

Cron y Crontab

En sistemas operativos basados en Unix, como es el caso de GNU/Linux, **cron** es un administrador regular de procesos en segundo plano (**daemons**) que ejecuta **procesos** o **scripts** a intervalos regulares (por ejemplo, cada minuto, día, semana o mes). Los procesos que deben ejecutarse y la hora en la que deben hacerlo se especifican en el archivo **crontab**. El nombre cron viene del griego chronos (χρόνος) que significa "tiempo".

Para agregar, quitar o modificar tareas, hay que editar el archivo **crontab**. Esto se hace mediante el comando **crontab -e**, que abrirá el editor definido en la variable de entorno EDITOR y **cargará el archivo crontab correspondiente al usuario que está logueado**.

Cada vez que se ejecuta el crontab, se envía un mensaje al usuario que aparece en la variable de entorno MAILTO, si está habilitado, indicándole la tarea realizada.

La ejecución de las tareas programadas es controlada por el servicio **crond** (cron daemon), por lo que es necesario verificar que éste esté ejecutándose en el sistema. El comando: **"ps ax |grep cron"** Debería mostrarnos que se está ejecutando el programa: **"/usr/sbin/cron"**

Si no es así, podemos arrancarlo con el comando **"/etc/init.d/cron start"**



Los usuarios habilitados para crear su archivo **crontab** se especifican en el archivo **cron.allow**. De manera análoga, los que no lo tienen permitido figuran en **cron.deny**. Estos dos últimos archivos se encuentran en **/etc/cron.d/**, o **/etc/**, dependiendo de la versión de Gnu/Linux.

Agregando /editando tareas programadas

Para agregar, quitar o modificar tareas, hay que editar el crontab.

El comando **crontab -e**, abrirá el editor y cargará el archivo **crontab** correspondiente al **usuario que está logueado**.

Si se está logueado con una **sesión root**, puede cargarse el archivo **crontab** de cualquier usuario mediante el comando **crontab -e -u nombre-del-usuario**.

Formato del archivo **crontab**:

```
# Each crontab file entry consists of a line with six fields,
# separated by spaces and tabs, that contain, respectively:
#
# * The minute (0 through 59)
# * The hour (0 through 23)
# * The day of the month (1 through 31)
# * The month of the year (1 through 12)
# * The day of the week (0 through 6 for Sunday through Saturday)
# * The shell command
#
# Proceso Diario: Respallos Incrementales de Directorios marcados
# Todos los días, a las 02:00 hrs. (AM)
0 2 * * * /home/admweb/scripts/shells/diario.sh
#Respaldo total por reinstalacion
0 2 23 6 * /home/admweb/scripts/shells/mensual.sh
# Proceso mensual: Respaldo Total de los Directorios marcados
# Dia 15 del mes a las 2:15 hrs. (AM)
15 2 15 * * /home/admweb/scripts/shells/mensual.sh
#20 11 28 * * /home/admencu/scripts/shells/test_mail.sh
# Reintento de respaldos fallidos
# =====
# Cada media hora (*:10, *:40)
10,40 * * * * /home/admweb/scripts/shells/bkp_retry.sh
~
~
~
```

Sintaxis

```
#Sincronizacion de hora
#-----
1      *      *      *      *      /usr/sbin/ntpdate werken.ufro.cl
```

a) El símbolo “#” es un **comentario**, todo lo que se encuentre después de ese carácter no será ejecutado por cron.

b) El **momento de ejecución** se especifica de acuerdo con la siguiente tabla:

Minutos: (0-59)

Horas: (0-23)

Días: (1-31)

Mes: (1-12)

Día de la semana (0-6): 1=lunes, 2=martes,... 6=sábado y 0=domingo (a veces también 7=domingo)

c) El asterisco “*” se usa como comodín. Es decir, representa todos los valores posibles de la variable.

d) La última columna corresponde a la **ruta absoluta del binario o script** que se quiere ejecutar.

Obs: El valor para cada variable debe ir separado por un “tab” (tecla tabulador).



Actividad práctica

1. Iniciar sesión como usuario “**root**”.
2. Crear los directorios **scripts** y **logs** en el directorio **root**.
3. Dentro del directorio **scripts** escribir el siguiente script (mensaje.sh):

```
#!/bin/bash
fecha = `date`
echo $fecha, “Este es un Mensaje” >> /root/logs/mensaje.log
```

4. Ingresar el script en el crontab para que se ejecute cada 1 minuto.
5. Revisar el archivo de salida (/root/logs/mensaje.log): `tail -f /root/logs/mensaje.log`
6. Modificar el script anterior para que cree un archivo de log por cada vez que se ejecute.