Lo anterior nos lleva a resaltar que es importante mantener ajustada correctamente la zona horaria del sistema, pues de lo contrario no obtendremos los resultados esperados con cron.

De acuerdo a la distribución que estemos utilizando en Linux, se puede inicializar usando los **directorios /etc/init.d** o **etc/rc.d/** y en cada minuto realiza un chequeo de los **/etc/crontab** o **/var/spool/cron** ubicando posibles ejecuciones pendientes.

¿Qué es Crontab?

Es tan simple como un archivo de texto. Si, aunque no lo parezca. Lo que lo hace especial es su contenido. Su contenido especifica una **lista de todos los scripts a ser ejecutados por el sistem**a. Así como también especifica las fechas, horas y los permisos de ejecución de los mismos.

En Linux, generalmente **cada usuario tiene su propio archivo crontab** y el que está ubicado en el directorio **/etc**, es propiedad del usuario **root**.

Para generar tu propio archivo (en caso de no ser usuario root) solo haces uso del comando:

```
crontab
```

Crontab es la forma más simple para administrar las tareas de **cron en sistemas de tipo multiusuario**, ya sea como usuario root o simple usuario de sistema.
A continuación, les dejo una **pequeña ayuda con respecto a la definición del cron dentro** del crontab, con un ejemplo de muestra. (Tiene unos # adelante para que lo pueden colocar al principio de su archivo crontab como comentario y así siempre lo tendrán a mano).

```
# Ejemplo de definición de cron:
# .---------------- minutos (0 - 59)
# | .------------- horas (0 - 23)
# | | .---------- días del mes (1 - 31)
# | | | .------- meses (1 - 12) O jan,feb,mar,apr, ...
# | | | | .---- días de la semana (0 - 6) (Domingo=0 or 7)
# | | | | |
# * * * * * USER COMMAND
MAILTO="cron@localhost"
SHELL=/bin/sh
```

Donde cada asterisco representa una **fracción de tiempo** que determina el momento de la ejecución, seguido del usuario bajo el cual se realizará la ejecución (ese usuario puede ser root o uno que tenga los permisos de ejecución asignados) y por último el comando a ejecutar.

Administración de los jobs del Cron

Como ya has notado, el funcionamiento es muy simple. Para cerrar el tema, les mostraré los **comandos básicos** y esenciales para controlar y verificar el cron de nuestro sistema Linux.

Si queremos o requerimos modificar el archivo actual usamos el siguiente:

```
crontab -e
```

Para obtener la lista de todas las tareas que se encuentran configuradas en crontab, hacemos uso de:

```
crontab -l
```

Para eliminar el actual crontab del sistema, tenemos:

```
crontab -r
```

Fuentes: https://www.profesionalreview.com/2017/02/11/cron-y-crontab/

FORMATO DE LAS TAREAS

Si se deja un asterisco, quiere decir "cada" minuto, hora, día de mes, mes o día de la semana. Por ejemplo:

* * * * * /bin/ejecutar/script.sh

Ejecuta este script:

- Cada minuto
- De cada hora
- De cada día del mes
- De cada mes
- De cada día de la semana

Otro ejemplo:

30 2 * * 1 /bin/ejecutar/script.sh

Ejecuta este script:

- En el minuto 30
- De las 2 de la noche
- De cada día del mes
- De cada mes
- Sólo si es lunes

En resumen, todos los lunes a las 2:30 horas se ejecutará el script.

Intervalos de tiempo

Ejecuta un script de lunes a viernes a las 2:30 horas:

30 2 * * 1-5 /bin/ejecutar/script.sh

Ejecuta un script de lunes a viernes cada 10 minutos desde las 2:00 horas durante una hora:

0,10,20,30,40,50 2 * * 1-5 /bin/ejecutar/script.sh

Esto quizá puede ser largo. La sintaxis de crontab permite lo siguiente. Imaginemos que queremos ejecutarlo cada 5 minutos:

*/5 2 * * 1-5 /bin/ejecutar/script.sh

PALABRAS RESERVADAS

Muchas veces tenemos palabras reservadas para facilitar el uso de programas o lenguajes de programación. **Cron no podía ser menos**, así que tenemos algunas que suelen ser las más comunes. Ya cada uno que lo configure conforme a sus necesidades. Aquí van:

- @reboot: se ejecuta una única vez al inicio.
- @yearly/@annually: ejecutar cada año.
- @monthly: ejecutar una vez al mes.
- @weekly: una vez a la semana.
- @daily/@midnight: una vez al día.
- @hourly: cada hora.

Por ejemplo, **para ejecutar el script cada hora**:

@hourly /bin/ejecutar/script.sh