
Automatización de tareas

Diseño y Administración de Sistemas y Redes

Juan Céspedes <cespedes@gsync.es>



Curso 2005–2006



Tareas periódicas.

- Automatizan la gestión del sistema.
- Fiabilidad. Se llevan a cabo a pesar de posibles olvidos o descuidos por parte del administrador.
- Se ejecutan en el momento preciso (día y hora).
- Ayudan o detectan situaciones de error.
- Facilitan el control del sistema.
- Posibilidad de tareas por usuario.
- Programas: `cron` y `at`.

Usos de las tareas periódicas

- Generación de informes periódicos (fin de mes, etc.).
- Estado de las comunicaciones.
- Borrado de ficheros temporales (`/tmp`, `/var/tmp`).
- Tareas de respaldo de información.
- Control de los procesos presentes en el sistema.
- Parada del sistema según horarios de trabajo.
- Recordatorios.
- Descarga de *software* en horarios de poco tráfico.

cron

- Es uno de los demonios esenciales de un sistema, siempre está arrancado (`/usr/sbin/cron`).
- Se encarga de ejecutar tareas programadas para un determinado momento, bajo la identidad del usuario que lo programó y con precisión de 1 minuto.
- Se controla a través del uso de determinados ficheros de configuración (solo para el superusuario) y mediante el uso de la orden "`crontab`" (para todos los usuarios).

crontab

```
crontab [ -u usuario ] fichero  
crontab [ -u usuario ] { -l | -r | -e }
```

- “**-u usuario**”: disponible solo para **root**, permite ver o modificar las tareas programadas para otro usuario.
- “**fichero**”: Reemplaza la lista de tareas programadas por las que aparezcan en el fichero indicado.
- “**-l**”: Muestra la lista de todas las tareas programadas.
- “**-r**”: Elimina todas las tareas programadas.
- “**-e**”: Edita la lista de tareas programadas.

Ficheros de configuración de **crontab**

Normalmente, la orden **crontab** puede ejecutarla cualquier usuario, permitiendo que incluya tareas periódicas en el sistema, pero existen un par de ficheros de configuración para limitar su uso: **/etc/cron.allow** y **/etc/cron.deny**.

- “**/etc/cron.allow**”. Si existe, indica que solamente la lista de usuarios incluidos en ese fichero (un usuario por línea) tiene permiso para utilizar **crontab**.
- “**/etc/cron.deny**”. Si no existe **/etc/cron.allow**, y existe este fichero, los usuarios incluidos en él no tienen permiso para utilizar **crontab**.

Lista de tareas para *cron*

- Cada tarea se especifica en una línea diferente, en la que se indica el momento en el que debe realizarse, y qué se debe ejecutar.
- Cada línea contiene 6 campos separados por espacios:
 - minuto (0–59)
 - hora (0–23)
 - día del mes (1–31)
 - mes (1–12)
 - día de la semana (0–7). 0=7=domingo, 1=lunes, 2=martes. . .
 - orden a ejecutar

Formato de cada línea en *cron*

Cada campo numérico (del primero al quinto de cada línea), puede tener:

- Un “*”, que indica “todos”.
- Un número, que indica un momento exacto (“4”).
- Lista de números separadas por comas (“3,4,6”).
- Rango de números separados por guiones (“3-7”).

Ejemplo:

```
0 9,18 * * 1-5 echo "hora de comer" | wall
```

(ejecuta la orden de lunes a viernes a las 9:00 y a las 18:00).

Ambigüedades en la especificación del momento de ejecución

- El día en el que se ejecuta cada orden se puede indicar de 2 maneras:
 - día del mes (3^{er} campo)
 - día de la semana (5^o campo)

En caso de aparecer los dos campos (esto es, que ninguno es “*”), la interpretación que hace **cron** es que la orden debe ejecutarse cuando se cumpla *cualquiera* de ellos.

Ejemplo:

```
0,30 * 13 * 5 echo 'Viernes 13!' | wall
```

(ejecuta la orden cada media hora, **todos los viernes** y además **todos los días 13 de cada mes**).

Momentos “especiales” (solo Linux)

En lugar de especificar los 5 primeros campos, se puede usar una cadena de las siguientes:

- **@reboot**: Se ejecuta al iniciarse la máquina.
- **@yearly**: Se ejecuta una vez al año.
- **@monthly**: Se ejecuta una vez al mes.
- **@weekly**: Se ejecuta una vez por semana.
- **@daily**: Se ejecuta una vez al día.
- **@hourly**: Se ejecuta una vez por hora.

Entorno de ejecución de las tareas

- Cada tarea se ejecuta por una *shell* `/bin/sh`.
- Para cada tarea están disponibles algunas variables de entorno, como `SHELL`, `LOGNAME` y `HOME`.
- La entrada estándar de cada tarea se redirige de `/dev/null`, la salida estándar y la de error se envían por correo electrónico al propietario de la tarea.

cron para el administrador

- La orden `crontab` almacena la lista de tareas para cada usuario en el fichero `/var/spool/cron/crontabs/usuario`.
- `cron` examina periódicamente estos ficheros, y además el contenido del fichero `/etc/crontab`, que tiene una lista de tareas periódicas de *administración*.
- Estas tareas periódicas de administración tienen el mismo formato que la lista de tareas de usuarios, pero con un campo más: el nombre del usuario que ha de llevar a cabo la tarea. Ejemplo:

```
0 8 * * * root /etc/init.d/gdm restart
```

(reinicia el servicio “`gdm`”, como usuario `root`, todos los días a las 8:00).

- Además, lee todos los ficheros del directorio `/etc/cron.d` como extensiones del `/etc/crontab` (esto es, contienen también el campo “usuario”).

Normalmente, el contenido del fichero `/etc/crontab` es siempre el mismo, y similar a este:

```
17 * * * * root run-parts --report /etc/cron.hourly
25 6 * * * * root run-parts --report /etc/cron.daily
47 6 * * 7 * * root run-parts --report /etc/cron.weekly
52 6 1 * * * * root run-parts --report /etc/cron.monthly
```

Esto es, ejecuta con la periodicidad indicada todos los *scripts* que se encuentren en esos directorios.

Si algún paquete del sistema necesita realizar algún tipo de tarea periódica, añada *scripts* al directorio `/etc/cron.*` adecuado, o, si sus necesidades no se ajustan a ninguno de ellos, añaden un fichero con formato de `cron` al directorio `/etc/cron.d`.

anacron

Las tareas periódicas indicadas en los directorios `/etc/cron.daily`, `/etc/cron.weekly`..., suelen ser tareas rutinarias del sistema que conviene realizar de vez en cuando, como limpiar los temporales, gestionar colas de correo, realizar *back-ups*, etc.

Estas tareas se realizan en horas de poco uso de la máquina (de madrugada).

Pero es habitual que haya máquinas que no estén encendidas 24 horas al día: estaciones de trabajo que se encienden y apagan para cada uso, y no suelen estar encendidas en el momento planificado para llevar a cabo estas tareas periódicas.

anacron: Servicio que se inicia al encender la máquina, se da cuenta de qué tareas periódicas deberían haberse realizado desde la última vez que se apagó esta, y las realiza.

at

- El demonio `cron` se utiliza para realizar tareas periódicas, que se realizan en un momento determinado y se vuelven a realizar pasado un período de tiempo.
- `at` y `atd` se utilizan para ejecutar una tarea concreta *en un instante determinado*, y no volver a ejecutarla más.
- “`atd`” es un demonio, que siempre está arrancado, y se encarga de ejecutar las tareas programadas.
- “`at`” se ejecuta para añadir tareas nuevas a la lista de tareas por realizar.
- “`atq`” lista las tareas pendientes para el usuario que las ejecuta.
- “`atrm`” elimina una tarea de la lista de tareas pendientes.

Ficheros de configuración de `at`

Son similares a los de `crontab`:

- “`/etc/at.allow`”. Si existe, indica que solamente la lista de usuarios incluidos en ese fichero (un usuario por línea) tiene permiso para utilizar `at`, `atq` y `atrm`.
- “`/etc/at.deny`”. Si no existe `/etc/at.allow`, y existe este fichero, los usuarios incluidos en él no tienen permiso para utilizar `at`, `atq` y `atrm`.
- Si ninguno de los dos ficheros existe, solo el superusuario puede utilizar estas órdenes.

Normalmente, “`at.allow`” no existe, y “`at.deny`” es un fichero vacío, lo cual permite el uso de `at` para todos los usuarios del sistema.

Uso de `at`

- `at fecha/hora`. Lee órdenes de la entrada estándar, para que se realicen en la *fecha/hora* especificadas.

fecha/hora puede ser:

- `HH[:MM[am|pm] [Mes día]`
- `"now", "midnight", "noon", "teatime", "today" o "tomorrow"`.
- `fecha/hora + número (minutes, hours, days, weeks)`

Ejemplos:

- `1550 Feb 14 + 3 days`
- `10am Jul 31`
- `1am tomorrow`