MANUAL BÁSICO DE CRON

autor: sergio.gonzalez.duran@gmail.com

Esta es una introducción a cron, cubre lo básico de lo que cron puede hacer y la manera de usarse.

¿Qué es cron?

Cron es el nombre del programa que permite a usuarios Linux/Unix ejecutar automáticamente comandos o scripts (grupos de comandos) a una hora o fecha específica. Es usado normalmente para comandos de tareas administrativas, como respaldos, pero puede ser usado para ejecutar cualquier cosa. Como se define en las páginas del manual de cron (#> man cron) es un demonio que ejecuta programas agendados.

En prácticamente todas las distribuciones de Linux se usa la versión Vixie Cron, por la persona que la desarrolló, que es Paul Vixie, uno de los grandes gurús de Unix, también creador, entre otros sistemas, de BIND que es uno de los servidores DNS más populares del mundo.

Iniciar cron

Cron es un demonio (servicio), lo que significa que solo requiere ser iniciado una vez, generalmente con el mismo arranque del sistema. El servicio de cron se llama crond. En la mayoría de las distribuciones el servicio se instala automáticamente y queda iniciado desde el arranque del sistema, se puede comprobar de varias maneras:

```
#> /etc/rc.d/init.d/crond status
#> /etc/init.d/crond status   Usa cualquiera de los dos dependiendo de tu distro
crond (pid 507) is running...

o si tienes el comando service instalado:
#> service crond status
crond (pid 507) is running...

se puede también revisar a través del comando ps:
# ps -ef   |  grep crond

si por alguna razón, cron no esta funcionando:
#> /etc/rc.d/init.d/crond start
Starting crond:            [ OK ]
```

Si el servicio no estuviera configurado para arrancar desde un principio, bastaría con agregarlo con el comando chkconfig:

```
#> chkconfig --level 35 crond on
```

Usando cron

Hay al menos dos maneras distintas de usar cron:

La primera es en el directorio /etc, donde muy seguramente encontrarás los siguientes directorios:

- cron.hourly
- cron.daily
- cron.weekly
- cron.monthly

Si se coloca un archivo tipo script en cualquiera de estos directorios, entonces el script se ejecutará cada hora, cada día, cada semana o cada mes, dependiendo del directorio.

Para que el archivo pueda ser ejecutado tiene que ser algo similar a lo siguiente:

```
        #!/bin/sh
#script que genera un respaldo
cd /usr/documentos
tar czf * respaldo
cp respaldo /otra_directorio/.
```

Nótese que la primera línea empieza con #!, que indica que se trata de un script shell de bash, las demás líneas son los comandos que deseamos ejecute el script. Este script podría nombrarse por ejemplo respaldo.sh y también debemos cambiarle los permisos correspondientes para que pueda ser ejecutado, por ejemplo:

```
#> chmod 700 respaldo.sh
#> ls -l respaldo.sh
-rwx------ 1 root root 0 Jul  20 09:30 respaldo.sh
```

La "x" en el grupo de permisos del propietario (rwx) indica que puede ser ejecutado.

Si este script lo dejamos en cron.hourly, entonces se ejecutará cada hora con un minuto de todos los días, en un momento se entenderá el porque.

Como segundo modo de ejecutar o usar cron es a través de manipular directamente el archivo /etc/crontab. En la instalación por defecto de varias distribuciones Linux, este archivo se verá a algo como lo siguiente:

```
#> cat /etc/crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

# run-parts
01 * * * * root run-parts /etc/cron.hourly
02 4 * * * root run-parts /etc/cron.daily
22 4 * * 0 root run-parts /etc/cron.weekly
42 4 1 * * root run-parts /etc/cron.monthly
```

Las primeras cuatro líneas son variables que indican lo siguiente:

SHELL es el 'shell' bajo el cual se ejecuta el cron. Si no se especifica, se tomará por defecto el indicado en la línea /etc/passwd correspondiente al usuario que este ejecutando cron.

PATH contiene o indica la ruta a los directorios en los cuales cron buscará el comando a ejecutar. Este path es distinto al path global del sistema o del usuario.

MAIL TO es a quien se le envía la salida del comando (si es que este tiene alguna salida). Cron enviará un correo a quien se especifique en este variable, es decir, debe ser un usuario válido del sistema o de algún otro sistema. Si no se especifica, entonces cron enviará el correo al usuario propietario del comando que se ejecuta.

HOME es el directorio raíz o principal del comando cron, si no se indica entonces, la raíz será la que se indique en el archivo /etc/passwd correspondiente al usuario que ejecuta cron.

Los comentarios se indican con # al inicio de la línea.

Después de lo anterior vienen las líneas que ejecutan las tareas programadas propiamente. No hay límites de cuantas tareas pueda haber, una por renglón. Los campos (son 7) que forman estas líneas están formados de la siguiente manera:

```
Minuto  Hora  DiaDelMes  Mes  DiaDeLaSemana  Usuario  Comando
```

Campo	Descripción
Minuto	Controla el minuto de la hora en que el comando será ejecutado, este valor debe de estar entre 0 y 59.
Hora	Controla la hora en que el comando será ejecutado, se especifica en un formato de 24 horas, los valores deben estar entre 0 y 23, 0 es medianoche.
Día del Mes	Día del mes en que se quiere ejecutar el comando. Por ejemplo se indicaría 20, para ejecutar el comando el día 20 del mes.
Mes	Mes en que el comando se ejecutará, puede ser indicado numéricamente (1-12), o por el nombre del mes en inglés, solo las tres primeras letras.
Día de la semana	Día en la semana en que se ejecutará el comando, puede ser numérico (0-7) o por el nombre del día en inglés, solo las tres primeras letras. (0 y 7 = domingo)
Usuario	Usuario que ejecuta el comando.
Comando	Comando, script o programa que se desea ejecutar. Este campo puede contener múltiples palabras y espacios.

Un asterisco * como valor en los primeros cinco campos, indicará inicio-fin del campo, es decir todo. Un * en el campo de minuto indicará todos los minutos.

Para entender bien esto de los primeros 5 campos y el asterisco usaré mejor varios ejemplos:

Ejemplo	Descripción
01 * * * *	Se ejecuta al minuto 1 de cada hora de todos los días
15 8 * * *	A las 8:15 a.m. de cada día
15 20 * * *	A las 8:15 p.m. de cada día
00 5 * * 0	A las 5 a.m. todos los domingos
* 5 * * Sun	Cada minuto de 5:00a.m. a 5:59a.m. todos los domingos

45 19 1 * *	A las 7:45 p.m. del primero de cada mes
01 * 20 7 *	Al minuto 1 de cada hora del 20 de julio
10 1 * 12 1	A la 1:10 a.m. todos los lunes de diciembre
00 12 16 * Wen	Al mediodía de los días 16 de cada mes y que sea Miércoles
30 9 20 7 4	A las 9:30 a.m. del dia 20 de julio y que sea jueves
30 9 20 7 *	A las 9:30 a.m. del dia 20 de julio sin importar el día de la semana
20 * * * 6	Al minuto 20 de cada hora de los sábados
20 * * 1 6	Al minuto 20 de cada hora de los sábados de enero

También es posible especificar listas en los campos. Las listas pueden estar en la forma de 1,2,3,4 o en la forma de 1-4 que sería lo mismo. Cron, de igual manera soporta incrementos en las listas, que se indican de la siguiente manera:

```
Valor o lista/incremento
```

De nuevo, es más fácil entender las listas e incrementos con ejemplos:

Ejemplo	Descripción
59 11 * 1-3 1,2,3,4,5	A las 11:59 a.m. de lunes a viernes, de enero a marzo
45 * 10-25 * 6-7	Al minuto 45 de todas las horas de los días 10 al 25 de todos los meses y que el día sea sábado o domingo
10,30,50 * * * 1,3,5	En el minuto 10, 30 y 50 de todas las horas de los días lunes, miércoles y viernes
*/15 10-14 * * *	Cada quince minutos de las 10:00a.m. a las 2:00p.m.
* 12 1-10/2 2,8 *	Todos los minutos de las 12 del día, en los días 1,3,5,7 y 9 de febrero a agosto. (El incremento en el tercer campo es de 2 y comienza a partir del 1)
0 */5 1-10,15,20-23 * 3	Cada 5 horas de los días 1 al 10, el día 15 y del día 20 al 23 de cada mes y que el día sea miércoles
3/3 2/4 2 2 2	Cada 3 minutos empezando por el minuto 3 (3,6,9, etc.) de las horas 2,6,10, etc (cada 4 horas empezando en la hora 2) del día 2 de febrero y que sea martes

Como se puede apreciar en el último ejemplo la tarea cron que estuviera asignada a ese renglón con esos datos, solo se ejecutaría si se cumple con los 5 campos (AND). Es decir, para que la tarea se ejecute tiene que ser un martes 2 de febrero a las 02:03. Siempre es un AND booleano que solo resulta verdadero si los 5 campos son ciertos en el minuto específico.

El caso anterior deja claro entonces que:

El programa cron se invoca cada minuto y ejecuta las tareas que sus campos se cumplan en ese preciso minuto.

Incluyendo el campo del usuario y el comando, los renglones de crontab podrían quedar entonces de la siguiente manera:

```
0 22 * * * root /usr/respaldodiario.sh
0 23 * * 5 root /usr/respaldosemanal.sh
0 8,20 * * * sergio mail -s "sistema funcionando" sgd@ejemplo.com
```

Las dos primeras líneas las ejecuta el usuario root y la primera ejecuta a las 10 de la noche de todos los días el script que genera un respaldo diario. La seguna ejecuta a las 11 de la noche de todos los viernes un script que genera un respaldo semana. La tercera línea la ejecuta el usuario sergio y se ejecutaría a las 8 de la mañana y 8 de la noche de todos los día y el comando es enviar un correo a la cuenta sgd@ejemplo.com con el asunto "sistema funcionando", una manera de que un administrador este enterado de que un sistema remoto esta activo en las horas indicadas, sino recibe un correo en esas horas, algo anda mal.

Siendo root, es posible entonces, modificar directamente crontab:

```
#> vi /etc/crontab
```

Ejecutando Cron con múltiples usuarios, comando crontab

Linux es un sistema multiusuario y cron es de las aplicaciones que soporta el trabajo con varios usuarios a la vez. Cada usuario puede tener su propio archivo crontab, de hecho el /etc/crontab se asume que es el archivo crontab del usuario root, aunque no hay problema que se incluyan otros usuarios, y de ahí el sexto campo que indica precisamente quien es el usuario que ejecuta la tarea y es obligatorio en /etc/crontab.

Pero cuando los usuarios normales (e incluso root) desean generar su propio archivo de crontab, entonces utilizaremos el comando crontab.

En el directorio /var/spool/cron (puede variar según la distribución), se genera un archivo cron para cada usuario, este archivo aunque es de texto, no debe editarse directamente.

Se tiene entonces, dos situaciones, generar directamente el archivo crontab con el comando:

```
$> crontab -e
```

Con lo cual se abrira el editor por default (generalemente vi) con el archivo llamado crontab vacio y donde el usuario ingresará su tabla de tareas y que se guardará automáticamente como /var/spool/cron/usuario.

El otro caso es que el usuario edite un archivo de texto normal con las entradas de las tareas y como ejemplo lo nombre 'mi_cron', después el comando $> crontab mi_cron se encargará de establecerlo como su archivo cron del usuario en /var/spool/cron/usuario:

```
$> vi mi_cron
# borra archivos de carpeta compartida
0 20 * * * rm -f /home/sergio/compartidos/*
# ejecuta un script que realiza un respaldo de la carpeta documentos el primer
día de cada mes
0 22 1 * * /home/sergio/respaldomensual.sh
# cada 5 horas de lun a vie, se asegura que los permisos sean los correctos en
mi home
1 *5 * * * 1-5 chmod -R 640 /home/sergio/*
:wq (se guarda el archivo)
$> ls
mi_cron
$> crontab mi_cron
(se establece en /var/spool/cron/usuario)
```

Resumiendo lo anterior y considerando otras opciones de crontab:

```
$> crontab archivo.cron (establecerá el archivo.cron como el crontab del
```

```
usuario)
$> crontab -e           (abrirá el editor preestablecido donde se podrá crear o
editar el archivo crontab)
$> crontab -l           (lista el crontab actual del usuario, sus tareas de
cron)
$> crontab -r           (elimina el crontab actual del usuario)
```

En algunas distribuciones cuando se editan crontabs de usuarios normales es necesario reiniciar el servicio para que se puedan releer los archivos de crontab en /var/spool/cron.

```
#> service crond restart
```

Para entender mejor como iniciar/detener/reiniciar servicios, en este artículo encontrarás más información.

Controlando el acceso a cron

Cron permite controlar que usuarios pueden o no pueden usar los servicios de cron. Esto se logra de una manera muy sencilla a través de los siguientes archivos:

- /etc/cron.allow
- /etc/cron.deny

Para impedir que un usuario utilice cron o mejor dicho el comando crontab, basta con agregar su nombre de usuario al archivo /etc/cron.deny, para permitirle su uso entonces sería agregar su nombre de usuario en /etc/cron.allow, si por alguna razón se desea negar el uso de cron a todos los usuarios, entonces se puede escribir la palabra ALL al inicio de cron.deny y con eso bastaría.

```
#> echo ALL >>/etc/cron.deny
o para agregar un usuario mas a cron.allow
#> echo juan >>/etc/cron.allow
```

Si no existe el archivo cron.allow ni el archivo cron.deny, en teoría el uso de cron esta entonces sin restricciones de usuario. Si se añaden nombres de usuarios en cron.allow, sin crear un archivo cron.deny, tendrá el mismo efecto que haberlo creado con la palabra ALL. Esto quiere decir que una vez creado cron.allow con un solo usuario, siempre se tendrán que especificar los demás usuarios que se quiere usen cron, en este archivo.

Espero que este pequeño manual sobre cron te sea de utilidad, por favor, si encuentras algún error, házmelo saber para corregirlo, gracias.