

Arranque y detención de sistemas GNU/Linux

Manolo Padrón Martínez (manolopm@cip.es)
Imobach González Sosa (imobachgs@softhome.net)

Ampliación de Sistemas Operativos

1 Introducción

2 El proceso de arranque

- Visión general
- Gestores de arranque
- El núcleo en el arranque
- El proceso `init` y los niveles de ejecución

3 Detención del sistema

- Visión general
- `shutdown`
- `halt`, `poweroff` y `reboot`

1 Introducción

2 El proceso de arranque

- Visión general
- Gestores de arranque
- El núcleo en el arranque
- El proceso `init` y los niveles de ejecución

3 Detención del sistema

- Visión general
- `shutdown`
- `halt`, `poweroff` y `reboot`

1 Introducción

2 El proceso de arranque

- Visión general
- Gestores de arranque
- El núcleo en el arranque
- El proceso `init` y los niveles de ejecución

3 Detención del sistema

- Visión general
- `shutdown`
- `halt`, `poweroff` y `reboot`

Características

Los sistemas GNU/Linux toman de los UNIX gran parte de los conceptos que aplican en el arranque.

Características

- Potencia
- Transparencia
- Flexibilidad

Pasos del proceso de arranque

- 1 El *firmware* cede el control al gestor de arranque.
- 2 El gestor de arranque carga en memoria el núcleo.
- 3 El núcleo:
 - 1 Inicializa dispositivos.
 - 2 Monta el disco RAM y la partición raíz.
 - 3 Lanza a `init`.
- 4 `init` realiza la configuración de software:
 - 1 Monta el sistema de ficheros raíz como lectura/escritura.
 - 2 Inicializa algunos subsistemas.
 - 3 Decide que servicios arrancar.

Pasos del proceso de arranque

- 1 El *firmware* cede el control al gestor de arranque.
- 2 El gestor de arranque carga en memoria el núcleo.
- 3 El núcleo:
 - 1 Inicializa dispositivos.
 - 2 Monta el disco RAM y la partición raíz.
 - 3 Lanza a `init`.
- 4 `init` realiza la configuración de software:
 - 1 Monta el sistema de ficheros raíz como lectura/escritura.
 - 2 Inicializa algunos subsistemas.
 - 3 Decide que servicios arrancar.

Pasos del proceso de arranque

- 1 El *firmware* cede el control al gestor de arranque.
- 2 El gestor de arranque carga en memoria el núcleo.
- 3 El núcleo:
 - 1 Inicializa dispositivos.
 - 2 Monta el disco RAM y la partición raíz.
 - 3 Lanza a `init`.
- 4 `init` realiza la configuración de software:
 - 1 Monta el sistema de ficheros raíz como lectura/escritura.
 - 2 Inicializa algunos subsistemas.
 - 3 Decide que servicios arrancar.

Pasos del proceso de arranque

- 1 El *firmware* cede el control al gestor de arranque.
- 2 El gestor de arranque carga en memoria el núcleo.
- 3 El núcleo:
 - 1 Inicializa dispositivos.
 - 2 Monta el disco RAM y la partición raíz.
 - 3 Lanza a `init`.
- 4 `init` realiza la configuración de software:
 - 1 Monta el sistema de ficheros raíz como lectura/escritura.
 - 2 Inicializa algunos subsistemas.
 - 3 Decide que servicios arrancar.

Definición

Básicamente, un gestor de arranque es un pequeño programa que se encarga de colocar el núcleo de Linux en memoria para que comience la ejecución.

Generalmente, se encuentran divididos en fases para evitar limitaciones de espacio.

Gestores según arquitectura

- x86-32 → LILO y Grub
- x86-64 → ELILO
- Alpha, PowerPC → `aboot`
- PowerPC → `yaboot`

Definición

Básicamente, un gestor de arranque es un pequeño programa que se encarga de colocar el núcleo de Linux en memoria para que comience la ejecución.

Generalmente, se encuentran divididos en fases para evitar limitaciones de espacio.

Gestores según arquitectura

- x86-32 → LILO y Grub
- x86-64 → ELILO
- Alpha, PowerPC → aboot
- PowerPC → yaboot

Definición

Básicamente, un gestor de arranque es un pequeño programa que se encarga de colocar el núcleo de Linux en memoria para que comience la ejecución.

Generalmente, se encuentran divididos en fases para evitar limitaciones de espacio.

Gestores según arquitectura

- x86-32 → LILO y Grub
- x86-64 → ELILO
- Alpha, PowerPC → aboot
- PowerPC → yaboot

LILO :: Fases

Fase 1

- Carga la segunda fase.
- Ocupa el primer sector.

Fase 2

- Carga el núcleo o transfiere el control.
- Ofrece una interfaz de usuario.
- Tiene una menor limitación de espacio.

LILO :: Fases

Fase 1

- Carga la segunda fase.
- Ocupa el primer sector.

Fase 2

- Carga el núcleo o transfiere el control.
- Ofrece una interfaz de usuario.
- Tiene una menor limitación de espacio.

LILO :: Fundamentos de instalación

LILO puede instalarse:

- En el Master Boot Record.
- En el sector de arranque de una partición.

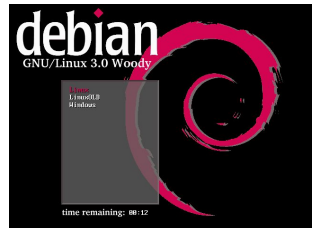
| |
|-----------------------------------|
| MBR |
| Sector de arranque Partición 1 |
| Sector de arranque Partición 2 |
| Sector de arranque Partición 3 |

LILO :: Instalación

LILO se instala simplemente ejecutando `lilo`.

Parámetros usuales

- C fichero Fichero de configuración.
- v Aumenta los detalles de la salida.
- t Hace una prueba.
- b dispositivo Dispositivo *boot*.
- r directorio Indica dónde montar `/`.



LILO :: Configuración

/etc/lilo.conf

```
# man 5 lilo.conf
boot=/dev/hda
root=/dev/hda3
default=Linux
timeout=300

image=/vmlinuz
    label=Linux

image=/boot/memtest86.bin
    label=memtest

other=/dev/hdb1
    label=WinXP
```

Grub :: Fases

Fase 1

- Carga la fase 1.5 ó la 2.
- Ocupa el primer sector.

Fase 1.5

- Si la fase 2 es muy grande, esta fase se ocupa de cargarla.
- Es la única fase que entiende sistemas de ficheros.

Fase 2

- Carga el núcleo o transfiere el control.
- Ofrece una interfaz de usuario avanzada.
- Ya no tiene limitación de espacio ocupado.

Grub :: Fases

Fase 1

- Carga la fase 1.5 ó la 2.
- Ocupa el primer sector.

Fase 1.5

- Si la fase 2 es muy grande, esta fase se ocupa de cargarla.
- Es la única fase que entiende sistemas de ficheros.

Fase 2

- Carga el núcleo o transfiere el control.
- Ofrece una interfaz de usuario avanzada.
- Ya no tiene limitación de espacio ocupado.

Grub :: Fases

Fase 1

- Carga la fase 1.5 ó la 2.
- Ocupa el primer sector.

Fase 1.5

- Si la fase 2 es muy grande, esta fase se ocupa de cargarla.
- Es la única fase que entiende sistemas de ficheros.

Fase 2

- Carga el núcleo o transfiere el control.
- Ofrece una interfaz de usuario avanzada.
- Ya no tiene limitación de espacio ocupado.

Grub :: Nomenclatura

- Los discos se nombran como `hdX`, donde `X` es su número, empezando en 0.
- La partición se especifica añadiendo su número, también partiendo de 0.

Ejemplos

- `hda` → `(hd0)`
- `hdb3` → `(hd1,2)`
- `sda` → `(hd0)`



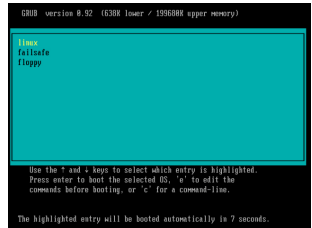
Grub :: Instalación

Hay dos formas de instalar *Grub*:

Automática Usando el script `grub-install`

Manual

- ① Crear disco de arranque copiando las fases 1 y 2 con `dd`.
- ② Arrancar con el disquete.
- ③ `root (hdX,Y)`.
- ④ `setup (hdZ)`.



Grub :: Configuración

/boot/grub/grub.conf

```
# man 5 grub.conf
default 0
timeout 30

title Linux
kernel (hd0,2)/vmlinuz root=/dev/hda3

title memtest
root (hd0,1)
kernel (hd0,1)/boot/memtest/memtest.bin

title WinXP
root (hd1,0)
makeactive
chainloader +1
```

Pasos

- 1 El núcleo se autodescomprime.
- 2 Inicializa dispositivos.
- 3 Monta el disco RAM como raíz si procede.
- 4 Monta la partición raíz como sólo lectura.
- 5 Llama a `init`:
 - 6 `/sbin/init`
 - 7 `/etc/init`
 - 8 `/bin/init`
 - 9 `/bin/sh`

Pasos

- 1 El núcleo se autodescomprime.
- 2 Inicializa dispositivos.
- 3 Monta el disco RAM como raíz si procede.
- 4 Monta la partición raíz como sólo lectura.
- 5 Llama a `init`:
 - 1 `/sbin/init`
 - 2 `/etc/init`
 - 3 `/bin/init`
 - 4 `/bin/sh`

Pasos

- 1 El núcleo se autodescomprime.
- 2 Inicializa dispositivos.
- 3 Monta el disco RAM como raíz si procede.
- 4 Monta la partición raíz como sólo lectura.
- 5 Llama a `init`:
 - 1 `/sbin/init`
 - 2 `/etc/init`
 - 3 `/bin/init`
 - 4 `/bin/sh`

Pasos

- 1 El núcleo se autodescomprime.
- 2 Inicializa dispositivos.
- 3 Monta el disco RAM como raíz si procede.
- 4 Monta la partición raíz como sólo lectura.
- 5 Llama a `init`:
 - 1 `/sbin/init`
 - 2 `/etc/init`
 - 3 `/bin/init`
 - 4 `/bin/sh`

Pasos

- 1 El núcleo se autodescomprime.
- 2 Inicializa dispositivos.
- 3 Monta el disco RAM como raíz si procede.
- 4 Monta la partición raíz como sólo lectura.
- 5 Llama a `init`:
 - 1 `/sbin/init`
 - 2 `/etc/init`
 - 3 `/bin/init`
 - 4 `/bin/sh`

Contenidos

- Los niveles de ejecución
- Presentación de `init`
- El papel de `init` en el arranque
- `/etc/inittab`
- El `inittab` de *Red Hat Linux*

Contenidos

- Los niveles de ejecución
- Presentación de `init`
- El papel de `init` en el arranque
- `/etc/inittab`
- El `inittab` de *Red Hat Linux*

Contenidos

- Los niveles de ejecución
- Presentación de `init`
- El papel de `init` en el arranque
- `/etc/inittab`
- El `inittab` de *Red Hat Linux*

Contenidos

- Los niveles de ejecución
- Presentación de `init`
- El papel de `init` en el arranque
- `/etc/inittab`
- El `inittab` de *Red Hat Linux*

Contenidos

- Los niveles de ejecución
- Presentación de `init`
- El papel de `init` en el arranque
- `/etc/inittab`
- El `inittab` de *Red Hat Linux*

Contenidos

- Los niveles de ejecución
- Presentación de `init`
- El papel de `init` en el arranque
- `/etc/inittab`
- El `inittab` de *Red Hat Linux*

Los niveles de ejecución :: Concepto

Definiciones

Configuración de software o “modo de funcionamiento” del sistema

Niveles usuales

0-6 Usados comúnmente. 0, 1 y 6 reservados.

7-9 Válidos, aunque no documentados y poco usados.

Linux Standard Base

| Nivel | Uso |
|-------|---|
| 0 | Apagado |
| 1 | Modo monousuario |
| 2 | Multiusuario sin NFS |
| 3 | Multiusuario completo |
| 4 | Reservado para uso local |
| 5 | Multiusuario con <code>xdm</code> o equivalente |
| 6 | Reinicio |

Los niveles de ejecución :: Concepto

Definiciones

Configuración de software o “modo de funcionamiento” del sistema

Niveles usuales

0-6 Usados comúnmente. 0, 1 y 6 reservados.

7-9 Válidos, aunque no documentados y poco usados.

Linux Standard Base

| Nivel | Uso |
|-------|---|
| 0 | Apagado |
| 1 | Modo monousuario |
| 2 | Multiusuario sin NFS |
| 3 | Multiusuario completo |
| 4 | Reservado para uso local |
| 5 | Multiusuario con <code>xdm</code> o equivalente |
| 6 | Reinicio |

Los niveles de ejecución :: Representación

```
/etc/rc.d/init.d/
```

Por cada servicio, existe un script que ofrece las siguientes acciones:

- start
- stop
- restart
- try-restart
- reload
- force-reload
- status

Ejemplos en Red Hat Linux

```
/etc/rc.d/init.d/apache reload  
/etc/rc.d/init.d/postfix stop  
/etc/rc.d/init.d/mysql restart
```

Ejemplos según el LSB

```
/etc/init.d/apache reload  
/etc/init.d/postfix stop  
/etc/init.d/mysql restart
```

Los niveles de ejecución :: Representación

```
/etc/rc.d/init.d/
```

Por cada servicio, existe un script que ofrece las siguientes acciones:

- start
- stop
- restart
- try-restart
- reload
- force-reload
- status

Ejemplos en Red Hat Linux

```
/etc/rc.d/init.d/apache reload  
/etc/rc.d/init.d/postfix stop  
/etc/rc.d/init.d/mysql restart
```

Ejemplos según el LSB

```
/etc/init.d/apache reload  
/etc/init.d/postfix stop  
/etc/init.d/mysql restart
```

Los niveles de ejecución :: Representación

`/etc/rc.d/rcN.d/`

- Los niveles de ejecución se representan mediante directorios de nombre `/etc/rc.d/rcN.d`, donde `N` es su identificador.
- Esos directorios contienen enlaces a los scripts de `/etc/rc.d/init.d/`.
- Sus nombres son del tipo `S99postfix` o `K99apache` y denotan qué acción se debe llevar a cabo:
 - `S/K` Iniciar/detener el servicio.
 - `número` Orden de secuencia.
 - `nombre` Nombre del servicio.

Los niveles de ejecución :: Representación

Ejemplo

```
K99xdm -> ../init.d/xdm
S10sysklogd -> ../init.d/sysklogd
S11klogd -> ../init.d/klogd
S20cupsys -> ../init.d/cupsys
S20inetd -> ../init.d/inetd
S20makedev -> ../init.d/makedev
S20mysql -> ../init.d/mysql
S20postfix -> ../init.d/postfix
S20ssh -> ../init.d/ssh
S89atd -> ../init.d/atd
S89cron -> ../init.d/cron
S91apache2 -> ../init.d/apache2
S99rmnologin -> ../init.d/rmnologin
S99stop-bootlogd -> ../init.d/stop-bootlogd
```

Presentación de `init`

Definición

El proceso `init` es el antecesor de todos los procesos del sistema.

Características

- Su *Process Identifier* (PID) es 1
- Inmune a `SIGKILL`.
- Nivel de usuario.
- Enfoque a la System V en la mayoría de distribuciones (frente a BSD).

Ejemplo

```
$ pstree -p
init(1)---aio/0(9)
    |-bash(342)---pstree(413)
    |-bash(343)---vi(372)
    |-events/0(3)
    |-getty(344)
    |-getty(345)
    |-getty(346)
    |-getty(347)
    |-kblockd/0(4)
    |-khubd(5)
    |-kjournald(12)
    |-klogd(279)
    |-kseriod(11)
    |-ksoftirqd/0(2)
    |-kswapd0(8)
    |-pdflush(6)
    |-pdflush(7)
    '-syslogd(276)
```

Presentación de `init`

Definición

El proceso `init` es el antecesor de todos los procesos del sistema.

Características

- Su *Process Identifier* (PID) es 1
- Inmune a `SIGKILL`.
- Nivel de usuario.
- Enfoque a la System V en la mayoría de distribuciones (frente a BSD).

Ejemplo

```
$ pstree -p
init(1)---aio/0(9)
    |--bash(342)---pstree(413)
    |--bash(343)---vi(372)
    |--events/0(3)
    |--getty(344)
    |--getty(345)
    |--getty(346)
    |--getty(347)
    |--kblockd/0(4)
    |--khubd(5)
    |--kjournald(12)
    |--klogd(279)
    |--kseriod(11)
    |--ksoftirqd/0(2)
    |--kswapd(8)
    |--pdflush(6)
    |--pdflush(7)
    `--syslogd(276)
```

Presentación de `init`

Definición

El proceso `init` es el antecesor de todos los procesos del sistema.

Características

- Su *Process Identifier* (PID) es 1
- Inmune a `SIGKILL`.
- Nivel de usuario.
- Enfoque a la System V en la mayoría de distribuciones (frente a BSD).

Ejemplo

```
$ pstree -p
init(1)---aio/0(9)
    |-bash(342)---pstree(413)
    |-bash(343)---vi(372)
    |-events/0(3)
    |-getty(344)
    |-getty(345)
    |-getty(346)
    |-getty(347)
    |-kblockd/0(4)
    |-khubd(5)
    |-kjournald(12)
    |-klogd(279)
    |-kseriod(11)
    |-ksoftirqd/0(2)
    |-kswapd(8)
    |-pdflush(6)
    |-pdflush(7)
    `--syslogd(276)
```

Presentación de `init`

`telinit`

- `telinit` permite cambiar el nivel de ejecución “en caliente”.
- Es un enlace simbólico a `init`.
- Basta con teclear `telinit` seguido del nivel de ejecución.

Ejemplo

```
telinit 1
```

Presentación de `init`

`telinit`

- `telinit` permite cambiar el nivel de ejecución “en caliente”.
- Es un enlace simbólico a `init`.
- Basta con teclear `telinit` seguido del nivel de ejecución.

Ejemplo

```
telinit 1
```

El papel de `init` en el arranque

- 1 Ejecuta `rc.sysinit`
 - Desmonta `initrd`; monta `/proc` y `/sys`.
 - Configurar parámetros del kernel (`sysctl.conf`).
 - Establece nombre de la máquina.
 - “Re-monta” la partición raíz ahora como lectura/escritura.
 - Activa las particiones de `swap`.
 - Monta sistemas de archivos no montados aún (no vía red).
 - Comprueba cuotas.
 - Configura puertos serie.
 - Cargar módulos.
- 2 Invoca a `/etc/rc.d/rc` para que lance los scripts del nivel de ejecución correspondiente.
- 3 Hace un “fork” por cada terminal virtual. Por defecto,
 - seis para los niveles 2-5.
 - uno para el nivel 1.

El papel de `init` en el arranque

- 1 Ejecuta `rc.sysinit`
 - Desmonta `initrd`; monta `/proc` y `/sys`.
 - Configurar parámetros del kernel (`sysctl.conf`).
 - Establece nombre de la máquina.
 - “Re-monta” la partición raíz ahora como lectura/escritura.
 - Activa las particiones de `swap`.
 - Monta sistemas de archivos no montados aún (no vía red).
 - Comprueba cuotas.
 - Configura puertos serie.
 - Cargar módulos.
- 2 Invoca a `/etc/rc.d/rc` para que lance los scripts del nivel de ejecución correspondiente.
- 3 Hace un “fork” por cada terminal virtual. Por defecto,
 - seis para los niveles 2-5.
 - uno para el nivel 1.

El papel de `init` en el arranque

- 1 Ejecuta `rc.sysinit`
 - Desmonta `initrd`; monta `/proc` y `/sys`.
 - Configurar parámetros del kernel (`sysctl.conf`).
 - Establece nombre de la máquina.
 - “Re-monta” la partición raíz ahora como lectura/escritura.
 - Activa las particiones de `swap`.
 - Monta sistemas de archivos no montados aún (no vía red).
 - Comprueba cuotas.
 - Configura puertos serie.
 - Cargar módulos.
- 2 Invoca a `/etc/rc.d/rc` para que lance los scripts del nivel de ejecución correspondiente.
- 3 Hace un “fork” por cada terminal virtual. Por defecto,
 - seis para los niveles 2-5.
 - uno para el nivel 1.

El papel de `init` en el arranque

- 1 Ejecuta `rc.sysinit`
 - Desmonta `initrd`; monta `/proc` y `/sys`.
 - Configurar parámetros del kernel (`sysctl.conf`).
 - Establece nombre de la máquina.
 - “Re-monta” la partición raíz ahora como lectura/escritura.
 - Activa las particiones de `swap`.
 - Monta sistemas de archivos no montados aún (no vía red).
 - Comprueba cuotas.
 - Configura puertos serie.
 - Cargar módulos.
- 2 Invoca a `/etc/rc.d/rc` para que lance los scripts del nivel de ejecución correspondiente.
- 3 Hace un “fork” por cada terminal virtual. Por defecto,
 - seis para los niveles 2-5.
 - uno para el nivel 1.

/etc/inittab

¿Qué es?

- `inittab` es el fichero de configuración de `init`.
- Indica procesos que se lanzan durante el arranque y durante la operación “normal”.

`id:runlevels:acción:proceso`

`id` Secuencia única de entre uno y cuatro caracteres que identifica unívocamente la entrada.

`runlevels` Niveles de ejecución en los que se aplica la entrada.

`acción` Acción a realizar con el proceso.

`proceso` Proceso a lanzar.

/etc/inittab

¿Qué es?

- `inittab` es el fichero de configuración de `init`.
- Indica procesos que se lanzan durante el arranque y durante la operación “normal”.

`id:runlevels:acción:proceso`

`id` Secuencia única de entre uno y cuatro caracteres que identifica unívocamente la entrada.

`runlevels` Niveles de ejecución en los que se aplica la entrada.

`acción` Acción a realizar con el proceso.

`proceso` Proceso a lanzar.

/etc/inittab

Acciones (I)

`respawn` El proceso se lanza cada vez que termina. Usado por `getty`.

`wait` El proceso se inicia cuando se entra en el nivel de ejecución indicado e `init` espera a que termine.

`boot` El proceso se ejecuta durante el inicio del sistema.

`bootwait` Igual que `boot`, pero `init` espera por su finalización.

`sysinit` El proceso se ejecuta durante el inicio, antes que entradas `bootwait` y `boot`.

/etc/inittab

Acciones (y II)

`initdefault` Especifica el nivel de ejecución por defecto.

`ctrlaltdel` El proceso se ejecuta cuando `init` reciba una señal `SIGINT`.

`powerwait` El proceso se ejecuta cuando el sistema se queda sin energía. `init` espera a que termine su ejecución. Obviamente, necesita un SAI.

`powerfail` Igual que `powerwait`, pero sin esperar a que termine.

`powerokwait` Se ejecuta cuando se informa a `init` de que el suministro se ha reestablecido.

El inittab de *Red Hat Linux*

```
# Nivel de ejecución por defecto.  
id:5:initdefault:  
  
# Scripts de cada nivel de ejecución  
10:0:wait:/etc/rc.d/rc 0  
11:1:wait:/etc/rc.d/rc 1  
12:2:wait:/etc/rc.d/rc 2  
13:3:wait:/etc/rc.d/rc 3  
14:4:wait:/etc/rc.d/rc 4  
15:5:wait:/etc/rc.d/rc 5  
16:6:wait:/etc/rc.d/rc 6
```

El `inittab` de *Red Hat Linux*

```
# ctrl+alt+del
# Qué hacer cuando recibe señal SIGINT
ca::ctrlaltdel:/sbin/shutdown -t3 -r now

# Energía (corte y restitución)
pf::powerfail:/sbin/shutdown -f -h +2 "Power Failure; System Shutting Down"
pr:12345:powerokwait:/sbin/shutdown -c "Power Restored; Shutdown Cancelled"

# Consolas virtuales
# init hace un 'fork' para cada una
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6

# Ejecución del 'login' gráfico
x:5:respawn:/etc/X11/prefdm -nodaemon
```

Detención del sistema :: Herramientas

Importancia

La detención de un sistema es tan importante como su inicio.

Herramientas

Existen cuatro herramientas vinculadas a la detención, apagado y reinicio del sistema:

- shutdown
- halt
- poweroff
- reboot

Detención del sistema :: Herramientas

Importancia

La detención de un sistema es tan importante como su inicio.

Herramientas

Existen cuatro herramientas vinculadas a la detención, apagado y reinicio del sistema:

- shutdown
- halt
- poweroff
- reboot

shutdown :: Fundamentos

¿Para qué sirve?

`shutdown` permite detener, reiniciar o apagar el sistema de forma segura y ordenada.

Cambio de nivel de ejecución

Se basa en el concepto de nivel de ejecución, enviando una señal `SIGINT` a `init` para que cambie a uno de éstos:

- 0 Apagar (opción `-h`).
- 1 Modo monousuario o mantenimiento (opción por defecto).
- 6 Reiniciar (opción `-r`).

shutdown :: Fundamentos

¿Para qué sirve?

`shutdown` permite detener, reiniciar o apagar el sistema de forma segura y ordenada.

Cambio de nivel de ejecución

Se basa en el concepto de nivel de ejecución, enviando una señal `SIGINT` a `init` para que cambie a uno de éstos:

- 0 Apagar (opción `-h`).
- 1 Modo monousuario o mantenimiento (opción por defecto).
- 6 Reiniciar (opción `-r`).

shutdown :: Funcionamiento

Pasos

Cuando se lanza `shutdown`:

- 1 Se notifica a los usuarios.
- 2 A falta de cinco minutos, se bloquea el sistema para que, excepto el `root`, nadie más pueda entrar (`/etc/nologin`).
- 3 Se envía `SIGTERM` y, pasado un tiempo, `SIGKILL` a los procesos no definidos (`inittab`) en el nuevo "run level".
- 4 Se invoca a `/etc/rc.d/rc` pasándole el nuevo nivel de ejecución.
- 5 Si se trata de detener o reiniciar la máquina, llama a `halt` o `reboot`.

shutdown :: Sintaxis

```
/sbin/shutdown [-t segundos] [-opciones] tiempo [aviso]
```

Argumentos

tiempo Instante en el que comenzar la operación. Dos formatos:

- hh:mm
- +m (now equivale a +0).

-t segundos Diferencia de tiempo entre SIGTERM y SIGKILL.

- h Detiene la máquina llamando a halt. Si se indica la opción -P (por defecto), halt invoca a poweroff.
- H inhibe este comportamiento.

shutdown :: Control de acceso

Aviso

En principio, sólo el usuario *root* puede lanzar `shutdown` desde la línea de comandos. Sin embargo, cualquiera puede hacerlo vía `ctrl+alt+del`.

¿Cómo evitarlo?

- Se crea una lista de usuarios autorizados en `/etc/shutdown.allow`.
- Se le pasa la opción `-a` a `shutdown`.

Así, cada vez que se pulse `ctrl+alt+del`, se comprueba si alguno de los usuarios de la lista o el *root* se encuentran en el sistema.

shutdown :: Control de acceso

Aviso

En principio, sólo el usuario *root* puede lanzar `shutdown` desde la línea de comandos. Sin embargo, cualquiera puede hacerlo vía `ctrl+alt+del`.

¿Cómo evitarlo?

- Se crea una lista de usuarios autorizados en `/etc/shutdown.allow`.
- Se le pasa la opción `-a` a `shutdown`.

Así, cada vez que se pulse `ctrl+alt+del`, se comprueba si alguno de los usuarios de la lista o el *root* se encuentran en el sistema.

halt, poweroff y reboot

¿Para qué sirven?

Sus funciones son:

`halt` Detener el sistema.

`poweroff` Apagar el sistema. Llamado por `halt`.

`reboot` Reiniciar.

Realmente, los tres son el mismo programa:

```
$ cd /sbin && ls -lh halt reboot poweroff
-rwxr-xr-x  1 root  root      9,0K 2003-12-23 11:19 halt
lrwxrwxrwx  1 root  root      4 2004-02-04 22:36 poweroff -> halt
lrwxrwxrwx  1 root  root      4 2004-02-04 22:36 reboot -> halt
```

`halt` se comporta de uno u otro modo según su invocación.

halt, poweroff y reboot

¿Para qué sirven?

Sus funciones son:

`halt` Detener el sistema.

`poweroff` Apagar el sistema. Llamado por `halt`.

`reboot` Reiniciar.

Realmente, los tres son el mismo programa:

```
$ cd /sbin && ls -lh halt reboot poweroff
-rwxr-xr-x  1 root  root      9,0K 2003-12-23 11:19 halt
lrwxrwxrwx  1 root  root      4 2004-02-04 22:36 poweroff -> halt
lrwxrwxrwx  1 root  root      4 2004-02-04 22:36 reboot -> halt
```

`halt` se comporta de uno u otro modo según su invocación.

halt, poweroff y reboot :: Funcionamiento

Pasos

Comprueba, consultando `/var/log/wtmp`, si el sistema se encuentra en el nivel de ejecución 0 ó 6.

Sí Pide al núcleo que detenga/apague/reinicie el sistema.

No Invoca a `shutdown`. La opción `-f` evita este comportamiento.

Referencias :: Libros y Manuales

beamericombook M. Carling, Stephen Degler y James Dennis
Administración de Sistemas Linux
Prentice Hall, 1999

beamericombook Vicente López Camacho y otros
Linux. Guía de Instalación y Administración
Mc Graw Hill, 2001

beamericombook Red Hat, Inc.
Red Hat Linux Reference Guide
<http://www.redhat.com/docs/manuals/linux/RHL-9-Manual/ref-guide/index.html>
Red Hat, Inc.

Referencias :: Enlaces y Artículos

beamericonbook

Mark Allen

How Linux Works CTDP Guide, versión 0.6.0

<http://www.comptechdoc.org/os/linux/howlinuxworks/index.html>

beamericonarticle

Roberto Alsina

The Linux Booting Process Unveiled

<http://www.pycs.net/lateral/stories/23.html>

beamericonarticle

Kim Oldfield

The Linux Boot Process (or What happens before the login prompt)

<http://www.pycs.net/lateral/stories/23.html>

beamericonarticle

Jens Benecke

The boot process

<http://www.linuxnetmag.com/en/issue4/m4boot1.html>

Referencias :: Artículos

beamericonarticle

Wayne Marshall

Boot with GRUB

<http://www.wbglinks.net/pages/reads/linux/grub.html>

Linux Journal, 2001