



## Arranque y parada del sistema

### Arranque del sistema operativo

- Los pasos que se siguen durante el proceso de arranque de Linux son:
  1. Carga del kernel a memoria
  2. Inicialización del sistema operativo y configuración hardware
  3. Ejecución de procesos especiales del núcleo
  4. Arranque en modo usuario único
  5. Arranque en modo multiusuario
  6. Ejecución de servicios



## Arranque y parada del sistema

### Carga del kernel a memoria

- Durante el arranque del ordenador, el sistema operativo ha de cargarse en memoria desde su situación permanente (disco, diskette, red,...)
- En Linux, el kernel es un programa que se llama **vmlinux** ó **vmlinuz** (versión comprimida)
  - ✍ Se encuentra en “/” o en “/boot”
  - ✍ El nombre especifica el número de versión. Ej.: *vmlinuz-2.2.5-15*
- La carga se realiza en dos fases:
  - ✍ La BIOS lee un pequeño programa (**cargador**)
  - ✍ Éste lee a memoria el sistema operativo elegido



## Arranque y parada del sistema

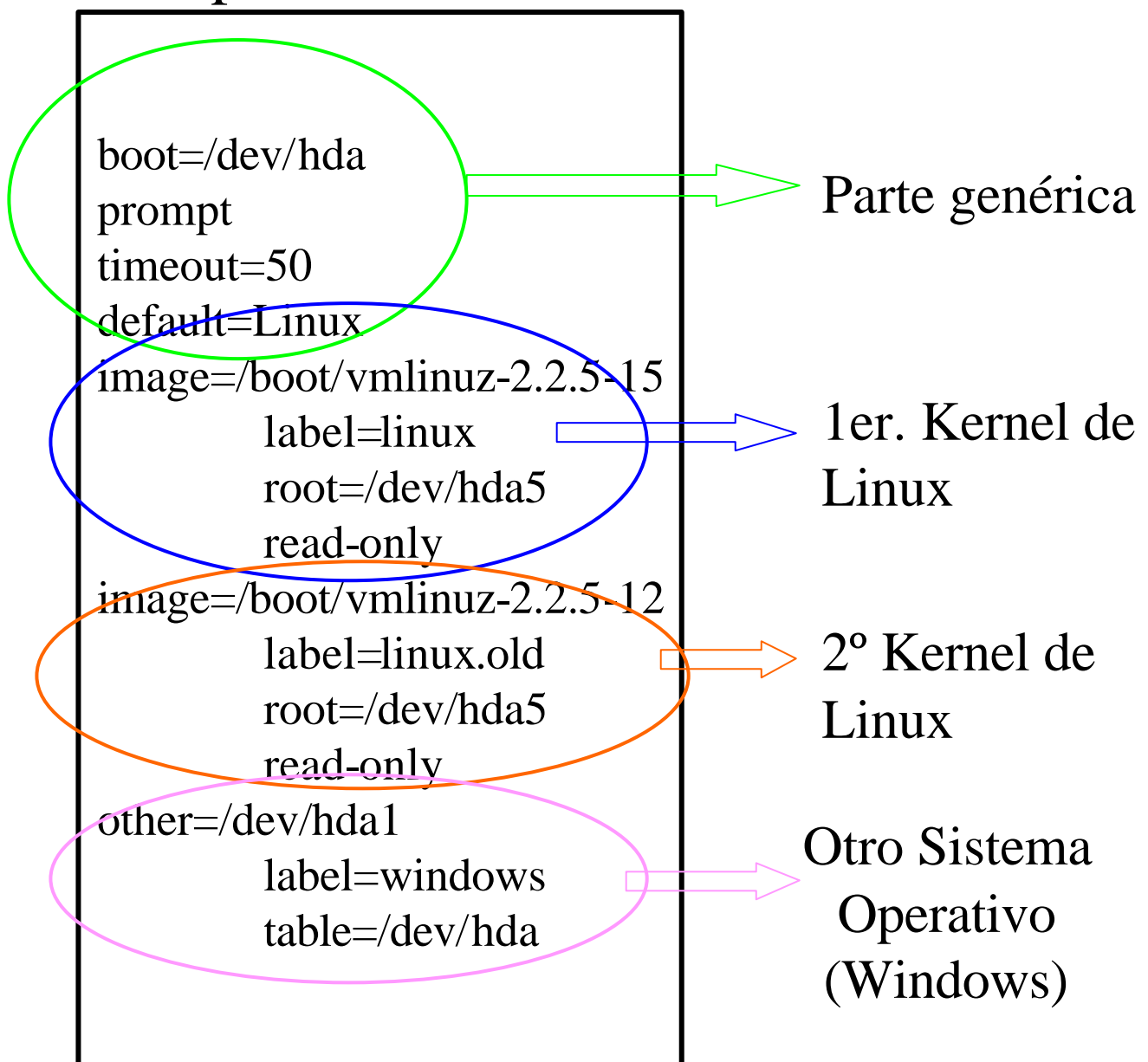
### Gestor de arranque de Linux: LILO

- LILO (LI<sup>n</sup>ux LO<sup>a</sup>der) es uno de los *cargadores* que se utilizan para el arranque de Linux.
- Permite arrancar varios sistemas operativos (DOS, OS/2, Windows 9x, Windows NT, Windows 2K)
- Permite arrancar distintas “*imágenes*” de kernel de Linux.
- Consta de varios programas, archivos de datos y un archivo de configuración ? **/etc/lilo.conf**
- El programa **/sbin/lilo** lee la información de configuración y reescribe el sector de arranque correspondiente (normalmente MBR)



## Configuración de LILO

- Fichero de configuración `/etc/lilo.conf` típico:





## Arranque y parada del sistema

### El proceso init

- Una vez el kernel ha sido “cargado” a memoria, inicia su ejecución
  - ✍ Comprueba el hardware
  - ✍ Carga los controladores
  - ✍ Monta el sistema de ficheros “/”
  - ✍ Inicia la ejecución del proceso 1:  
**init**
    - Arranca distintos “threads”
- Lee el fichero **/etc/inittab** para determinar las próximas acciones a realizar ? **Niveles de ejecución**
- Inicialmente ejecuta el script **/etc/rc.d/rc.sysinit** (arranca los servicios básicos)



## Arranque y parada del sistema

### El proceso init (ii)

- Niveles de ejecución
  - ✍ Indican “estados” en los que se puede encontrar un sistema en ejecución
  - ✍ Cada nivel indica la funcionalidad que el sistema debería estar ofreciendo
  - ✍ Siete niveles:

0	Detiene la máquina
1	Modo monousuario (sin procesos de servidor)
2	Modo multiusuario sin soporte de red
3	Modo multiusuario completo
4	No usado
5	Igual que nivel 3, pero arranca en modo gráfico
6	Reinicio del sistema



## Arranque y parada del sistema

### Scripts rc

- Normalmente, las acciones a realizar cuando se entra o se sale de un nivel de ejecución consisten en la ejecución de scripts encargados de iniciar o parar los servicios particulares del nivel de ejecución
- Gran número de servicios ? Se separan en múltiples scripts
- Solución: **scripts rc**
  - ✍ Un programa principal: **/etc/rc.d/rc**
  - ✍ Un conjunto de scripts a ejecutar en cada nivel
  - ✍ El script “rc” llama a los scripts apropiados en el orden adecuado



## Arranque y parada del sistema

### Scripts rc: Implementación

- Para cada nivel de ejecución ? Existe un subdirectorio en `/etc/rc.d` de nombre **`/etc/rcX.d`** (X ? nivel de ejecución)
- Los ficheros en el interior siguen una convención de nombres:
  - ✍ Primer carácter: S (arranque) ó K (parada)
  - ✍ Dos dígitos numéricos: Fijan el orden de ejecución
  - ✍ Nombre del script: Los scripts se encuentran en **`/etc/rc.d/init.d`**
- Los scripts de arranque admiten dos opciones como parámetros: **start** o **stop**. Los scripts que empiezan con K se ejecutan pasando la opción *stop* y los que empiezan con S se les pasa *start*





## Arranque y parada del sistema

### Scripts rc: Ejemplo

- Supongamos que hemos implementado un nuevo servicio: “Servidor de hora”. Queremos que se arranque automáticamente al entrar en el nivel 3
- Nos creamos un script, p. ej. [horad](#)
  - ✍ Acepta como parámetros: start o stop
  - ✍ Situamos el script en [/etc/rc.d/init.d](#)
  - ✍ Creamos un enlace simbólico en [/etc/rc.d/rc3.d](#) llamado, por ejemplo, [S50horad](#), que apunte a [horad](#).
- Cuando se entre en el nivel 3, el script rc llamará a todos los scripts que empiecen con S y les pasará como parámetro “start”



## Arranque y parada del sistema

### Scripts rc: Herramientas

- Existen comandos y herramientas gráficas que ayudan a la gestión de los scripts rc:
  - ✍ chkconfig
  - ✍ tksysv,...

### Apagado del sistema

- “Nunca” apagar directamente el equipo
- Utilizar los comandos “shutdown”, “reboot”, “halt” ó la combinación “Ctrl+Alt+Supr”
- El comando shutdown permite notificar a los usuarios que el sistema va a detenerse.