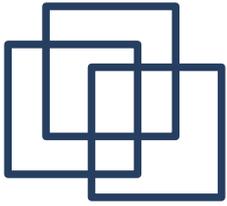


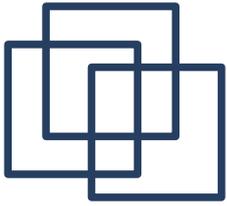
Planificación de tareas en Linux



Nota de Copyright

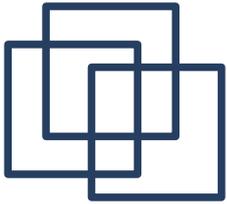
© 2006 Diego Chaparro. Algunos derechos reservados.

Este trabajo se distribuye bajo la licencia Creative Commons Attribution-ShareAlike. Para obtener la licencia completa, véase <http://creativecommons.org/licenses/by-sa/2.1/es>



Planificación de tareas

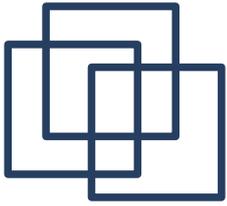
- Un administrador se encontrará con tareas repetitivas, estas tareas planificar para que se ejecuten de forma automática
- **at**
 - Planifica la ejecución de un proceso que se ejecutará una sola vez a una determinada hora
 - Sintaxis: *at [opcion] hora*
 - Opciones:
 - b Planifica un trabajo cuando la carga del sistema es baja
 - d Borra una tarea planificada. Igual a atrm
 - f fichero Lee la tarea de un fichero
 - l Muestra todos los trabajos planificados para el usuario.
 - m Manda un mail al usuario cuando termina la tarea



Planificación de tareas

- **at**

- Se usa así:
 1. Escribir **at** con las opciones y la hora que queremos
 2. Escribir el comando que queremos ejecutar
 3. Pulsar Ctrl+D
- Se pueden escribir varios comandos a ejecutar en cada tarea, y se ejecutarán uno detrás de otro, como si se escribieran en la línea de comandos con un ;
- La hora puede ser escrita en varios formatos:
 - HH:MM
 - El día de la semana por nombre o número
 - Usando palabras como: midnight, noon, tatime (4pm), today, tomorrow y now



Planificación de tareas

- **at**

- Ejemplos:

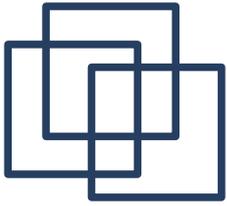
- at 23:15 -f fichero
 - at +3 hours -f fichero
 - at now sat reboot

- **atq**

- Listado de las tareas planificadas. Muestra:

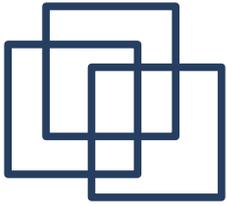
- Número de tarea, fecha, hora y clase de tarea
 - Muestra la cola en la que está la tarea. La a es en la que se ponen las tareas planificadas con at. En b las planificadas con bath. En = las tareas que se están ejecutando actualmente

- **atrm**



Planificación de tareas

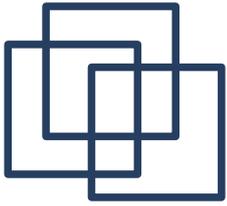
- **atrm**
 - Borra tareas planificadas
 - Ejemplo: *atrm 12 13*
- Control de tareas at
 - Las tareas consumen recursos del sistema, por tanto hay que controlar qué usuarios pueden planificar tareas
 - Si existe el fichero */etc/at.allow* y el usuario está incluido ahí entonces puede planificar, si no está no puede
 - Si no existe el fichero */etc/at.deny* y el usuario está incluido ahí entonces no puede planificar, en caso contrario si puede



Planificación de tareas

- **batch**

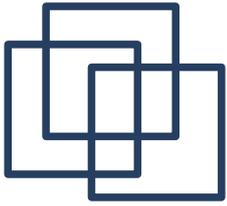
- Es lo mismo que `at -b`, es decir, planifica tareas que se ejecutan cuando la carga del sistema es baja
- Si no se especifica una hora se ejecuta en cuanto la carga del sistema es baja
- Si se especifica una hora se ejecuta después de esa hora cuando la carga del sistema es baja
- `batch` examina `/proc/loadavg` y ejecuta estas tareas cuando la carga es menor de 1.5



Planificación de tareas

- **crontab**

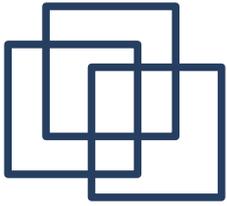
- Planifica tareas que se ejecutan recurrentemente
- Sintaxis: *crontab [opcion] [fichero] [usuario]*
- Primero que hay que haer es crear un fichero con el siguiente formato:
 - minuto 0-59
 - hora 0-23
 - día del mes 1-31
 - mes 1-12 (o nombres)
 - día de la semana 0-7 (o nombres)
 - comando a ejecutar
- Se puede especificar un * en los campos en los que el valor es indiferente para ejecutar el comando



Planificación de tareas

- **crontab**

- Se pueden especificar rangos de días y horas con un guión
- Se puede especificar varios valores separados por comas
- Ejemplo:
 - 00 1 * * 2,4fichero
- Después de crear el fichero hay que añadir esto al planificador con el comando crontab

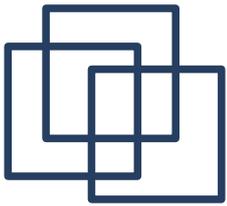


Planificación de tareas

- **crontab**

- Opciones:

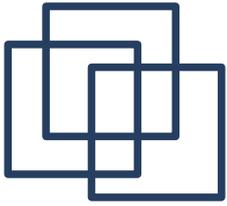
- e Crea o edita el fichero crontab del usuario
 - l Muestra el fichero de crontab de usuario
 - r Borra el fichero de crontab de usuario
 - u usuario Especifica sobre qué usuario es el fichero crontab especificado. Solo lo puede usar el root



Planificación de tareas

- **crond**

- Es el demonio que se encarga de chequear varios ficheros para ver si hay tareas a ejecutar
- El primero de los ficheros es el `/etc/crontab`, en el que se especifican ciertos directorios con las tareas que se ejecutan cada hora, día y semana:
 - `01 * * * * root run-parts /etc/cron.hourly`
 - `02 4 * * * root run-parts /etc/cron.daily`
 - `22 4 * * 0 root run-parts /etc/cron.weekly`
- `run-parts` es un script que ejecuta cada uno de los scripts en el directorio especificado
- Después `crond` mirará en `/var/spool/cron` por si hay otras tareas planificadas



Planificación de tareas

- **Manejar tareas de cron**

- Se puede especificar qué usuarios pueden planificar tareas o no
- Si existe el fichero `/etc/cron.allow`, y el usuario está en ese fichero tendrá permiso, sino no tendrá permiso
- Si no existe el fichero `/etc/cron.allow`, y en `/etc/cron.deny` aparece el usuario, no tendrá permiso, sino si tendrá permiso