

Proceso de Arranque y Cargador

Ampliación de Sistemas Operativos

José R. López Medina

2004 - 2005

1. Proceso de Arranque, Inicio y Cierre del Sistema

- **Método abierto y configurable**
- **Permite a los usuarios lanzar aplicaciones que deseen en momento de arranque**
- **Configuración del cierre no necesaria, pero posible**
- **Resolver problemas relacionados con el inicio y el cierre del sistema**

1.1. Proceso de Arranque

- **Etapas Básicas x86:**
 1. **BIOS** comprueba y **lanza primera etapa** del gestor de arranque
 2. **Primera etapa** se **autocarga** en memoria y **lanza segunda etapa** desde partición **/boot/**
 3. **Segunda etapa** carga **kernel** en memoria, cargándose **módulos** necesarios y montando **partición "root"** sólo lectura
 4. **Kernel** transfiere **control** proceso de arranque a **/sbin/init**
 5. **/sbin/init** carga **servicios** y **herramientas** de espacio del **usuario** y **monta particiones** listadas en **/etc/fstab**
 6. **Usuario** se **presenta con** intérprete comandos de registro para el sistema **Linux arrancado**.

1.2. Vista Detallada del Proceso de Arranque

- **Depende de la plataforma hardware**
- **Encontrado el kernel y cargándose el sistema, el proceso de arranque es el mismo para todas las arquitecturas**
- **Caso a estudiar: arquitectura x86**

1.2.1. La BIOS

- Cuando un ordenador se carga, el procesador busca al final de la memoria el programa BIOS y lo ejecuta.
- Es la encargada de **controlar** el **primer paso** del proceso de arranque así como proporcionar **interfaz de bajo nivel** para periféricos.
- Chequea los periféricos y **localiza dispositivo** con el que **arrancar** el sistema, **cargando** en memoria cualquier programa que encuentre en el **primer sector** de este dispositivo (MBR).
- **Master Boot Record (MBR):** 512B, contiene instrucciones código máquina para arranque del equipo, invocar gestor de arranque y tabla de particiones.
- Una vez cargado en memoria, la **BIOS cede control** del proceso de arranque al cargador del **gestor de arranque**.

1.2.2. El Gestor de Arranque

- Gestores de arranque más comunes:
 - GRUB (más nuevo)
 - LILO
- Etapas de los gestores de arranque:
 1. Código binario **máquina** en MBR que localiza gestor arranque de **segunda etapa** y **carga primera parte** de éste en memoria
 2. Usa información del MBR para **determinar opciones** de arranque disponibles para el **usuario**
- Tras la cargar segunda etapa se presenta al usuario una **pantalla** mostrando los distintos S.O. o **kernels disponibles**
- **Ubica** la imagen apropiada de *initial RAM disk* (**initrd**) en memoria y es usada por el kernel **para cargar controladores** del **arranque** del sistema.
- Tras estar el kernel e *initrd* en memoria, **controla** el proceso de **arranque para el kernel**.

1.2.3. El kernel

- Cuando es cargado, **se inicializa y configura** la memoria del PC así como el **hardware** conectado al sistema (procesadores, E/S, almacenamiento,...)
- **Busca** imagen *initrd* en zona específica de memoria, la **descomprimirá**, la **montará** y **cargará** los **controladores** necesarios.
- **Inicializa** los **ficheros** antes de **desmontar** *initrd*.
- Crea dispositivo “root”, **monta** **partición** “root” (sólo lectura) y **libera memoria** no usada, quedando el kernel cargado en memoria y operativo.
- **Configuración** del entorno de **usuario**: `/sbin/init`

1.2.4. Programa /sbin/init

- Coordina el resto del proceso de arranque y configura el ambiente del usuario
- Etapas del comando init:
 1. Ejecuta el script **/etc/rc.d/rc.sysinit**
 - */etc/sysconfig/clock*
 - */etc/rc.serial*
 2. Ejecuta el script **/etc/inittab**
 3. Configura la librería de función de fuente **/etc/rc.d/init.d/functions** para el sistema
 4. Inicia todos los procesos de fondo buscando en el directorio apropiado “rc” por el nivel de ejecución especificado en **/etc/inittab**
 - *K* → */etc/rc.d/init.d/<comando> stop*
 - *S* → */etc/rc.d/init.d/<comando> start*

1.3. Ejecutar Programas Adicionales en el Arranque

- “**init**” ejecuta el script ***/etc/rc.d/rc.local*** en el momento de **arranque** o al **cambiar niveles** de ejecución.
- Para arrancar servicios especiales o inicializar dispositivos basta con agregar los comandos a dicho script
- Para configurar puertos seriales se usa el script ***/etc/rc.serial*** (comandos “**setserial**”)

1.4. Niveles de Ejecución de SysV Init

- Controla programas “init” que se lanzan o detienen cuando se inicializa un nivel.
- Ficheros de configuración en **/etc/rc.d/**
- **/etc/rc.d/init.d/** contiene **scripts** usados por “**/sbin/init**” cuando se controlan los servicios

1.4.1. Niveles de Ejecución

- Son un **estado**, o modo, **definido por** los **servicios** listados **en** el directorio SysV **/etc/rc.d/rc<x>.d/**, donde **<x>** es el número de nivel de ejecución.
- Sistemas diferentes se pueden usar de formas diferentes.
- Las características de un nivel dado determinan qué servicios son detenidos o iniciados por “init”.

1.4.1. Niveles de Ejecución (2)

- **Niveles de ejecución** por defecto de RH:
 - 0 → Parar
 - 1 → Modo texto usuario único
 - 2 → Sin usar (usuario-definible)
 - 3 → Modo texto multiusuario completo
 - 4 → Sin usar (usuario-definible)
 - 5 → Modo gráfico multiusuario completo
 - 6 → Rearrancar
- Nivel de ejecución por defecto */etc/inittab*:
 - *Id:5:initdefault:*

1.4.2. Utilidades de los Niveles de Ejecución

- “initscript utility”: herramientas que permiten configurar los niveles de ejecución.
 - `/sbin/chkconfig`
 - `/sbin/ntsysv`

1.5. Apagar

- Comando “**/sbin/shutdown**” (usuario root)
- Opciones más usadas:
 - “/sbin/shutdown –h now” (apagar)
 - “/sbin/shutdown –r now” (reiniciar)
- Resto de usuarios disponen de los comandos (niveles 1 al 5):
 - “halt” (apagar)
 - “reboot” (reiniciar)
- **OJO:** No apagar el PC hasta que se indique que el sistema ha sido detenido, podría causar que el sistema de archivos se corrompa.

2. Gestores de Arranque

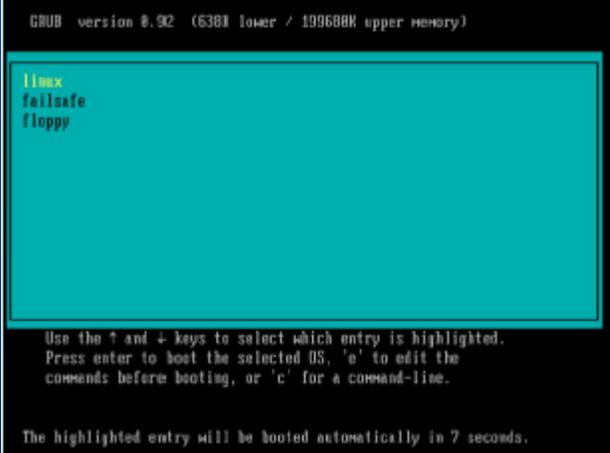
- **Existe en el disco duro primario del sistema (u otros dispositivos) y es responsable de la carga del kernel de Linux o de otros SO en memoria.**

2.1. Gestores de Arranque y Arquitectura del Sistema

- **Cada arquitectura usa un gestor de arranque diferente**
 - **Alpha: about**
 - **Itanium: ELILO**
 - **x86: GRUB y LILO**

2.2. GRUB

- GNU **GR**and **U**nified **B**oot loader
- Permite al usuario seleccionar qué SO instalado o kernel descargar en el momento de arranque del sistema
- Permite **transmitir argumentos** al kernel



```
GRUB version 0.92 (638K loader / 199688K upper memory)

linux
failsafe
floppy

Use the ↑ and ↓ keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the
commands before booting, or 'c' for a command-line.

The highlighted entry will be booted automatically in 7 seconds.
```

2.2.1. Proceso de Arranque en x86 y GRUB

- Se carga a sí mismo en memoria en diferentes etapas:
 1. La **etapa 1** o cargador de arranque primario **se lee en memoria** con el BIOS desde el MRB. **Lee** en la memoria al gestor de arranque de la **etapa 1.5** si es necesario.
 2. La **etapa 2** o el gestor de arranque secundario **se lee** en la memoria. **Lee** el sistema operativo o el **kernel** y “**initrd**” en la **memoria**.

2.2.2. Funciones GRUB

- **Características más importantes**
 - Proporciona **entorno pre-OS** basado en **comandos** para máquinas x86
 - Soporta el modo **Direccionamiento Lógico de Bloques (LBA)**.
 - Puede leer **particiones ext2**

2.3. Instalación de GRUB

- Si tras instalación de Linux se quiere instalar GRUB habrá que realizar los siguientes pasos:
 1. Instalar la última versión de GRUB
 2. Ubicar GRUB en el MBR:
 - `“/sbin/grub-install /dev/hdX”` siendo X = a,b,c,d (dependiendo de la ubicación del HD)

2.4. Terminología de GRUB

- **Cómo se hace referencia a los dispositivos**

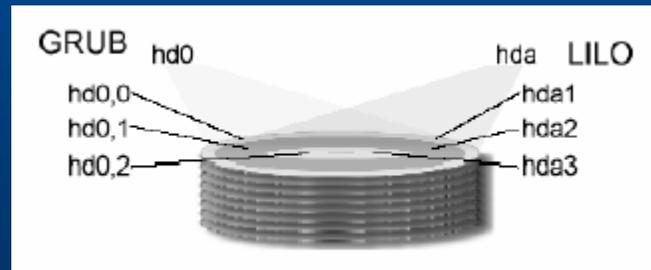
2.4.1. Nombre de Dispositivos

- **Nomenclatura utilizada:**

- (*<tipo-disp><nº-disp-bios>, <nº-partición>*)

- (hd0,0) → Primer HD, Primera partición

- (hd1,4) → Segundo HD, Quinta partición



2.4.2. Nombres de Archivos y Listas de Bloqueo

- **Especificación de archivo:**
 - *(<tipo-disp><n-disp-bios>,<n-part>) /path/file*
- **Especificación de archivos que no aparecen en el sistema de archivos (gestor arranque de cadena, Windows):**
 - *0+50,100+25,200+1*
 - *(hd0,0)+1*
 - *chainloader+1*

2.4.3. Sistema de Archivos Raíz de GRUB

- **No tiene nada que ver con el sistema de archivos raíz de Linux.**
- **Es la partición raíz de un dispositivo concreto.**

2.5. Interfaces de GRUB

- **Dispone de 3 interfaces para distintos niveles de funcionalidad:**
 - **Interfaz de menú**
 - **Interfaz del editor de menú de entrada**
 - **Interfaz de comandos**

2.5.1. Orden de Uso de Interfaces

- Al cargar segunda etapa se busca el archivo de configuración y lo utiliza para crear la lista de menú → **Interfaz de menú.**
- Si no se encuentra o no se puede leer el archivo de configuración → **Interfaz de línea de comandos.**
- Si el archivo de configuración no es válido, se imprime error y se pide introducir valores → **Interfaz del editor de menú de entrada.**

2.6. Comandos de GRUB

- **Comandos más útiles:**

- **boot:** Arranca S.O. especificado
- **chainloader <archivo>:** Carga archivo como gestor de encadenamiento
- **displaymem:** Uso actual de memoria
- **initrd <archivo>:** Especifica initrd para usar al arrancar
- **install <stage-1> <install-disk> <stage2> p <config-file>:** Instala GRUB en MBR.
- **kernel <kernel-file> <option-1> ... <option-N>**
- **root <disp-y-partición>:** Partición raíz de GRUB
- **rootnoverify <disp-y-partición>:** No monta la partición
- **info grub**

2.7. Archivo de Configuración de Menú de GRUB

- **/boot/grub/grub.conf**
- Usado para crear la lista de la interfaz de menú de GRUB
- Permite seleccionar un grupo predefinido de comandos para su ejecución

2.7.1. Comandos Especiales del Archivo de Configuración

- **color <color-normal> <color-selec>**
 - colorred/blackgreen/blue
- **default <nombre-título>**: Título por defecto
- **fallback <nombre-título>**: Título secundario
- **hiddenmenu**: Esconde interfaz de GRUB si no se pulsa la tecla [Esc]
- **password <contraseña>**
- **timeout**
- **splashimage**: Imagen de pantalla que se usa al arrancar
- **title**
- **#**: Sirve para comentarios

2.7.2. Estructura del Archivo de Configuración

- Comandos para configurar preferencias globales de interfaz de menú al inicio, seguido de las entradas de SO o kernels

```
default=0
timeout=10
splashimage=(hd0,0) /grub/splash.xpm.gz
```

```
title Linux
root(hd0,0)
kernel /vmlinuz ro root=/dev/sda2
initrd /initrd.img
```

```
title Windows
rootnoverify (hd0,0)
chainloader+1
```

2.8. LILO

- **L**inux **L**Oader, usado para arrancar Linux en x86 por muchos años.
- Según el tipo de hardware, GRUB puede causar problemas.



2.8.1. LILO y el Proceso de Arranque x86

- A diferencia de GRUB, es un gestor de arranque en 2 etapas:
 1. La **etapa 1** o el gestor de arranque primario **se lee en la memoria** por la BIOS desde el MBR.
 2. La **etapa 2** o el gestor de arranque secundario **se lee en memoria, lee el SO o el kernel** y lleva a cabo **“initrd”** en memoria.

2.8.2. LILO vs GRUB

- LILO funciona parecido a GRUB a excepción de:
 - **No posee** ninguna **interfaz** del **comando** interactiva.
 - **Almacena información** sobre la localización del kernel o de si otro SO se debe cargar **en el MBR**.
 - **No puede leer** las particiones **ext2**.

2.9. Opciones en /etc/lilo.conf

- El comando `/etc/sbin/lilo` usa este archivo para determinar la información para el MBR.

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
message=/boot/message
lba32
default=linux
```

```
image=/boot/vmlinuz
label=linux
initrd=/boot/initrd.img
read-only
root=/dev/hda5

other=/dev/hda1
label=dos
```

2.10. Cambiar los Niveles de Ejecución en Tiempo de Arranque

- **LILO:**

1. Acceder intérprete de comandos “boot:” **[Ctrl]+[x]**
2. Escribir “**linux<nº-nivel-ejecución>**” (1-5, single o emergency)

- **GRUB:**

1. En pantalla gráfica, seleccionar etiqueta Linux y pulsar **[e]** para editarla.
2. Ir parte inferior línea del kernel y pulsar **[e]** para modificarla.
3. Escribir número 1-5, single o emergency + **[Intro]**
4. Pulsar **[b]** para arrancar el sistema

2.11. Recursos Adicionales

- **Consultar los siguientes recursos para descubrir más cosas sobre cómo funcionan GRUB y LILO.**

2.11.1. Documentación Instalada

- `/usr/share/doc/grub-<version>/`
- “info grub”
- `/usr/share/doc/lilo-<version>/`

2.11.2. Sitios Web Útiles

- <http://www.gnu.org/software/grub/>
- <http://www.uruk.org/orig-grub/>
- <http://www.redhat.com/mirrors/LDP/HOWTO/mini/Multiboot-with-GRUB.html>
- <http://www.linuxgazette.com/issue64/kohli.html>
- <http://www.tldp.org/HOWTO/mini/LILO.html>