Qué es crontab

Crontab es un simple archivo de texto que guarda una lista de comandos a ejecutar en un tiempo especificado por el usuario. Crontab verificará la fecha y hora en que se debe ejecutar el script o el comando, los permisos de ejecución y lo realizará en el background. Cada usuario puede tener su propio archivo crontab, de hecho el */etc/crontab* se asume que es el archivo crontab del usuario root, cuando los usuarios normales (e incluso root) desean generar su propio archivo de crontab, entonces utilizaremos el comando crontab.

Crontab es la manera mas sencilla de administrar tareas de cron en sistemas ultiusuario, ya sea como simple usuario de sistema o usuario root.

Agregar tareas a crontab

Ejecutamos la edición del crontab con crontab -e, en algunas distros (como ubuntu) nos da la opcion de elegir el editor de textos que deseemos, los demás nos quedamos con vi.

```
$> 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)
```

```
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.

U

El archivo crontab mostrará algo asi.

```
# m h dom mon dow user command
```

donde:

- **m** corresponde al minuto en que se va a ejecutar el script, el valor va de 0 a 59
- **h** la hora exacta, se maneja el formato de 24 horas, los valores van de 0 a 23, siendo 0 las 12:00 de la medianoche.
- **dom** hace referencia al día del mes, por ejemplo se puede especificar 15 si se quiere ejecutar cada dia 15
- **dow** significa el día de la semana, puede ser numérico (0 a 7, donde 0 y 7 son domingo) o las 3 primeras letras del día en inglés: mon, tue, wed, thu, fri, sat, sun.
- **user** define el usuario que va a ejecutar el comando, puede ser root, u otro usuario diferente siempre y cuando tenga permisos de ejecución del script.
- **command** refiere al comando o a la ruta absoluta del script a ejecutar, ejemplo: */home/usuario/scripts/actualizar.sh*, si acaso llama a un script este debe ser ejecutable

Para que quedara claro unos cuantos ejemplos de tareas de cron explicados:

```
15 10 * * * usuario /home/usuario/scripts/actualizar.sh
```

Ejecutará el script actualizar.sh a las 10:15 a.m. todos los días

```
15 22 * * * usuario /home/usuario/scripts/actualizar.sh
```

Ejecutará el script actualizar.sh a las 10:15 p.m. todos los días

```
00 10 * * 0 root apt-get -y update Usuario root
```

Ejecutará una actualización todos los domingos a las 10:00 a.m

```
45 10 * * sun root apt-get -y update
```

Administracion de trabajos en cron

```
crontab archivo
```

Remplaza el existente archivo crontab con un archivo definido por el usuario

```
crontab -e
```

Editar el archivo crontab del usuario, cada linea nueva sera una nueva tarea de crontab.

```
crontab -l
```

Lista todas las tareas de crontab del usuario

```
crontab -d
```

Borra el crontab del usuario

```
crontab -c dir
```

Define el directoriod e crontab del usuario (este debe tener permisos de escritura y ejecucion del usuario)

```
crontab -u usuario
```

prefijo para manejar el crontab de otro usuario, ejemplos:

```
$ sudo crontab -l -u root
$ sudo crontab -e usuario2
#crontab -d -u usuario
```