
Arranque y parada del sistema

Diseño y Administración de Sistemas y Redes

Juan Céspedes <cespedes@gsync.escet.urjc.es>



Curso 2005–2006



Arranque genérico de un sistema

- BIOS, *firmware* o ROM de arranque del sistema:
 - POST (*Power On Self Test*).
 - Inicialización del *hardware* necesario para el arranque.
 - Búsqueda del dispositivo de arranque.
 - Cargador del Sistema Operativo a memoria.
 - Ejecución del cargador.
- Cargador del Sistema Operativo:
 - Localización de la imagen del *kernel*.
 - Carga y configura la imagen.
 - Ejecución del *kernel*.

Dispositivos de arranque

- Disco duro, *floppy* o CD-ROM.
 - El primer sector del dispositivo se lleva a memoria y se ejecuta
 - Este primer sector localiza y ejecuta el resto del cargador
 - El cargador carga la imagen del *kernel*, la configura, le pasa las opciones pertinentes y la ejecuta
- Arranque de red.
 - Configuración TCP/IP (BOOTP, DHCP o manual)
 - Descarga de imagen del servidor.

LILO

- *Linux LOader*, es el cargador del *kernel* más común para Linux
- Uso casi exclusivo para arranques desde disco duro
- Configuración en `/etc/lilo.conf`
- Cargador en sector de arranque y en `/boot/map`
- Ejemplo de `/etc/lilo.conf`:

```
boot=/dev/hda
root=/dev/hda1
image=/boot/vmlinuz
    label=Linux
image=/boot/vmlinuz-2.6.10
    label=2.6.10
```

GRUB

- *GRand Unified Bootloader*, es el cargador del *Proyecto GNU*
- Muy versátil y configurable, permite arrancar casi cualquier sistema operativo
- Entiende internamente FAT, ext2, Reiser, JFS...
- Configuración en `/boot/grub/menu.lst`. Ejemplo:

```
title Debian GNU/Linux
root (hd0,1)
kernel /boot/vmlinuz root=/dev/hda1 ro

title Debian GNU/Linux, kernel 2.6.10
root (hd0,1)
kernel /boot/vmlinuz-2.6.10 root=/dev/hda1 ro
```

SYSLINUX / ISOLINUX

- Orientado a arranques desde floppy o CD
- Prácticamente todas las distribuciones de Linux lo utilizan como cargador durante la instalación del sistema
- Configuración en `/syslinux.cfg` (`/isolinux.cfg`):

```
DEFAULT Linux
DISPLAY readme.txt
LABEL Linux
    KERNEL linux
    APPEND ro initrd=image.gz
LABEL dos
    KERNEL opendos.bss
```

Arranque remoto

- En el cliente, tiene que estar activo el “arranque por red”
- En el servidor, es necesario:
 - Servidor de DHCP (para configurar la red)
 - Servidor de TFTP (transmisión de cargadores, configuraciones y *kernels*)
 - Cargador (los más comunes son PXELINUX, ETHERBOOT, COMBOOT) que descargan el *kernel*, lo configuran y lo ejecutan

Configuración del *kernel*

- Al *kernel* se le pueden pasar una serie de parámetros para:
 - Configurar dispositivos (discos duros, tarjetas de red, puertos serie, memoria (**mem=XXX**)...)
 - Indicar sistema de ficheros raíz (**root=/dev/XXX**)
 - Condiciones especiales: monousuario, distinto init (**init=XXX**)...
- Cada cargador tiene su manera de pasarle parámetros al *kernel*
- Dentro del *kernel*, los parámetros con los que se ha arrancado se pueden ver en **/proc/cmdline**

Arranque del *kernel* de UNIX

- Comprobación del procesador.
- Búsqueda e inicialización de *hardware*.
- Inicialización de la pila TCP/IP.
- Monta sistema de ficheros raíz (“root filesystem”).
- Prepara la consola, terminales virtuales, etc.
- Ejecuta el primer proceso del sistema, “/sbin/init”.
- Todos los mensajes que genera el *kernel* se pueden ver con la orden “**dmesg**”.

Sistema de ficheros raíz

- Puede ser local (disquete, disco duro, partición)
- remoto (DHCP + NFS)
- *ramdisk* (descargado junto con el *kernel*)

Lo más habitual últimamente es el uso de **initrd**:

- Imagen comprimida que se distribuye junto con el *kernel*
- Se carga en RAM (*ramdisk*) y se ejecuta un *script* (**linuxrc**)
- Este *script* carga módulos, configura discos y monta el sistema raíz
- Una vez montado, el **initrd** desaparece y procede el arranque normal

Proceso *init*

- Primer proceso del sistema.
- PID=1 (siempre).
- “Padre de todos los procesos”.
- Adopta y recoge a todos los procesos huérfanos.
- No se puede morir.
- Fichero de configuración: `/etc/inittab`
- Niveles de ejecución: “*runlevels*”.

Runlevels

- Los *runlevels* son números del 0 al 6.
- 3 de ellos tienen significado especial:
 - 0: Parar la máquina (*shutdown*).
 - 1: Modo monousuario (*single-user*).
 - 6: Reiniciar la máquina (*reboot*).
- El resto pueden configurarse para que arranquen y paren determinados servicios en cada uno.
- NOTA: El significado de cada *runlevel* es un convenio, depende exclusivamente de lo que indique el fichero de configuración `/etc/inittab`.

/etc/inittab

- Formato de cada línea:
`identificador:runlevel:acción:proceso-a-ejecutar`
- Las acciones más importantes son “sysinit”, “initdefault”, “wait”, “respawn”.
 - **sysinit**: Orden a ejecutar en el arranque del sistema, antes de preocuparse por los *runlevels*.
 - **initdefault**: Indica cuál es el *runlevel* al que pasará el sistema después de la inicialización.
 - **wait**: Ejecuta la orden dada si estamos en el *runlevel* indicado, y espera a que termine.
 - **respawn**: Ejecuta la orden dada si estamos en el *runlevel* indicado, y la relanza una y otra vez si esta termina.

Runlevels en un sistema Linux

- En el fichero `/etc/inittab` están las siguientes líneas:

```
si::sysinit:/etc/init.d/rcs
10:0:wait:/etc/init.d/rc 0
11:1:wait:/etc/init.d/rc 1
...
16:6:wait:/etc/init.d/rc 6
```
- ... que se encargan de ejecutar los *scripts* que haya en el directorio `/etc/rcS.d`, `/etc/rc0.d`... `/etc/rc6.d`.
- En `/etc/rcN.d` hay *scripts* que comienzan por “s”, que inician un servicio (se les llama con el argumento “start”), y que comienzan por “k”, que lo finalizan (argumento “stop”).

scripts de inicio

- Todos los *scripts* de inicio y fin de servicios están en `/etc/init.d`, pero `init` los ejecuta desde `/etc/rcN.d`, donde hay enlaces simbólicos al directorio real:

```
/etc/rc0.d/K20ssh -> ../init.d/ssh
/etc/rc1.d/K20ssh -> ../init.d/ssh
/etc/rc2.d/S20ssh -> ../init.d/ssh
/etc/rc3.d/S20ssh -> ../init.d/ssh
/etc/rc4.d/S20ssh -> ../init.d/ssh
/etc/rc5.d/S20ssh -> ../init.d/ssh
/etc/rc6.d/K20ssh -> ../init.d/ssh
```

- Solo hay un *script* por servicio, y se comporta de distinta manera dependiendo de si su primer argumento es “`start`” o “`stop`”.

Ejemplo de *script* de inicio

```
#!/bin/sh -e
# Script de arranque del servidor ssh
#
case "$1" in
  start)
    echo -n "Arrancando servidor ssh..."
    sshd
    echo "done."
    ;;
  stop)
    echo -n "Parando servidor ssh..."
    kill -TERM $( cat /var/run/sshd.pid )
    echo "done."
    ;;
  restart)
    $0 stop
    $0 start
    ;;
  *)
    echo "Uso: $0 {start|stop|restart}"
    exit 1
esac
exit 0
```

Órdenes relacionadas con el arranque y parada

- `runlevel`: Indica el *runlevel* anterior y el actual.
- `init`, `telinit`: Avisan al proceso `init` de un cambio de *runlevel*.
- `shutdown`, `halt`, `reboot`: se utiliza para parar o reiniciar el sistema:
 - `“shutdown -h now” = “halt” = “init 0”`.
 - `“shutdown -r now” = “reboot” = “init 6”`.