

Administración y configuración de GNU/Linux – Parte I

1. Servicios del sistema	1
Inicio	1
Parada	4
2. Programación de tareas	4
Ficheros y directorios con tareas	4
El comando cron	5
3. Copias de seguridad	7
4. Rendimiento del sistema	7

1. Servicios del sistema

Los servicios del sistema son **procesos que se ejecutan en segundo plano** a la espera de ser llamados por el usuario para ofrecerle la función o utilidad para la que están programados.

Inicio

A la hora de arrancar el sistema existen unas **configuraciones de arranque** para los sistemas Linux. Son configuraciones ya preparadas, pero que pueden ser modificadas, si es necesario, por el administrador o superusuario del sistema. Esas configuraciones son los **runlevels** o **niveles de arranque o ejecución**. **Existen ocho niveles** diferentes. Cada nivel con su función son los siguientes:

Nivel	Función
0	Apagar el equipo.
1	Arrancar en modo monousuario como root.
2	Arrancar en modo multiusuario sin soporte de red.
3	Arrancar en modo multiusuario con soporte de red.
4	Arrancar en modo multiusuario con soporte de red.
5	Arrancar en modo multiusuario con soporte de red y entorno gráfico.
6	Reiniciar el equipo.
S	Modo monousuarios. Es similar al nivel 1.

Los niveles 0, 1 y 6 son niveles reservados. El resto puede ser definible y modificable por el usuario.

Estos niveles de ejecución están configurados en todas las versiones de UNIX y Linux. Hasta la versión de Ubuntu 6.06 LTS, el sistema de arranque se basaba en el sistema operativo UNIX System V, que fue una versión de UNIX desarrollada por la empresa AT&T, cuyo **sistema de arranque SysVinit, y sus scripts en el directorio /etc/init.d son y siguen siendo utilizados en muchas distribuciones de Linux para el inicio y apagado del sistema.**

Dentro de ese directorio encontramos el **fichero Inittab**, que **configura qué nivel de ejecución va a cargar que proceso o servicio, es decir, el proceso init**, que es el **primero que se carga en el inicio del sistema**, tiene definida todas las tareas que realiza en el fichero inittab.

A partir de la versión de Ubuntu 6.10 se instaló un nuevo sistema de arranque más potente, versátil, basado en eventos, llamado **Upstart**, desarrollado por **Canonical**. El proceso que se carga al inicio, **init**, **se llama igual que en el sistema anterior**. A partir de él, se cargan los demás procesos del sistema operativo y los servicios. **Se suprime el fichero inittab**. Con **Upstart**, los **servicios son iniciados y parados por eventos**, los servicios se pueden volver a iniciar si se paran de forma inesperada o por un error del sistema.

Sin embargo, este sistema de arranque, si ve que el equipo está configurado para el sistema de arranque anterior, es decir, si encuentra el fichero /etc/inittab, lo utiliza porque **mantiene la compatibilidad con el sistema de arranque anterior**, lo que va en detrimento de las ventajas y posibilidades de Upstart en favor de la compatibilidad.

Los **scripts de cada evento y cada nivel** los podemos encontrar en el **directorio /etc/event.d**, como los ficheros que llaman a las **terminales virtuales (tty1 a tty6)**, los ficheros que **ejecutan los scripts de los niveles de arranque (rc0 a rc6 y rcs)**, el fichero **rc-default** que define el **nivel de arranque por defecto** y otros ficheros, como **ctrl-alt-del** (para reiniciar el sistema), **last-good-boot** (último kernel que arrancó).

Desde el terminal, se puede reiniciar el sistema y arrancar en otro nivel usando el **comando telinit** desde la línea de comandos o **init**, que normalmente es ejecutado por el kernel, pero es necesario tener privilegios de administrador o superusuario para ejecutarlos.

Reiniciamos el sistema en el nivel de arranque 1 y, después, lo volvemos a reiniciar en el 5, pero primero entramos como root y cambiamos su contraseña.

```
paco@ubuntupaco:~$ sudo passwd root
paco@ubuntupaco:~$ su
paco@ubuntupaco:~$ telinit 1
paco@ubuntupaco:~$ telinit 5
```

Lo **normal** es que para un **usuario de escritorio se arranque en el nivel 5**, pero si queremos usar el equipo en modo texto para arrancar en modo multiusuario y con soporte de red, para que funcione, por ejemplo, como servidor lo podríamos configurar para que arrancara en el nivel 3.

Los scripts que se encargan que arrancar y parar los servicios o reiniciarlos si hemos cambiado su configuración se encuentran en el directorio **/etc/init.d**. Los parámetros que se le pueden pasar dependen del script, pero normalmente son **start, stop, restart, status**,...

Los servicios que se suelen cargar en el sistema son, entre otros, los siguientes:

- **Gestor de ventanas.**

gdm (Gnome Display Manager), si se usa Gnome o **kdm** (KDE Display Manager), si se tiene instalado el escritorio KDE. Existen otros, como E17, para el IceWM,...

- **Servicio de impresora.**
cups (Common Unix Printing System, Sistema de impresión común de Unix) es un servicio de impresora que consta de una cola de impresión, con un planificador que va mandando los trabajos a la cola.
- **Servicios de red**
NetworkManager, servicio para las conexiones de red.
wicd, ofrece el mismo servicio que el anterior, pero se suele instalar este si NetworkManager, que viene instalado por defecto en Ubuntu, da errores.
xinetd, servicio de red.
avahi-daemon, cliente para descubrir servicios de configuración de red.
- **Servicio para compartir recursos en red**
samba (smb), para compartir archivos e impresoras con un equipo que tenga instalado Window dentro de la misma red local.
- **Control de ahorro de energía (apmd, acpid).**
- **Planificación de tareas.** Ejecución de tareas en un momento determinado:
atd, es el servicio del comando **at**, que ejecuta **tareas en un instante concreto**, como puede ser una hora y una fecha especificada previamente.
crond, es el servicio del comando **cron**, que **ejecuta procesos o scripts de forma programada**, cada día, cada hora, etc. En el fichero **crontab** se especifica cuándo se ejecuta cada tarea.
anacron, programa tareas de forma parecida a cron.
- **Gestión de dispositivos (udev).** Controla los **archivos del directorio /dev**.
- **Gestión de conexiones Bluetooth (bluetooth)**
DeviceKit, que sustituye al servicio **HAL** (Hardware Abstraction Layer, Capa de abstracción de hardware). Funciona como interfaz entre el hardware y el software del sistema informático, pero DeviceKit implementa algunas mejoras, como avisar cuando el hardware es agregado o quitado del sistema.
- **Gestión de la tarjeta de sonido (alsa-utils).**
- **D-Bus** (Desktop Bus), se encarga de la comunicación entre los procesos.

En modo texto vamos al directorio donde están los scripts que sirven para iniciar o parar los servicios del sistema, /etc/init.d. Paramos los servicios cups (impresora) y NetworkManager (red) y los volvemos a poner en funcionamiento y comprobamos que antes no funcionaban tanto la impresora como la red pero ahora sí.

```
paco@ubuntupaco:~$ cd /etc/init.d
paco@ubuntupaco:~$ sudo ./cups stop
paco@ubuntupaco:~$ sudo ./NetworkManager stop
```

Comprobamos que tanto la impresora como la red están desactivadas, es decir, no funcionan.

```
paco@ubuntupaco:~$ sudo ./cups start
paco@ubuntupaco:~$ sudo ./NetworkManager start
```

Parada

Los servicios usados en la parada del sistema son:

- **acpi y acpid**

Para apagar el ordenador sin que cause ningún problema. Para parar o reiniciar el equipo tendremos que pulsar el botón de apagado desde el entorno gráfico, o bien mediante comandos. Los comandos de parada o reinicio del sistema, que deben ser ejecutados por el superusuario o por un usuario que pueda tener sus privilegios.

halt y poweroff

Paran el equipo.

reboot

Reinicia el sistema.

shutdown

Para o reinicia el sistema.

Sintaxis:

shutdown [opciones] tiempo [mensaje]

Opciones:

-h → suspende el sistema.

-r → reinicia el sistema.

Argumentos:

tiempo

Si el tiempo es **now** se ejecuta inmediatamente. Si queremos que se apague en otro instante de tiempo, en vez de now, especificaremos **una hora concreta o el número de minutos que queremos que pasen** cuando se apague el sistema.

mensaje

Muestra el mensaje que se le pasa como argumento en la **pantalla**. Este comando lo puede usar un usuario normal si su nombre está en el fichero **/etc/shutdown.allow**.

2. Programación de tareas

Programar una tarea es planificar que dicha tarea se ejecute en un instante determinado o cada cierto tiempo de forma periódica. Para ello se pueden usar los siguientes comandos:

- **at** → permite planificar la ejecución de tareas.
- **cron, anacron, crontab** → permiten planificar órdenes o procesos de forma periódica en el tiempo o ver los procesos planificados.

Ficheros y directorios con tareas

/etc/at.allow y **/etc/at.deny**

Contienen los nombres de los usuarios a los que les está permitido o no usar el comando **at**.

/etc/cron.allow y /etc/cron.deny

Usuarios a los que les está permitido o no para usar **crontab**. Si el fichero **cron.allow** no existe, el usuario puede usar crontab a menos que su nombre esté en **cron.deny**.

Si se quiere denegar el uso a todos, se escribe **ALL en cron.deny**. Si no existe ninguno de los dos ficheros, el uso está permitido a todos los usuarios. Si cron.allow existe, todo el que no aparezca tendrá el uso de cron prohibido.

Los ficheros at.allow, at.deny, cron.allow y cron.deny no tienen que existir obligatoriamente, pero si existen, los comandos at y cron se basarán en ellos para ver a qué usuarios les está permitido o no usar dichos comandos.

/etc/crontab

Contiene las tareas programadas del administrador.

Los archivos que contienen las tareas programadas de cada usuario están en los directorios:

/var/spool/cron/atjobs
/var/spool/cron/atspool
/var/spool/cron/crontabs

Los directorios con las tareas programadas cada hora, cada día, semanalmente o mensualmente, son, respectivamente: **/etc/cron.hourly**, **/etc/cron.daily**, **/etc/cron.weekly** y **/etc/cron.monthly**.

El comando cron

Es el **programa que permite ejecutar automáticamente comandos o scripts a una hora o fecha específica**. Es usado normalmente para tareas administrativas, pero puede ser usado para ejecutar cualquier otro programa. Un servicio, como cron, una vez iniciado, normalmente de forma automática al arrancar el sistema operativo, ya no es necesario volverlo a iniciar.

Vamos a comprobar si está iniciado el servicio cron.

```
paco@ubuntupaco:~$ ps -ely | grep cron
paco@ubuntupaco:~$ ps -aF | grep cron
```

El comando cron se puede usar almacenando un script en uno de los directorios cron.hourly, cron.daily, cron.weekly y cron.monthly, en /etc. Dependiendo de dónde lo guardemos, se ejecutará cada hora, cada día cada semana o cada mes, respectivamente.

Si miramos directamente el **fichero /etc/crontab**, veremos que las cuatro primeras líneas son las variables:

- **SHELL**, indica bajo qué shell se va a ejecutar el cron, que por defecto es el del usuario.
- **PATH**, especifica los directorios en los que cron buscará el comando a ejecutar, y no tiene que coincidir con el PATH del sistema.
- **MAIL TO** es a quien se le envía la salida del comando si la tuviera. Si no se especifica, entonces cron enviará el correo al usuario propietario del comando que se ejecuta.
- **HOME**, que contiene el directorio principal del comando cron, que por defecto será el directorio personal del usuario.

Tras las variables, están las líneas con las tareas programadas. **Cada línea consta de 7 campos**, cada uno de los cuales indica:

Campo	Función	Valores
Minuto	Minuto en que se ejecutará el comando.	0-59
Hora	Hora en que se ejecutará el comando.	0-23
Día	Día del mes en que se va a ejecutar.	1-31
Mes	Mes en que se ejecutará el comando.	1-12
Día de la semana	Día de la semana en que se ejecutará el comando.	0-6
Usuario	Usuario que ejecuta el comando.	Nombre del usuario
Comando	Comando o script a ejecutar.	Comando o script

Si en los primeros **cinco campos aparece el carácter *** indicará **todos los valores**. También se pueden especificar **lista**, **rangos** de valores e **incrementos**. El programa cron se invoca cada minuto y ejecuta las tareas que sus campos se cumplan en ese preciso minuto.



crontab

Se utiliza para configurar los ficheros crontab de cada usuario.

Sintaxis1:

crontab [opciones]

Sintaxis2:

crontab fichero

Opciones:

-e → edita o crea el fichero crontab.

-l → lista el contenido del fichero.

-r → elimina el fichero contrab del usuario.

Argumento:

fichero

Convierte el fichero que se le indique en el argumento, como fichero crontab del usuario.

El fichero **/etc/crontab** es el **archivo crontab de root**, pero se le podría añadir una tarea a otro usuario, indicándolo en el **sexto campo del fichero crontab**. Algunas veces, **después de modificar** el fichero crontab habrá que **reiniciar el servicio cron**, para que los cambios surtan efecto.

3. Copias de seguridad

Las copias de seguridad son una **tarea importante que hay que realizar con planificación y regularidad** para proteger los datos que utilizamos de una posible pérdida ya sea por un fallo o error en el sistema o bien por un borrado accidental. Las copias de seguridad **se deben hacer periódicamente** y para evitar el olvido es **aconsejable que se programen** para que se ejecuten cada cierto tiempo.

Las copias de seguridad se utilizan **para recuperar los datos en el caso de que ocurra una pérdida de los datos**, como puede ser un fallo del soporte, como que el disco duro se dañe, que un programa como un virus elimine los datos, o que se eliminen por error. Antes de realizar la copia es necesario decidir **ciertas cuestiones**, como el **soporte** donde se va a almacenar la copia de seguridad, es decir, si va a almacenarse en una cinta magnética, en otro disco duro, ya sea interno o extraíble vía USB, o en una memoria flash o pendrive, dependiendo de la cantidad de información a almacenar.

Otra cuestión es el **tipo** de copia de seguridad a realizar. Dependiendo de la cantidad de archivos que guardemos en la copia, podemos hablar de diferentes tipos de copias de seguridad. Podemos hablar de tres tipos de copia de seguridad:

- **Total o íntegra.** Copia de todos los archivos y directorios que se le indiquen.
- **Incremental.** Copia solo los **archivos que se hayan modificado desde la última copia de seguridad**, ya sea incremental o total. Para restaurar el sistema de archivos tendríamos que recurrir a la última copia total más las copias incrementales que se hayan ido realizando con posterioridad.
- **Diferencial.** Copia los **archivos que se hayan modificado desde la última copia de seguridad total**. Copia más archivos que la incremental, por lo que **tarda más tiempo en realizarse y ocupa más espacio**, pero a cambio ocupa menos espacio que una copia total, y a la hora de restaurar el sistema de archivos solo necesitaremos utilizar la última copia incremental y la última total, por lo que el proceso de restauración es más rápido.

Si la cantidad de información a salvar no es excesivamente grande podemos realizar copias totales, lo que facilitaría la labor de recuperar el sistema. Si la cantidad de información es muy alta, pero se modifica poco, es mejor realizar copias diferenciales.

Otro punto importante es la **elección de la información a salvar**. La información de cada usuario es muy importante, luego habrá que realizar una copia de seguridad de las **carpetas personales de los usuarios** (/home), también sería bueno salvar algunos archivos de configuración (/etc) y la carpeta donde se almacenan las incidencias del sistema (/var/log), por si ya restaurado podemos descubrir qué causó el fallo.

Otras medidas a tener en cuenta a la hora de realizar las copias de seguridad es comprimir la copia, para que ocupe menos espacio, y automatizar el proceso para que no se olvide en ningún momento su realización.

4. Rendimiento del sistema

Para que el sistema funcione de forma más óptima, será necesario tomar ciertas medidas para mejorar su rendimiento, como pueden ser:

- **Desinstalar paquetes no usados** para liberar disco duro.
- **Desinstalar servicios que no utilizemos**, para no consumir recursos de la memoria principal y del procesador.

Para ver si el rendimiento de ciertos elementos del sistema son buenos, lo podremos observar mediante el monitor del sistema. En la pestaña de Recurso podremos ver si el uso de la CPU, de la memoria y de la red es eficiente o se está colapsando algún recurso.

Carga promedio del sistema

La carga promedio del sistema (load average) se puede definir como la **media de procesos compitiendo por el uso del procesador durante un tiempo determinado**, que en Linux son 1, 5 y 15 minutos.

La carga promedio mínima de un sistema es de 0 y la máxima es ilimitada, aunque no suele ser superior a 10. Lo normal es que sea de 1 o 2, pero lo ideal es que sea menor a 1. El comando **top** muestra al final de la primera línea los tres números de la carga promedio.

Miramos la carga promedio del sistema

```
paco@ubuntupaco:~$ top
```

Podemos ver el resultado, junto a **load average**, y como podemos ver, **los tres números de carga promedio son menores a 1**, que es el valor **ideal**.

```

administrador@ubuntu: ~
File Edit View Search Terminal Help
top - 09:05:36 up 4 min, 2 users, load average: 0.12, 0.23, 0.12
Tasks: 138 total, 1 running, 136 sleeping, 0 stopped, 1 zombie
Cpu(s): 4.6%us, 4.6%sy, 0.0%ni, 90.7%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
Mem: 1025224k total, 435692k used, 589532k free, 41812k buffers
Swap: 1046524k total, 0k used, 1046524k free, 221320k cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM   TIME+  COMMAND
 1225 root        20   0 52888  24m 6372  S   6.3   2.4   0:04.66 Xorg
 1997 administ  20   0 90824  13m 10m   S   1.7   1.4   0:00.56 gnome-terminal
 1820 administ  20   0 74588  18m 14m   S   0.7   1.8   0:01.93 vmware-user-loa
 2083 root        20   0 2632  1144 860   R   0.7   0.1   0:00.06 top
   256 root        20   0     0     0     0   S   0.3   0.0   0:00.12 jbd2/sda1-8
 1793 administ  20   0 102m  17m 12m   S   0.3   1.7   0:00.56 nautilus
 1802 administ  20   0 107m  14m 11m   S   0.3   1.5   0:00.68 gnome-panel
 1873 administ  20   0 81596  11m 9644   S   0.3   1.2   0:00.20 wnck-applet
 1974 administ  20   0 80860  11m 9576   S   0.3   1.2   0:00.17 update-notifier
     1 root        20   0 3052  1820 1248   S   0.0   0.2   0:01.90 init
     2 root        20   0     0     0     0   S   0.0   0.0   0:00.00 kthreadd
     3 root        20   0     0     0     0   S   0.0   0.0   0:00.03 ksoftirqd/0
     4 root        20   0     0     0     0   S   0.0   0.0   0:00.20 kworker/0:0
     5 root        20   0     0     0     0   S   0.0   0.0   0:00.55 kworker/u:0
     6 root        RT   0     0     0     0   S   0.0   0.0   0:00.00 migration/0
     7 root         0 -20     0     0     0   S   0.0   0.0   0:00.00 cpuset
     8 root         0 -20     0     0     0   S   0.0   0.0   0:00.00 khelper

```

Visor de sucesos

Es una aplicación que se puede utilizar para realizar tareas administrativas, como evaluar el rendimiento del sistema y hacer un diagnóstico del mismo. Muestra información sobre los servicios del sistema, la interacción entre los servicios y las aplicaciones y, en general, sobre el rendimiento del equipo.