

Sistemas de arranque en Linux

Tabla de Contenidos

Sistemas de arranque en Linux.....	2
1 Gestor de arranque.....	2
2 LILO.....	2
Creación del disco de arranque o instalación del LILO.....	2
Instalación de Lilo.....	3
Utilizando LILO.....	4
Opciones de LILO.....	6
3 GRUB.....	8
¿Qué es Grub?.....	8
Características de GRUB	8
Nomenclatura	11
Instalación	13
Arranque	17
Configuración	20
Interfaz de usuario de GRUB	23
Invocar el intérprete grub	27
Como instalar GRUB mediante grub	29
La correspondencia entre unidades de la BIOS y dispositivos del SO	30
Invocar grub-install	31



★ Sistemas de arranque en Linux

1 Gestor De Arranque

Todas las particiones primarias y extendidas comienzan con lo que se llama sector de arranque. El IPL por defecto, únicamente pasa el control al sector de arranque de la partición marcada con el flag (bandera) activo, también llamado flag de arranque. Como se puede suponer, sólo debería haber una partición con el flag de activo marcado. De esta manera se controla la partición de arranque.

En el sector de arranque se encuentra otro programa que es el encargado de arrancar el sistema operativo. En el caso de Linux, estos programas son Lilo y Grub.

Podemos usar otros gestores de arranque principal, en lugar del de por defecto, que incorporen diversas funcionalidades como menús, colores, usuarios, claves u otras opciones. Algunos gestores de arranque comerciales son el System Commander, el BootMagic, o el BootStar.

Lilo no sólo puede arrancar Linux, sino que también puede arrancar otros sistemas operativos. Además, no ha de instalarse necesariamente en el sector de arranque de una partición, sino que puede instalarse en el MBR, es decir, en el IPL, convirtiéndose en el gestor de arranque principal.

Lilo también puede instalarse en el sector de arranque de una partición extendida, lo cual viene muy bien si se tiene Linux instalado en una unidad lógica y queremos usar otro gestor de arranque principal que no sea Lilo.

2 LILO

Creación del disco de arranque o instalación del LILO

Cada distribución proporciona mecanismos para arrancar Linux cuando ya esté instalado en su sistema. En la mayoría de los casos se creará un disquete "boot" que contiene el núcleo de Linux configurado para usar con su recién creado sistema de ficheros raíz. Para arrancar Linux, deberá hacerlo desde ese disquete y tras el arranque se pasará el control a su disco duro. En otras distribuciones, el disco de arranque es el propio disquete de instalación.

La mayoría de las distribuciones le van a dar la opción de instalar LILO en su disco duro. LILO es un programa que se instala en el registro maestro de arranque del disco, y está preparado para arrancar varios sistemas operativos, entre los que se incluyen MS-DOS y Linux, permitiéndole elegir qué sistema quiere arrancar en cada momento.

En el caso de la distribución Slackware, la opción Configure del menú setup le permitirá tanto crear un disquete de arranque como instalar LILO. Esas opciones deberían ser suficientemente autoexplicativas. El menú Configure también sirve para especificar el módem y el ratón de que dispone, así como su zona horaria.

Con el fin de instalar LILO correctamente, necesita conocer bastante información acerca de la configuración de su disco por ejemplo, qué particiones contiene cierto sistema operativo, cómo arrancar cada sistema operativo, etc. La mayoría de las distribuciones, cuando se instala LILO, tratan de elegir la mejor configuración para éste. Aunque no es lo habitual, la instalación automatizada de LILO puede fallar, dejando el registro de arranque maestro de su disco inservible (aunque es difícil que éste llegue a hacerle perder datos de su disco). En concreto, si utiliza el Boot Manager de OS/2, usted no deberá instalar LILO mediante el procedimiento automático para ello, habrá que seguir instrucciones especiales.

En muchos casos, lo mejor es usar un disquete de arranque, hasta que usted esté en condiciones de configurar LILO a mano.

Instalación de Lilo

Para instalar Lilo hay que crear primero el archivo de configuración `/etc/lilo.conf` y después ejecutar como superusuario el comando: `lilo`.

Hay programas como el `linuxconf`, que permiten crear el archivo de configuración de Lilo, pero la verdad es que una vez que se conoce el formato de este archivo, es más fácil, seguro, rápido y potente crearlo uno mismo a mano.

Antes de instalar Lilo, podemos hacer una copia de seguridad de la zona que vamos a sobrescribir. Por ejemplo, para hacer una copia del MBR del primer disco IDE, ejecutamos el comando:

```
dd if=/dev/hda of=<archivo> bs=512 count=1
```

Para restaurar el IPL salvado sólo tenemos que ejecutar el comando:

```
dd if=<archivo> of=/dev/hda bs=446 count=1
```

También podemos instalar el IPL por defecto, desde MS-DOS, con el comando:



fdisk /mbr

Después de cargarse Lilo en el arranque, mira si se encuentra activo BloqMay o BloqDesp, o si se ha pulsado Control, Alt o Shift, en cuyo caso muestra el boot prompt, desde el que se puede elegir el OS que se quiere arrancar. Para más información sobre este tema, véase el BootPrompt-HowTo.

Utilizando LILO

LILO es un programa que reside en el sector de arranque del disco duro. Este programa se ejecuta cuando el sistema se inicia desde el disco duro, y puede arrancar automáticamente Linux desde una imagen de núcleo almacenada en el propio disco duro.

LILO puede utilizarse también como una primera etapa de carga de varios sistemas operativos, permitiendo seleccionar en tiempo de arranque qué sistema operativo (como Linux o MS-DOS) arrancar. Cuando se arranca utilizando LILO, se inicia el sistema operativo por defecto, a menos que pulse `|_ctrl_|`, `|_alt_|`, o `|_shift_|` durante la secuencia de arranque. Si se pulsa cualquiera de estas teclas, se le presentará un indicador de arranque, donde debe teclear el nombre del sistema operativo a arrancar (como puede ser "linux" o "msdos"). Si se pulsa la tecla `|_tab_|` en el indicador de arranque, se le presentará una lista de los sistemas operativos disponibles.

La forma más simple de instalar LILO es editar el fichero de configuración, `/etc/lilo.conf`, y ejecutar el comando

```
# /sbin/lilo
```

El fichero de configuración de LILO contiene una "estrofa" para cada sistema operativo que se pueda querer arrancar. La mejor forma de mostrarlo es con un ejemplo de un fichero de configuración LILO. El ejemplo siguiente es para un sistema que tiene una partición raíz Linux en `/dev/hda1` y una partición MS-DOS en `/dev/hda2`.

```
# Le indicamos a LILO que modifique el registro de arranque de
```

```
# /dev/hda (el primer disco duro no-SCSI). Si se quiere arrancar desde
```

```
# una unidad distinta de /dev/hda, se debe cambiar la siguiente línea
```

```
boot = /dev/hda
```

```
# Nombre del cargador de arranque. No hay razón para cambiarlo, a menos
```

```
# que se est haciendo una modificación será del LILO
```



```
install = /boot/boot.b
```

```
# Dejemos a LILO efectuar alguna optimización.
```

```
compact
```

```
# Estrofa para la partición raíz de Linux en /dev/hda1.
```

```
image = /etc/Image # Ubicación del kernel
```

```
label = linux # Nombre del SO (para el menú de arranque de LILO)
```

```
root = /dev/hda1 # Ubicación de la partición raíz
```

```
vga = ask # Indicar al núcleo que pregunte por modos SVGA
```

```
# en tiempo de arranque
```

```
# Estrofa para la partición MSDOS en /dev/hda2.
```

```
other = /dev/hda2 # Ubicación de la partición
```

```
table = /dev/hda # Ubicación de la tabla de partición para /dev/hda2
```

```
label = msdos # Nombre del SO (para el menú de arranque)
```

La primera "estrofa" de sistema operativo en el menú del fichero de configuración será el sistema operativo que arrancará LILO por defecto. Se puede seleccionar otro sistema operativo en el indicador de arranque de LILO, tal y como se indicó anteriormente.

Recuerde que cada vez que actualice la imagen del núcleo en disco, se debe reejecutar `/sbin/lilo` para que los cambios queden reflejados en el sector de arranque de su unidad.

También tenga en cuenta que si utiliza la línea `"root ="`, no hay motivo para utilizar `rdev` para poner la partición raíz en la imagen del núcleo. LILO se encarga de ponerlo en tiempo de arranque.



Opciones de LILO

- Opciones globales

boot=<device>

Lugar donde instalar Lilo. Por defecto se toma la partición donde está el actual sistema de archivos raíz de Linux.

linear

Usar direcciones lineales en lugar de 3D. Necesario en discos SCSI.

message=<file>

Imprime el archivo indicado al cargar Lilo. Este archivo debe ocupar menos de 65535 bytes. El carácter 0xff hace que se borre la pantalla.

timeout=<tsecs>

Décimas de segundo que espera Lilo la pulsación de una tecla antes de arrancar un OS. Por defecto se espera indefinidamente.

prompt

Fuerza la aparición del prompt de Lilo.

default=<name>

OS de arranque por defecto.

- Opciones comunes

label=<name>

Define el nombre de la imagen Linux o del OS. Por defecto se usa el nombre del archivo principal de la imagen Linux o del OS.

alias=<name>

Define un segundo nombre para la imagen Linux o el OS.

single-key

Permite arrancar la imagen Linux o el OS mediante la pulsación de una tecla. Para ello, o bien la etiqueta, o bien su posible alias, o ambos, han de ser de sólo una letra.

password=<passwd>

Define una clave de acceso.

Excepto las opciones label y alias, el resto pueden colocarse junto a las opciones generales, pasando entonces a ser el valor por defecto en todas las imágenes Linux y OS.

- Opciones particulares de imágenes Linux

root=<device>

Especifica el dispositivo sobre el que montar el sistema de ficheros raíz.

read-only

Opción recomendable para uso normal.

append=<cadena>

Añade la cadena indicada a la línea de parámetros que se pasa al kernel en el arranque.

Estas opciones pueden colocarse junto a las opciones generales, pasando entonces a ser el valor por defecto en todas las imágenes Linux.

- Opciones particulares de otros OS

Las siguientes opciones sirven para poder arrancar desde otro disco duro.



table=<device>

Especifica el dispositivo que contiene la tabla de partición del OS que se quiere arrancar.

map-drive=<bios-code> to=<new-bios-code>

Remapea los discos en la Bios, permitiendo arrancar desde otros discos distintos del de por defecto.

3 GRUB

¿Qué es Grub?

GNU GRUB es un gestor de arranque muy potente, capaz de arrancar un amplio abanico de sistemas operativos libres, así como sistemas operativos "en propiedad" a través del arranque en cadena (mecanismo para arrancar sistemas operativos no soportados a través de otro gestor de arranque. Se usa normalmente para arrancar DOS o Windows). GRUB está diseñado para hacer frente a la complejidad de arrancar un ordenador personal.

Una de las facetas importantes de GRUB es su flexibilidad; GRUB entiende sistemas de ficheros y formatos ejecutables del núcleo, así que te permite arrancar un sistema operativo cualquiera, de la manera que quieras, sin necesidad de saber la posición física del núcleo en el disco.

Así, por ejemplo, puedes arrancar el núcleo simplemente especificando el nombre del archivo y el disco (y la partición) donde se encuentra. Para que GRUB conozca estos datos, los puedes teclear a mano usando la interfaz de la línea de órdenes (mirar más adelante la sección La flexible interfaz de línea de órdenes), o usando la interfaz del menú (mirar más adelante la sección La sencilla interfaz de menú) a través de la cual puedes seleccionar fácilmente que SO arrancar. GRUB obtiene el menú de un fichero de configuración (mirar la sección de Configuración) que tú puedes crear para personalizar el menú. Hacemos notar que no sólo puedes entrar a la línea de órdenes cuando quieras, sino que también puedes editar entradas específicas del menú antes de usarlas.

Características de GRUB

La condición más importante que cumple GRUB es que obedece la *Especificación Multiarranque*, tal y como se

describe en section 'Motivation' in *The Multiboot Specification*.

El resto de los objetivos, en orden aproximado de importancia, son:

- Las funciones básicas han de ser claras y sencillas para el usuario final.
- Rica funcionalidad que cubra las necesidades de programadores y expertos del núcleo.
- Compatibilidad "hacia atrás" para poder arrancar FreeBSD, NetBSD, OpenBSD y Linux. Núcleos "en propiedad" (como los de DOS, Windows NT y OS/2) están soportados a través de la función de arranque en cadena.

Todos los núcleos se arrancan en más o menos el mismo estado que describe la Especificación Multiarranque, a excepción de los modos específicos de compatibilidad (como arranque en cadena o el formato *piggyback* de Linux). En la actualidad, sólo están soportados los núcleos que se cargan a la altura o por encima de 1 megabyte. Cualquier intento de carga por debajo de ese límite simplemente fallará de inmediato con un mensaje de error notificando el problema.

Además de los requerimientos ya mencionados, GRUB posee las siguientes características (Nótese que la Especificación de Multiarranque no exige todas las características soportadas por GRUB):

- Reconoce varios formatos ejecutables

Soporte para muchas de las variantes de *a.out* además de *ELF*. También carga las tablas de símbolos.

- Soporte para núcleos que no son Multiarranque

Soporte para varios de los núcleos libres de 32-bit que no se ajustan a la especificación Multiarranque (principalmente FreeBSD, NetBSD, OpenBSD y Linux). Admite asimismo el arranque en cadena de otros gestores.

- Carga de múltiples módulos

Admite totalmente la carga de múltiples módulos, una característica del Multiarranque.

- Carga de un archivo de configuración

Admite la configuración mediante un archivo simple de texto con órdenes predeterminados. También es posible cargar otro archivo de configuración de forma dinámica e insertar un archivo de configuración predeterminado en un archivo de imagen de GRUB. La lista de órdenes es un conjunto que incluye a las admitidas desde la línea de órdenes.

- Presenta una interfaz de menú

Dispone de una interfaz de menú que presenta las opciones de arranque preseleccionadas, con un tiempo de espera programable. No existe un límite establecido para el número de entradas en el menú, y la actual implementación tiene espacio para varios cientos.

- Tiene una interfaz de línea de órdenes flexible

Dispone de una línea de órdenes bastante flexible, a la que se puede acceder desde el menú, y que permite editar cualquiera de las órdenes preseleccionadas o escribir instrucciones de arranque desde cero. Si no existe un fichero de configuración, GRUB presenta la línea de órdenes.

La lista de órdenes es un subconjunto de los admitidos en los archivos de configuración. La edición de órdenes guarda similitud con la línea de órdenes de Bash (mirar sección 'Command Line Editing' in *Bash Features*), con terminación automática -mediante el tabulador **TAB**- de órdenes, dispositivos, particiones o archivos en un directorio, dependiendo del contexto.

- Soporte para varios sistemas de ficheros

Admite de manera transparente varios sistemas de ficheros, además de una útil notación explícita de lista de bloques. Actualmente están soportados los sistemas de ficheros *BSD FFS*, *DOS FAT16 and FAT32*, *Minix fs*, *Linux ext2fs*, *ReiserFS*, *JFS*, *XFS*, and *VSTafs*.

- Soporte para descompresión automática

Puede descomprimir archivos comprimidos con *gzip*. Esta función es a la vez automática y transparente de cara al usuario (es decir, todas las funciones operan sobre el contenido descomprimido de los archivos especificados). Esto reduce considerablemente el tamaño de los archivos y el tiempo de carga, una ventaja notable especialmente en disquetes.

Es posible imaginar la necesidad de cargar módulos del núcleo en estado comprimido, por lo que existe una orden diferente de carga de módulos para evitar que sean descomprimidos.

- Acceso a los datos en cualquiera de los dispositivos instalados

Admite la lectura de datos de cualquier disco duro o disquete detectado por la BIOS, independientemente de cual sea el dispositivo raíz.

- Independencia de las traducciones de geometría del disco

Al contrario que otros gestores de arranque, GRUB hace irrelevante la traducción particular del disco. Se puede convertir un disco que esté instalado y funcionando con una traducción a otra sin efectos contraproducentes o

cambios en la configuración de GRUB.

- Detecta toda la RAM instalada

GRUB puede por lo general encontrar toda la RAM instalada en un ordenador compatible PC, usando una técnica avanzada de consulta a la BIOS para encontrar todas las regiones de memoria. Como se describe en la Especificación Multiarranque (mirar sección 'Motivation' in *The Multiboot Specification*), no todos los núcleos hacen uso de esta información, pero GRUB la provee para aquellos que sí lo hacen.

- Admite el modo LBA (Dirección de Bloque Lógico)

En las llamadas al disco tradicionales (*modo CHS*) existe un problema de traducción de geometría, esto es, la BIOS no puede leer por encima de 1024 cilindros, lo que supone que el espacio accesible está limitado a 508 Mb en el peor de los casos, o a 8 Gb en el mejor. GRUB no puede resolver este problema de forma universal, ya que no existe una interfaz genérica que sea usada en todas las máquinas. Sin embargo, numerosos equipos recientes usan la nueva interfaz de Acceso de Bloque Lógico, modo (*LBA*). GRUB detecta automáticamente si el modo LBA está disponible, y hace uso de él en caso de que lo esté. En modo LBA, GRUB puede acceder a todo el disco.

- Soporte para arranque a través de la red

GRUB es básicamente un gestor de arranque basado en el disco, pero soporta también el arranque a través de red. Se pueden cargar imágenes de sistemas operativos a través de la red usando el protocolo *TFTP*.

- Soporte para terminales remotas

GRUB puede ser usado en ordenadores que no posean consola ya que admite terminales remotas, con lo que puede ser controlado desde otro equipo. Por el momento solo está implementado el soporte para terminales a través del puerto de serie.

Nomenclatura

La sintaxis de dispositivos usada en GRUB es algo diferente de la que probablemente hayas visto antes en tu(s) SO (s), y debes conocerla para poder especificar discos o particiones.

Observa los ejemplos y las explicaciones que siguen:

(fd0)



Para empezar, GRUB requiere que se encierre el nombre del dispositivo entre paréntesis '(' y ')'. Lo que 'fd' significa es que se trata de un disquete (floppy disk). El numero '0' es el número de la unidad, empezando a contar desde *cero*. La expresión significa que GRUB usará todo el disquete.

(hd0,1)

Aquí, 'hd' indica un disco duro (hard disk). El primer entero, '0' indica el número de la unidad, es decir, que se trata del primer disco duro, mientras que el segundo entero, '1', indica el número de la partición (o el número de rodaja (slice) en la terminología de BSD). De nuevo, observa que los números de partición se cuentan desde *cero* y no desde uno. La expresión representa la segunda partición de la primera unidad de disco duro. En este caso, GRUB usa una partición del disco y no el disco entero.

(hd0,4)

Esto indica la primera *partición lógica* de la primera unidad de disco. Observa que el número de partición para las particiones lógicas se empieza a contar desde '4', sin que importe el número de particiones primarias en el disco duro.

(hd1,a)

Esto indica la partición 'a' de BSD en el segundo disco duro. Si necesitas especificar el número de rodaja (slice) que se debería usar, puedes hacerlo como: '(hd1,0,a)'. Si se omite el número de "rodaja", GRUB busca la primera "rodaja" del pc que contiene una partición 'a' de BSD.

Por supuesto, para poder acceder a discos o particiones con GRUB, necesitas especificar el dispositivo mediante una instrucción, como 'root (fd0)' o 'unhide (hd0,2)'. Para ayudarte a averiguar qué número es una partición que quieres, las opciones de la línea de órdenes (mirar sección La flexible interfaz de línea de órdenes) auto-completan los argumentos. Esto quiere decir que, por ejemplo, sólo necesitas escribir 'root ' y pulsar el **tabulador** para que GRUB liste las unidades, particiones o nombres de archivo, por lo que debería resultar sencillo poder determinar el nombre de la partición que buscas incluso con un conocimiento mínimo de la sintaxis.

Advierte que GRUB *no* distingue entre IDE y SCSI - simplemente cuenta las unidades desde cero sin que importe su tipo. Por lo general, el número de una unidad IDE es menor que el de una SCSI, aunque esto no se cumpliría si cambias la secuencia de arranque intercambiando las unidades IDE y SCSI en la BIOS.

La pregunta ahora es: ¿Cómo se especifica un archivo? De nuevo, observa un ejemplo:



(hd0,0)/vmlinuz

Esto especifica un archivo llamado `vmlinuz` en la primera partición del primer disco duro. Nota que también se pueden auto completar los nombres de archivo.

Instalación

Para empezar, necesitas tener bien instalado GRUB en tu sistema, bien mediante un tar con las fuentes o mediante un paquete para tu Sistema Operativo.

Para poder usar GRUB debes instalarlo en tu disco. Existen dos formas de conseguirlo - usando la aplicación grub-install en un SO tipo UNIX, o mediante una segunda etapa nativa. Ambas son bastante similares, sin embargo, es posible que la aplicación escoja una unidad equivocada de la BIOS así que conviene ser precavido.

Asimismo, si instalas GRUB en un Sistema Operativo tipo UNIX, asegúrate de tener un disco de arranque de emergencia a mano, para poder rescatar el ordenador en caso de que, por algún motivo, el disco duro quedara inservible (no pudiera arrancarse).

GRUB viene con imágenes de arranque, que se instalan normalmente en el directorio `/src/share/grub/i386-pc`. Necesitas copiar los archivos ``stage1'`, ``stage2'` y ``*stage1_5'` al directorio ``/boot/grub'`. Llamamos aquí *directorio de imágenes* y *directorio de arranque* respectivamente al directorio donde se instalan las imágenes de GRUB y al directorio que GRUB usa para encontrarlas.

Crear un disquete de arranque de GRUB

Para crear un disco de arranque de GRUB, necesitas tomar los archivos ``stage1'` y ``stage2'` del directorio de imágenes, y escribirlos respectivamente en el primer y segundo bloque de un disquete.

Precaución: Este método destruye cualquier dato que esté guardado en el disquete.

En sistemas operativos tipo UNIX, esto se puede hacer con las órdenes siguientes:



```
# cd /usr/share/grub/i386-pc

# dd if=stage1 of=/dev/fd0 bs=512 count=1

1+0 records in

1+0 records out

# dd if=stage2 of=/dev/fd0 bs=512 seek=1

153+1 records in

153+1 records out

#
```

El nombre del dispositivo puede ser diferente. Consulta el manual de tu Sistema Operativo.

Instalar GRUB nativamente

Precaución: Instalar la primera etapa de GRUB de esta manera borrará el sector de arranque normal de un Sistema Operativo.

Actualmente, GRUB puede arrancar directamente GNU Mach, Linux, FreeBSD, NetBSD y OpenBSD, por lo que usarlo en un sector de arranque no debería dar problemas. Sin embargo, es buena idea en general, hacer una copia de seguridad del primer sector de la partición en la que vayas a instalar la primera etapa de GRUB. Si instalas GRUB en el primer sector del disco duro no es tan importante tomar precauciones, ya que es fácil restaurarlo (por ejemplo, usando `FDISK /MBR' desde DOS).

Instalar GRUB de manera nativa en el sistema es sin duda recomendable. Si decides hacerlo así, necesitarás hacer un disquete de arranque de GRUB y arrancar el ordenador con él.

Una vez empezado, GRUB presentará la interfaz de la línea de órdenes (mirar sección La flexible interfaz de línea de órdenes). Lo primero es fijar el directorio de arranque como *dispositivo raíz* de GRUB (el dispositivo raíz de GRUB no significa necesariamente la partición raíz de tu SO; Si necesitas especificar la partición raíz para tu SO, añade ese argumento en la orden kernel).

```
grub> root (hd0,0)
```

Si no estás seguro de cual es la partición que contiene los archivos, puedes usar la orden `find` de la manera siguiente:

```
grub> find /boot/grub/stage1
```

Esto busca el archivo `/boot/grub/stage1` y lista los dispositivos que lo contienen.

Una vez que has fijado correctamente el directorio raíz, utiliza la orden `setup`:

```
grub> setup (hd0)
```

Esta orden instala GRUB en el MBR de la primera unidad. Si lo que quieres es instalar GRUB en el *sector de arranque* de una partición en vez de en el MBR, especifica la partición en la que quieres instalar GRUB:

```
grub> setup (hd0,0)
```

Si instalas GRUB en una partición o en una unidad que no sea la primera, tendrás que arrancar en cadena a GRUB desde otro gestor de arranque. Mira en el manual del gestor de arranque en particular la manera de arrancar GRUB en cadena.

Ahora ya puedes arrancar GRUB sin usar un disquete. Lee la sección de Arranque para aprender a arrancar tus sistemas operativos desde GRUB.

Instalar GRUB usando grub-install

Precaución: No se aconseja utilizar este procedimiento ya que hay posibilidades de que deje el ordenador sin poder arrancar. Por ejemplo, la mayoría de los sistemas operativos no comunican a GRUB la manera de corresponder correctamente unidades de la BIOS a dispositivos del Sistema Operativo, GRUB simplemente *adivina* la correspondencia. Esto funciona en la mayoría de los casos, pero no siempre, por lo que GRUB pone a tu disposición un fichero de correspondencias definido por el usuario *device map* que debes reparar si fuera incorrecto.

A pesar de esto, si quieres instalar GRUB bajo un Sistema Operativo tipo UNIX (como GNU), ejecuta el programa

grub-install (mirar sección Instalar Grub install) como superusuario (*root*).

Su uso es muy sencillo. Solo necesitas pasarle un argumento al programa: donde quieres instalar GRUB. El argumento puede ser un fichero de dispositivo o una unidad/partición de GRUB. Por ejemplo, para instalar GRUB en el MBR del primer disco IDE desde Linux puedes usar:

```
# grub-install /dev/hda
```

Bajo GNU/Hurd, la siguiente orden tiene el mismo efecto:

```
# grub-install /dev/hd0
```

Si se trata de la primera unidad en la BIOS, ésta orden es equivalente:

```
# grub-install '(hd0)'
```

Los ejemplos anteriores asumen que usas imágenes de GRUB situadas en el directorio raíz. Si quieres que GRUB utilice imágenes en otro directorio, necesitas especificar la opción ``root-directory'`. Un caso típico es cuando quieres crear un disquete de arranque de GRUB con sistema de ficheros. Un ejemplo:

```
# mke2fs /dev/fd0
```

```
# mount -t ext2 /dev/fd0 /mnt
```

```
# grub-install --root-directory=/mnt '(fd0)'
```

```
# umount /mnt
```

Otro ejemplo es el caso en que tienes una partición de arranque distinta montada sobre ``/boot'`. GRUB es un gestor de arranque, y no sabe nada de puntos de montaje. Por lo tanto, la manera de invocar grub-install sería:

```
# grub-install --root-directory=/boot /dev/hda
```

Por cierto, como observamos anteriormente, es bastante difícil averiguar las unidades de la BIOS desde un Sistema Operativo tipo UNIX. Por este motivo, grub-install te pedirá que compruebes que ciertamente acertó con las correspondencias una vez terminada la instalación. Si el resultado fuera incorrecto, es bastante difícil que el

ordenador pueda arrancar sin problemas.

Advierte que grub-install es en realidad un fichero de órdenes (shell script), y el verdadero trabajo lo realiza el intérprete de órdenes grub (mirar sección invocar el intérprete Grub). Podrías por lo tanto ejecutar directamente grub para instalar GRUB, sin utilizar grub-install. Sin embargo, esto no es recomendable a no ser que conozcas bien el funcionamiento de GRUB. Instalar un gestor de arranque desde un Sistema Operativo en ejecución puede ser muy peligroso.

Arranque

GRUB puede arrancar de manera similar núcleos que se acojan al Multiarranque, pero para algunos sistemas operativos libres es necesario utilizar algo de magia adecuada al SO.

Como arrancar sistemas operativos

GRUB tiene dos maneras distintas de arrancar sistemas. Una es cargar el sistema operativo directamente, y la otra es cargar en cadena otro gestor de arranque que se encargará de arrancar el sistema operativo. En general, el primer método es el más deseable, porque no requiere que instales o mantengas otros gestores de arranque, y GRUB es lo bastante flexible para cargar un sistema operativo en cualquier disco o partición. Sin embargo, resulta necesario algunas veces utilizar el segundo método ya que GRUB no admite de forma nativa todos los sistemas operativos existentes.

- Como arrancar directamente un Sistemas Operativos con GRUB:

El Multiarranque es el formato nativo soportado por GRUB. Por conveniencia, existe también soporte para Linux, FreeBSD, NetBSD y OpenBSD. Para el resto de los sistemas operativos que quieras arrancar debes utilizar el arranque en cadena.

En general, GRUB arranca cualquier Sistema Operativo que se acoja al Multiarranque con los pasos siguientes:

- Fija el dispositivo raíz de GRUB en la unidad que contiene las imágenes del SO mediante la orden root.
- Carga la imagen del núcleo con la orden kernel.
- Si necesitas módulos, cárgalos con la orden module o modulenounzip.
- Ejecuta la orden boot.



Linux, FreeBSD, NetBSD y OpenBSD se pueden arrancar de manera similar. Puedes cargar la imagen del núcleo mediante la orden `kernel` y después usar la orden `boot`. Si el núcleo requiere algún parámetro, simplemente añade los parámetros a `kernel`, detrás del nombre del archivo del núcleo.

- Carga otro gestor de arranque para arrancar SOs no soportados

Si quieres arrancar un sistema operativo que no esté soportado (por ejemplo Windows 95), debes arrancar en cadena un gestor de arranque propio de ese sistema operativo. Normalmente, el gestor de arranque está insertado en el *sector de arranque* de la partición en la que está instalado el sistema operativo.

- Fija el dispositivo raíz de GRUB en la partición mediante la orden `rootnoverify`:

```
grub> rootnoverify (hd0,0)
```

- Marca la partición como *activa* mediante la orden `makeactive` (Esto no es necesario para la mayoría de sistemas operativos modernos):

```
grub> makeactive
```

- Carga el gestor de arranque usando la orden `chainloader`:

```
grub> chainloader +1
```

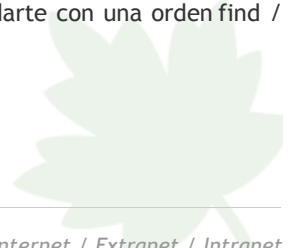
`+1' indica que GRUB debe leer un sector desde el principio de la partición.

- Ejecuta la orden `boot`.

GNU/Hurd

Debido a que GNU/Hurd se acoge a la especificación Multiarranque resulta sencillo arrcarlo; no hay nada especial en hacerlo. Pero no olvides que tienes que especificar una partición raíz al núcleo.

- Fija el dispositivo raíz de GRUB en la misma unidad que la de GNU/Hurd. Puedes ayudarte con una orden `find /boot/gnumach` o similar.



- Carga el núcleo y el módulo de esta manera:

```
grub> kernel /boot/gnumach root=hd0s1
```

```
grub> module /boot/serverboot
```

- Ejecuta la orden boot.

GNU/Linux

Resulta relativamente sencillo arrancar GNU/Linux desde GRUB, ya que se asemeja de alguna manera a un SO Multiarranque.

- Fija el dispositivo raíz de GRUB en el mismo dispositivo que GNU/Linux. Puedes ayudarte de una orden como `find /vmlinuz` o similar.
- Carga el núcleo:

```
grub> kernel /vmlinuz root=/dev/hda1
```

Si necesitas especificar algún otro parámetro del núcleo añádelos sin más a la orden. Por ejemplo, para poner la `'vga'` en `'ext'` usa:

```
grub> kernel /vmlinuz root=/dev/hda1 vga=ext
```

Consulta la documentación en el árbol del código fuente del núcleo de Linux para la información completa de todas las opciones disponibles.

Si usas un `initrd`, ejecuta la orden `initrd` después de la orden `kernel`:

```
grub> initrd /initrd
```

- Para terminar, ejecuta la orden boot.



Precaución: Si usas un `initrd` y especificas la opción ``mem='` al núcleo, para lograr que use una cantidad de memoria menor que la existente necesitas especificar a GRUB el mismo tamaño de memoria. Para comunicar el tamaño a GRUB, ejecuta la orden `uppermem` *antes* de cargar el núcleo.

Configuración

Como habrás notado, es necesario que escribas unas cuantas órdenes para arrancar un Sistema Operativo. Hay una manera más rápida de hacerlo: GRUB posee una interfaz de menú desde la que puedes seleccionar una entrada (usando las flechas del teclado) que hará todo lo necesario para arrancar un Sistema Operativo.

Para habilitar el menú necesitas un archivo de configuración, ``menu.lst'` en el directorio de arranque. Vamos a analizar un archivo de ejemplo.

El fichero contiene al principio varias opciones generales relativas a la interfaz de menú. Puedes poner estas opciones antes de las entradas que empiezan con la orden `title`.

```
#
```

```
# Ejemplo de archivo de configuración del menú
```

```
#
```

como habrás adivinado, estas líneas son comentarios. GRUB ignorará cualquier línea que empiece con el caracter de almohadilla (``#`), así como líneas en blanco.

```
# Arrancar la primera entrada por defecto
```

```
default 0
```

La primera entrada (contando a partir del número cero, no uno) será la elección por defecto.

```
# Arrancar automáticamente despues de 30 segundos
```

```
timeout 30
```



Como indica el comentario, GRUB arrancará automáticamente en 30 segundos, a no ser que sea interrumpido por la presión de una tecla.

```
# Descolgarse a la segunda entrada
```

```
fallback 1
```

Si por alguna razón la entrada por defecto no funcionara, intentar la segunda (esta opción rara vez se usa, por razones obvias).

Pasamos ahora a las definiciones de Sistema Operativo. Verás que todas las entradas empiezan con la orden especial `title`, y que las instrucciones se encuentran a continuación. Observa que no hay una orden `boot` al final de cada entrada. Esto se debe a que GRUB ejecuta automáticamente `boot` si ha cargado con éxito el resto de órdenes.

El argumento de la orden `title` se utiliza a modo de título o descripción de la entrada del menú. La orden `title` presenta el argumento tal cual, por lo que puedes escribir básicamente lo que quieras.

```
# Para arrancar GNU/Hurd
```

```
title GNU/Hurd
```

```
root (hd0,0)
```

```
kernel /boot/gnumach.gz root=hd0s1
```

```
module /boot/serverboot.gz
```

Esto arranca GNU/Hurd en el primer disco duro.

```
# Para arrancar GNU/Linux
```

```
title GNU/Linux
```

```
kernel (hd1,0)/vmlinuz root=/dev/hdb1
```

Esto arranca GNU/Linux, pero en el segundo disco duro.



```
# Para arrancar WindowsNT o Windows95
```

```
title Windows NT / Windows 95 boot menu
```

```
root      (hd0,0)
```

```
makeactive
```

```
chainloader +1
```

```
# Para cargar DOS si WindowsNT está instalado
```

```
# chainload /bootsect.dos
```

Lo mismo que el anterior, pero para Windows.

```
# Para instalar GRUB en el disco duro
```

```
title Install GRUB into the hard disk
```

```
root      (hd0,0)
```

```
setup      (hd0)
```

Esto simplemente (re)instala GRUB en el disco duro.

```
# cambia los colores
```

```
title Change the colors
```

```
color light-green/brown blink-red/blue
```

En la última entrada, se usa la orden color para cambiar los colores del menú. Esta orden es algo especial, ya que se puede usar en el menú o en la línea de órdenes.



Interfaz de usuario de GRUB

GRUB posee tanto una simple interfaz de menú donde escoger entradas ya preparadas en un archivo de configuración como una línea de órdenes flexible para ejecutar cualquier combinación de órdenes de arranque que se desee.

Tan pronto como se carga GRUB, éste busca el archivo de configuración. Si encuentra uno, activa la interfaz de menú utilizando las entradas que se encuentren en el archivo. Si utilizas la opción del menú de *línea de órdenes* o si GRUB no encuentra un archivo de configuración, se activa la línea de órdenes.

La flexible interfaz de línea de órdenes

La interfaz de la línea de órdenes consiste, como en Unix o DOS, de un cursor (prompt) con espacio a continuación para introducir texto. GRUB ejecuta las órdenes inmediatamente cuando son introducidas. Las órdenes son un subconjunto de las disponibles en el fichero de configuración, y se usan exactamente con la misma sintaxis.

La edición de texto y el movimiento del cursor se puede hacer a través de un subconjunto de las funciones disponibles en el intérprete de órdenes Bash:

C-f

Flecha derecha

Moverse hacia delante un caracter.

C-b

Flecha izquierda

Moverse hacia atrás un caracter

C-a

INICIO

Moverse al inicio de la línea.

C-e



END

Moverse al final de la línea.

C-d

DEL

Borrar el caracter bajo el cursor.

C-h

BS

Borrar el caracter a la izquierda del cursor.

C-k

Suprimir el texto desde la posición actual del cursor al final de la línea.

C-u

Suprimir hacia atrás, desde el cursor hasta el principio de la línea.

C-y

Recuperar el texto suprimido en la posición del cursor.

C-p

Flecha arriba

Moverse hacia arriba en la historia de órdenes.

C-n

Flecha abajo

Moverse hacia abajo en la historia de órdenes.



Cuando escribes órdenes de forma interactiva, si el cursor está en medio o antes de la primera palabra en la línea de órdenes, presionar el **tabulador** (o **C-i**) te mostrará una lista de las órdenes disponibles, y si el cursor está detrás de la primera palabra, el **tabulador** mostrará una lista de discos, particiones o nombres de archivos, dependiendo del contexto.

Nota que no puedes utilizar la función de auto completar en el sistema de ficheros de TFTP. Esto se debe a que TFTP no implementa el listado de ficheros por motivos de seguridad.

La sencilla interfaz de menú

La interfaz de menú es muy sencilla de usar. Las órdenes son razonablemente intuitivos además de estar descritos en pantalla.

Básicamente, la interfaz de menú presenta una lista de *opciones de arranque* entre las que el usuario puede escoger. Usa las flechas del teclado para seleccionar la entrada que quieras, y pulsa **RETORNO** para ejecutarla. Opcionalmente, al concluir el tiempo de espera, se arranca la entrada seleccionada por defecto (la primera si no se ha especificado una), a no ser que se pulse una tecla.

Se pueden introducir órdenes pulsando la tecla **c**, lo que presenta una línea de órdenes (que opera exactamente igual que en la versión de GRUB sin fichero de configuración, con la excepción de que te permite volver al menú pulsando **ESC**). También se pueden editar las *entradas de arranque* pulsando la tecla **e**.

Si has protegido la interfaz de menú mediante una contraseña, todo lo que podrás hacer será escoger una entrada pulsando **RETORNO**, o pulsar **p** para introducir la contraseña

Editar una entrada del menú

El editor de entradas del menú se parece bastante a la interfaz de menú principal, pero en las líneas del editor verás las órdenes de la entrada seleccionada en lugar de los nombres de las distintas entradas.

Si se pulsa **ESCAPE** en el editor, se descartan todos los cambios hechos en la entrada seleccionada, volviendo a la pantalla del menú principal.

Cuando se selecciona una línea en particular, el editor coloca al usuario en una versión especial de la línea de órdenes de GRUB para que edite la línea. Cuando el usuario pulsa **RETURN**, GRUB reemplaza la línea en cuestión en la

opción de arranque con los nuevos cambios (a no ser que se aborte a través de la tecla **ESCAPE**, en cuyo caso se descartan los cambios).

Si quieres añadir una línea nueva a la entrada del menú, pulsa **o** para añadirla debajo de la línea sobre la que te encuentres o **O** para añadirla sobre ella.

Para borrar una línea pulsa la tecla **d**. Aunque desafortunadamente GRUB no implementa la opción de *deshacer*, se obtiene prácticamente el mismo resultado retornando al menú principal.

La interfaz de menú oculta

En el caso de utilizar una terminal tipo "dumb" o de haber solicitado explícitamente a GRUB que esconda el menú mediante la orden `hiddenmenu`, GRUB no presentará la interfaz de menú, y arrancará automáticamente la opción por defecto, a no ser que se interrumpa pulsando **ESCAPE**.

Cuando se interrumpe la cuenta atrás del tiempo de espera y la terminal es del tipo "dumb", GRUB retorna a la interfaz de la línea de órdenes.

La lista de órdenes sólo para el menú

Las reglas para componer un archivo de configuración son las siguientes:

- Deben ir primero las órdenes específicas del menú.
- Los archivos *deben* estar en formato de texto simple.
- Los comentarios en un archivo de configuración se indican con una ``#` en el principio de la línea.
- Las opciones van separadas por espacios.
- Todos los números pueden especificarse en sistema decimal o hexadecimal. Los números hexadecimales deben ir precedidos por ``0x'`, sin importar que se usen mayúsculas o minúsculas.
- Se ignoran las opciones extra o el texto al final de la línea, a no ser que se especifique lo contrario.

- Las órdenes que no se reconocen se añaden a la opción actual, excepto si están antes de que empiecen las opciones, donde son ignoradas.

Las órdenes siguientes solo pueden usarse en el menú:

default	Fija la entrada por defecto
fallback	Fija la entrada de repuesto
hiddenmenu	Esconde la interfaz de menú
timeout	Pone un tiempo de espera
title	Encabeza una entrada del menú

Invocar el intérprete grub

Observa que el intérprete de grub es un emulador; no funciona bajo el sistema nativo, por lo que a veces puede equivocarse. Por lo tanto no deberías fiarte de él demasiado. Si hubiera cualquier problema con él, no dudes en usar un sistema GRUB nativo, especialmente si no acierta con la correspondencia entre unidades de la BIOS y dispositivos del Sistema Operativo.

Introducción al intérprete de grub

Puedes usar la orden grub para instalar GRUB bajo tus sistemas operativos y como campo de pruebas cuando añadas una nueva característica en GRUB o cuando corrijas un fallo. La emulación se realiza reemplazando las llamadas de la BIOS con llamadas de sistema de UNIX y funciones de libc.

La orden grub acepta las opciones siguientes:

`--help`

Imprime un sumario de las opciones de la línea de órdenes.

`--version`

Imprime el número de versión.

`--verbose`



Imprime mensajes más detallados para quien desee eliminar posibles fallos.

``--device-map=archivo'`

Usar el fichero *archivo* como mapa de dispositivos. El formato se describe en La correspondencia entre unidades de la BIOS y dispositivos del SO.

``--no-floppy'`

No hacer comprobación de las unidades de disquete. Esta opción no tiene efecto si se ha especificado la opción ``--device-map'` (mirar sección La correspondencia entre unidades de la BIOS y dispositivos del SO).

``--probe-second-floppy'`

Comprueba la segunda unidad de disquete. Si no se especifica esta opción el intérprete de grub no la comprueba, ya que a veces lleva bastante tiempo. Si especificas un archivo de mapa de dispositivos (mirar sección La correspondencia entre unidades de la BIOS y dispositivos del SO) el intérprete grub ignora esta opción.

``--config-file=archivo'`

Lee el archivo de configuración *archivo* en lugar de `` /boot/grub/menu.lst'`. El formato es el mismo que el de un archivo normal.

``--boot-drive=unidad'`

Fija el *dispositivo de arranque* de la segunda etapa en *unidad*. Este argumento debería ser un entero (decimal, octal o hexadecimal).

``--install-partition=par'`

Fija la *particion de instalación* de segunda etapa en *par*. El argumento debe ser un entero (decimal, octal o hexadecimal).

``--no-config-file'`

No usar el archivo de configuración aunque se pueda leer.

``--no-curses'`

No usar la interfaz de "curses" incluso aunque esté disponible.



`--batch`

Esta opción equivale a `--no-config-file --no-curses`.

`--read-only`

Deshabilita la escritura a cualquier disco.

`--hold`

Esperar a que se adjunte un debugger. Esta opción es útil cuando se desea eliminar errores del código de inicio.

Como instalar GRUB mediante grub

El procedimiento de instalación es el mismo que bajo una segunda etapa *nativa*. mirar sección Instalación para más información. Aquí se describe la información específica de la orden grub.

Con lo que debes tener cuidado es con el *buffer cache*. grub utiliza los dispositivos directamente (raw) en vez de los sistemas de ficheros que proveen tus sistemas operativos, por lo que existe el problema potencial de que inconsistencias del caché puedan corromper tu sistema de ficheros. Lo que se recomienda es:

- Si es posible, desmonta antes de ejecutar grub las unidades en las que GRUB pueda escribir datos.
- Si una unidad no se puede desmontar pero se puede montar con la etiqueta sólo de escritura, hazlo así. Eso debería ser seguro.
- Si una unidad debe ser montada con la etiqueta de escritura-lectura, asegúrate de que no se está llevando a cabo ninguna actividad sobre ella mientras se ejecuta la orden grub.
- Reinicia tu sistema operativo tan pronto como sea posible. Es probable que esto no sea necesario si sigues las reglas anteriores, pero reiniciar es el método más seguro.

Además de esto, introduce la orden quit cuando acabes la instalación. Esto es *muy importante* porque quit hace consistente el buffer caché. No pulses C-c.

Si quieres instalar GRUB de manera no interactiva, especifica la opción `--batch` en la línea de órdenes. Aquí tienes un ejemplo sencillo:

```
#!/bin/sh

# Usa /usr/sbin/grub si estás en un sistema antiguo.

/sbin/grub --batch <<EOT 1>/dev/null 2>/dev/null

root (hd0,0)

setup (hd0)

quit

EOT
```

La correspondencia entre unidades de la BIOS y dispositivos del SO

Cuando especificas la opción `--device-map`, el intérprete de grub crea automáticamente el *fichero del mapa de dispositivos* a no ser que ya exista. El nombre de archivo preferido es `/boot/grub/device.map`.

Si el fichero del mapa de dispositivos ya existe, el intérprete de grub lo lee para corresponder unidades de la BIOS a dispositivos del Sistema Operativo. El archivo consiste en líneas como esta:

dispositivo archivo

dispositivo es una unidad, con la sintaxis propia de GRUB, y *archivo* es un fichero del del Sistema Operativo, normalmente un fichero de dispositivo.

La razón por la que el intérprete de grub te da un fichero de dispositivos es que en algunos contextos no es capaz de adivinar la correspondencia de unidades de la BIOS a dispositivos del SO correctamente. Por ejemplo, si intercambias la secuencia de arranque entre IDE y SCSI en la BIOS, el intérprete equivoca el orden.

Así, puedes editar el archivo si el intérprete de grub comete algún error. Puedes poner los comentarios que necesites en el archivo, ya que el intérprete de grub asume que una línea es sólo un comentario si el primer carácter es `#`.



Invocar grub-install

El programa grub-install instala GRUB en tu disco mediante el intérprete de grub. Debes especificar el nombre del dispositivo en el que quieres instalar GRUB, de la siguiente manera:

```
grub-install disp_instalación
```

El nombre del dispositivo *disp_instalación* es un nombre de dispositivo del sistema operativo o de GRUB.

grub-install acepta las opciones siguientes:

`--help`

Imprime un sumario de las opciones de la línea de órdenes.

`--version`

Imprime el número de version de GRUB.

`--force-lba`

Fuerza a GRUB a usar el modo LBA incluso con una BIOS defectuosa. Usa esta opción sólo si tu BIOS no funciona en modo LBA aunque sí soporta este modo.

`--root-directory=dir`

Instala las imágenes de GRUB bajo el directorio *dir* en vez del directorio raíz. Esta opción es útil cuando quieres instalar GRUB en una partición aparte o en un disco extraíble. Aquí tienes un ejemplo donde existe una partición de arranque aparte montada sobre `/boot`:

```
grub-install --root-directory=/boot '(hd0)'
```

`--grub-shell=archivo`

Usa *archivo* como el intérprete de grub. Puedes añadir las opciones que quieres a *archivo* detrás del nombre del archivo, como sigue:



```
grub-install --grub-shell="grub --read-only" /dev/fd0
```

```
`--recheck`
```

Volver a comprobar el mapa de dispositivos, incluso si ya existe el fichero `/boot/grub/device.map`. Deberías usar esta opción cada vez que añadas o quites un disco del ordenador.

