

PROGRAMACIÓN DE TAREAS

<http://www.adminschoice.com/crontab-quick-reference>

Existen una serie de tareas que en determinados momentos nos interesaría ejecutar sin estar delante del ordenador. Por ejemplo, realizar copias de seguridad por la noche, programar el apagado automático a una hora por si se nos olvida apagar el equipo (muy útil en empresas) o saber en cada momento qué usuarios están conectados en una máquina. Ubuntu (y Linux en general) proporcionan una serie de aplicaciones para llevar a cabo esta tarea. Pasemos a estudiarlas.

Antes de programar las tareas hay que comprobar que el servicio cron se encuentra en ejecución mediante el comando:



```
#service cron status
```

Para modificar el fichero de configuración de *cron*, ejecuta el comando:



```
#crontab -e
```

y aparece un fichero con el siguiente formato:

```
PATH=/bin
0 0 * * * /root/comprobar_seguridad.sh
0 0 1 * * /root/copia_seguridad.sh
```

La sintaxis de las tareas programadas es:

```
# .----- minuto (0 - 59)
# | .----- hora (0 - 23)
# | | .----- día del mes (1 - 31)
# | | | .----- mes (1 - 12) o jan,feb,mar,apr ... (los meses
en inglés)
# | | | | .---- día de la semana (0 - 6) (Domingo = 0 o 7) o
sun,mon,tue,wed,thu,fri,sat (los días en inglés)
# | | | | |
# | | | | |
* * * * * Comando a ejecutar
```

En el ejemplo anterior se ejecuta el script *comprobar_seguridad.sh* todos los días a las 0:00h y se ejecuta *copia_seguridad.sh* el primer día de cada mes.

Además, de este método para programar tareas, *crontab* también permite al administrador que cree tareas modificando el fichero */etc/crontab*, con el siguiente formato casi idéntico a lo visto anteriormente:

55	23	*	*	0	root	/usr/local/sbin/copiasemanal.sh
Rango 0 - 59	Rango 0 - 23	Rango 1 - 31	Rango 1 - 12	Rango 0 - 6		Comando
					Usuario	
					Día de la semana	Lunes = 1, Martes = 2, Miércoles = 3 Jueves = 4, Viernes = 5, Sábado = 6, Domingo = 0
					Mes	Enero = 1, Febrero = 2, Marzo = 3, Abril = 4, Mayo = 5, Junio = 6, Julio = 7 Agosto = 8, Septiembre = 9, Octubre = 10, Noviembre = 10, Diciembre = 12
					Día del mes	
					Hora	
					Minuto	

Ejecuta *copiasemanal.sh* cada domingo a las 23:55

Por último, otra forma de poder programar tareas es guardar el script o programa que quiere ejecutar en las siguientes carpetas de configuración de *cron*:

```
/etc/cron.hourly      # Ejecuta el script cada hora
/etc/cron.daily       # Ejecuta el script diariamente
/etc/cron.weekly      # Ejecuta el script semanalmente
/etc/cron.monthly     # Ejecuta el script mensualmente
```

Para asegurar el sistema sólo el usuario root puede modificar los scripts que ejecuta crontab.

Gnome-schedule

Además de *crontab*, disponemos de una aplicación para GNOME, que nos permite administrar las tareas de *cron* de manera gráfica y sencilla llamada “Gnome Schedule”.

Este programa gráfico es la interfaz gráfica del servicio *cron*, el cual lee un fichero - *crontab* - donde le indicamos las tareas y cuando las queremos realizar y él se encarga de ejecutarlas por nosotros en el preciso momento que le hayamos indicado en *crontab* , por lo que tenemos primero que:

Crear un archivo crontab para cada usuario que quiera usar este programa (naturalmente hacemos esto desde cada sesión de un usuario):

1-Primero creamos con *gedit* un archivo de texto, con el texto:

```
SHELL=/bin/bash
DISPLAY=:0
```

2-Guardas este archivo con el nombre: cron-usuario , en el directorio /home/usuario (Con cada usuario el archivo cron-usuario llevara el nombre del usuario)
3-En la terminal ejecutas: crontab cron-usuario (luego si quieres ya puedes borrar el archivo de texto cron-usuario).

1) Para instalarlo ponemos:



```
#apt-get install gnome-schedule
```

2) Para ejecutarlo:

- Desde entorno gráfico buscar: Tareas programadas (como el usuario de la sesión).

Creación de tareas con gnome-schedule.

Podemos crear 3 tipos de tareas:

1. Tareas repetitivas.
2. Tareas que ocurrirán una única vez.
3. Tareas desde una plantilla predefinida. Permite crear una tarea a partir de otra creada previamente y guardada como plantilla.

OPCIONES:

== En *Descripción*: ponemos un nombre que nos diga lo que va a hacer la tarea.

== En *Comando*: ponemos lo siguiente para crear una tarea para ejecutar una de estas tres cosas:

-- **un comando**: hay que poner el *comando*, por ejemplo: tar cvf backup-firefox.tar /home/usuario/.mozilla

-- **un script**: hay que poner la *ruta completa* de donde esta el script: por ejemplo: /home/usuario/Scripts/copia-de-firefox

-- **un programa gráfico**: el nombre del *ejecutable* de un programa gráfico, por ejemplo: sylpheed

== En *Comportamiento*: se puede elegir entre:

-- *Comportamiento predeterminado*: para un programa gráfico

-- *Suprimir salida*: si se ejecuta de forma automática sin necesidad de que intervenga para nada el usuario.

-- *Aplicación X*: para un comando o script en el que se necesite contestar en la terminal

-- *Aplicación X: suprimir salida*: para un comando o script en el que no se necesite contestar en la terminal, sino que se ejecute de forma automática.

Tareas repetitivas.

Supongamos que queremos apagar el equipo todos los días a las 15:00h.

- 1) Pulsamos el botón “Nuevo”.
- 2) Pulsamos el botón “Una tarea que se lanzará repetidamente”.
- 3) Rellenamos el formulario que nos aparece como sigue:

- Descripción: Apagado automático.
- Comando: /sbin/halt.
- Seleccionamos Comportamiento predeterminado. Si la aplicación fuera gráfica seleccionaríamos “Aplicación X”.
- Seleccionamos en “Hora y fecha” la opción “Avanzado” y modificamos el campo “hora” poniendo el valor 15.
- Pulsamos el botón “Añadir”.

- 4) Pulsamos el botón “Aceptar”.

Tareas no repetitivas.

Supongamos que queremos reiniciar el entorno gráfico en una hora determinada:

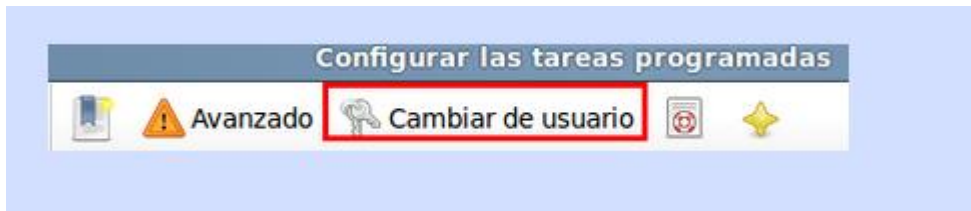
- 1) Pulsamos el botón “Nuevo”.
- 2) Pulsamos el botón “Una tarea que se lanza una sola vez”.
- 3) Rellenamos el formulario que nos aparece como sigue:

- Descripción de la tarea: Reiniciar entorno gráfico.
- Como la aplicación a ejecutar no es gráfica, no marcamos la opción “Aplicación X”.
- Fecha y tiempo: Ponemos la fecha y hora deseada.
- Tarea: /etc/init.d/gdm restart.
- Pulsamos el botón “Añadir”.

- 4) Pulsamos el botón “Aceptar”.

Gnome-Schedule con permisos de root: El programa Gnome-Schedule se puede usar también con permisos de root, si queremos que las tareas las realice el root, ejecutando: `sudo gnome-schedule`; Si queremos que se ejecute un script o comando que en su sintaxis lleva el comando `sudo`, es decir que necesita permisos de root para ejecutarse, tendremos que usar Gnome-Schedule con permisos de root para ejecutar dicha tarea.

Desde el programa Gnome-Schedule del usuario (y de cada usuario si hay varios) y desde el del root, se accede cada uno a sus propias tareas. Desde el Gnome-Schedule ejecutado con permisos de root se puede acceder (y modificar) a las tareas de cada uno de los usuarios que tengamos en el sistema, a través de una nueva opción (cambiar de usuario) que aparece en la barra de herramientas del programa.



Ejemplos de crontab

Veamos algunos ejemplos de programación de tareas en cron:

0 23 10 * * /home/usuario/copia.sh

- Ejecuta el script “copia.sh” todos los días 10 de cada mes a las 11:00 de la noche.

55 23 * * 0 /home/usuario/copia_semanal.sh

- Ejecutará el script “copia_semanal” todos los domingos a las 23:55h.

0 9,17 * * 1-5 /home/usuario/copia.sh

- Ejecuta el script “copia” de lunes a viernes a las 9:00h y a las 17:00h.

30 17 * * 1,2,3,4,5 /home/usuario/copia.sh

- Ejecuta el script a las 17:30h todos los días de lunes a viernes.

00 12 1,15,28 * * /home/usuario/copia.sh

- Ejecuta el script a las 12 a.m. todos los días 1, 15 y 28 de cada mes.

Comando AT

El comando o programa **at**, permite planificar la ejecución de ciertas tareas en un momento dado.

Tanto el administrador del sistema como los usuarios tienen acceso a esta herramienta.

Creación de tareas.

El comando “at” tiene la siguiente sintaxis:

at [opciones] [hora] [fecha]

Al ejecutar el comando **at** el sistema se queda esperando a que el usuario introduzca una serie de comandos que quiere ejecutar en el día y hora indicado. Para terminar de introducir comandos, hay que pulsar la combinación de teclas <CTRL+D>.

El comando `at` acepta horas con formato HH:MM para ejecutar un trabajo a una determinada hora del día (si esa hora ya ha pasado, se asume que es del día siguiente) y fechas con el formato MMDDAA o MM/DD/AA.

Otra opción muy útil es: “now + X unidades”, que cuenta a partir de ahora más un tiempo indicado, siendo las unidades permitidas “minutes (minutos), hours (horas) o days (días)”. Para ejecutar una tarea dentro de 5 minutos:



```
$at now + 5 minutes
```

Ejemplo:



```
$at 21:00 102411
at> cp /etc/network/interfaces /backup
at> <CTRL+D>
job 1 at Mon Oct 24 21:00:00 2011
```

- Lo que hará este ejemplo es copiar el fichero “interfaces” en el directorio /backup el 24 octubre de 2011 a las 21:00 horas.

Ver listado de tareas creadas.

Para ver las tareas que están pendientes de ejecución podemos poner:



```
$atq
```

O también:



```
$at -l
```

Eliminar tareas.

Para eliminar tareas creadas hacemos lo siguiente:



```
$atrm <número de tarea>
```

O también:



```
$at -d <número de tarea>
```

El número de tarea lo podemos saber usando atq o at -l.