

Detalles GRUBby (Versión Legacy)

BOTAS SUCIAS

Milica Vajphiko, 123RF

La versión antigua del Gran Cargador de Arranque Unificado, GRUB, aún arranca la mayoría de las distribuciones Linux.

POR BRUCE BYFIELD

Cuando arrancamos un ordenador GNU/Linux, una de las primeras cosas que vemos es un cargador de arranque que nos permite elegir qué kernel o sistema operativo usar. Actualmente, el cargador de arranque que aparece casi siempre es GNU GRUB (GRand Unified Bootloader).

GRUB existe actualmente en dos versiones: la antigua GRUB, que ha sido usada durante más de una década, y GRUB 2, que en estos momentos se halla en sus últimas fases de desarrollo. Sin embargo, a excepción de Ubuntu 9.10, la mayoría de las distribuciones más importantes están usando aún la versión antigua de GRUB, ya que es fácil de editar manualmente. Algunas distribuciones, especialmente las más pequeñas, sospecho que continuarán utilizando esta versión de GRUB por el motivo anterior. Por tanto, en este artículo hablaremos de la versión antigua y dejaré GRUB 2 para el número que viene.

Como su nombre indica, GRUB está diseñado para ser un cargador de arranque flexible, capaz de iniciar una amplia variedad de sistemas operativos desde una igualmente extensa variedad de dispositivos internos y externos y sistemas de ficheros. Esta flexibili-

dad es la principal razón por la que GRUB es el cargador de arranque más popular en el software libre. No importa la distribución ni las complejidades de nuestra configuración, GRUB las controla de manera inquebrantable, con extraordinariamente pocos problemas.

Los primeros 512 bytes de cualquier dispositivo o partición arrancable se conocen como el master boot record (MBR), y contiene el cargador de arranque principal junto con la tabla de las particiones de dispositivos. GRUB funciona sustituyendo el cargador de arranque principal con su Stage 1. A menudo, GRUB se encuentra instalado en la primera partición de un disco duro, aunque también podemos instalarlo en una partición diferente.

Como MBR es demasiado pequeño para casi cualquier cosa, Stage 1 suele señalar a Stage 2, el cual se encuentra localizado fuera del MBR y funciona con ficheros de configuración en `/boot/grub/` para darnos un menú de los diferentes kernels y sistemas operativos configurados que podemos usar.

A veces también se utiliza un Stage 1.5 opcional, dependiendo de los sistemas operativos y de los sistemas de ficheros involu-

crados. Por ejemplo, FreeBSD usa Stage 1.5 de manera rutinaria, al igual que los sistemas de ficheros JFS y XFS.

Una vez que hacemos una selección en Stage 2 (o el predeterminado que comienza una vez que expira la pausa), GRUB inicia nuestra selección por sí mismo o bien pasa el control a otro cargador de arranque, como el usado por varias versiones de Windows. Como Stage 2 no necesita ser metido en el MBR, GRUB puede ser bastante más versátil que la mayoría de los cargadores de arranque y puede permitir más opciones.

Para configurar los detalles de cómo funciona GRUB, usamos una línea de comandos – la del menú de Stage 2 o la del sistema operativo en sí – o editamos el fichero de configuración en un editor de textos. Ambos métodos requieren un poco de conocimiento (aunque algunas distribuciones como Debian añaden comentarios extensivos a `menu.lst` para guiarnos), pero ambos son también relativamente seguros con tal de que se tomen algunas precauciones elementales, tales como hacer copias de seguridad de ficheros esenciales y manteniendo siempre al menos un elemento de menú arrancable en todo momento.

Nombres de Particiones en GRUB

Antes de editar GRUB necesitamos conocer el sistema de nombres que usa para los dis-

positivos arrancables. El sistema de nombres es diferente del de GNU/Linux o Windows, lo cual puede prestarse a confusión, aunque, afortunadamente, es bastante simple de utilizar.

En GRUB, cada partición está descrita por cinco caracteres encerrados entre paréntesis. Los dos primeros son minúsculas y describen el tipo de dispositivo, como *hd* para hard drive o disco duro o *cd* para una unidad de CD. El tercer carácter es el número del dispositivo, que comienza por 0 ó 1, seguido de una coma, y luego el número de la partición en el disco: 0-3 para una partición principal y 4 o superior para particiones extendidas.

Así, (*hd0,1*) hace referencia a la segunda partición primaria en el primer disco duro, mientras que (*hd1,5*) se refiere a la segunda partición extendida en el segundo disco duro. Las particiones lógicas que podemos ver cuando usamos Fdisk o GNU Parted no están incluidas en los cálculos de GRUB.

Si encuentra esto confuso, es normal. Después de todo, si estamos ejecutando GNU/Linux y Windows en la misma máquina y usamos Parted para una referencia, tendríamos cuatro anotaciones diferentes para identificar el mismo número de dispositivo. Para evitar problemas, la primera vez que usamos GRUB es buena idea crear una tabla de equivalentes (como se muestra, por ejemplo, en la Tabla 1) para ayudarnos a evitar cometer errores. En caso contrario, podríamos introducir fácilmente el disco erróneo cuando editamos.

Comportamiento Básico de la Edición de GRUB

El fichero de configuración de GRUB es */boot/grub/menu.lst*. Al igual que muchos ficheros de configuración, podemos editar *menu.lst* abriéndolo en un editor de textos como usuario root.

El fichero *menu.lst* está dividido en dos secciones. La primera sección establece las funcionalidades generales de GRUB (Figura 1). La otra define las características de cada sistema operativo o kernel que podemos arrancar. Sin embargo, también pueden servir como separadoras. Por ejemplo, si tenemos arranque dual Debian y Windows, Debian añade una sección con el título *Other operating systems*: antes de la sección de Windows. En todas las secciones, un campo está separado habitualmente de su entrada mediante una etiqueta, pero, para hacer la edición más con-

Tabla 1: Notaciones de Discos

Disco Duro y Partición	GNU/Linux	GRUB	Windows
Primer disco duro, primera partición	dev/hda1	(hd0,0)	C
Primer disco duro, segunda partición	dev/hda2	(hd0,1)	D
Primer disco duro, primera partición extendida	/dev/hda5	(hd0,4)	Varios
Segundo disco duro, primera partición	/dev/hdb1	(hd1,0)	Varios

fusa, algunos campos usan un signo igual después del campo.

Muchas de las posibles entradas en la primera sección son opcionales, pero pueden ayudarnos a personalizar el comportamiento de GRUB. Otras pueden añadirse para casos especiales, como el uso de hipervisores Xen.

Quizás la más comúnmente utilizada es *timeout <segundos>*, la cual establece el número de segundos antes de que se arranque la selección predeterminada si no hacemos otra elección. La selección predeterminada es o bien el sistema operativo o bien el kernel definido en la segunda sección o el definido como *default <sección>*, el cual enumera las secciones desde 0, comenzando con la segunda sección.

Otra opción de la primera sección extensamente utilizada es *howmany = <número>*, que configura el número de entradas que pueden añadirse a *menu.lst* cuando un proceso lo edita automáticamente, como puede ser cuando se añade un kernel empaquetado. Este campo es importante cuando tenemos el directorio */boot* en una partición separada con espacio limitado. Sin embargo, *howmany* puede ser peligroso si se establece demasiado bajo: si algo va mal, nos puede dejar sin un kernel arrancable.

También podemos asegurar GRUB asignando una contraseña en la primera sección y

añadiendo el campo *lock* a las otras secciones. Para añadir una contraseña, primero ejecutamos */sbin/grub-md5-crypt* para cifrar una contraseña. Luego añadimos a la primera sección *password —md5 <hash de la contraseña>*. Aunque podemos usar *password* con una contraseña no cifrada, si lo hacemos, debemos cambiar los permisos de *menu.lst* de modo que sólo el usuario root pueda leerlo.

Otras opciones de primera sección son estéticas. Para personalizar el menú con un número limitado de colores usamos *color <fondo normal>/<primer plano normal>* *<fondo resaltado>/<primer plano resaltado>* [1]. Esta opción es popular con distribuciones que quieren resaltar GRUB con sus propios colores – o al menos con unos colores que se les parecen. Podemos también añadir un mensaje creando un fichero de texto y apuntando a él especificando la partición y directorio – por ejemplo, *gfx-menu (hd0,0) /boot/grub/message*.

Si realmente deseamos personalizar la apariencia de GRUB, podemos añadir una imagen de arranque. Muchas distribuciones incluyen un paquete de imágenes de arranque, aunque también podemos crear las nuestras propias en formato *.xpm.gz* en GIMP con dimensiones de 640x480 píxeles y no más de 16 colores. Luego podemos apuntar GRUB a la imagen con un comando como *splashimage = (hd0,0)/boot/grub/message/splash.xpm.gz*, modificándolo según las circunstancias específicas.

Edición de Selecciones de GRUBs

Las otras secciones en *menu.lst* definen una entrada de modo que GRUB pueda usarla para arrancar nuestro ordenador. Típicamente, un kernel de Linux o una distribución tendrán cuatro campos (Figura 2):

- *title*: Cómo aparece la selección en el menú de GRUB. Normalmente, el título incluye el nombre de la distribución y el kernel que está siendo usado.
- *root*: Partición en la que los ficheros de configuración de GRUB se encuentran instalados.
- *kernel*: Donde encontrar el kernel que se ha de arrancar (normalmente en */boot*). Si

```
default 0
timeout 5
color cyan/blue white/blue
password --md5 $1$gLhU0/$aW78kHK1QfV3P2b2znUoe/
```

Figura 1: La primera sección del menu.lst de GRUB contiene opciones generales para la presentación del menú de arranque.

```
title Debian GNU/Linux, kernel 2.6.30-2-686-bigmem
root (hd0,1)
kernel /boot/vmlinuz-2.6.30-2-686-bigmem root=/dev/sda2 ro quiet
initrd /boot/initrd.img-2.6.30-2-686-bigmem
```

Figura 2: Para arrancar Linux, GRUB requiere un mínimo de cuatro campos en una sección.

```
title Microsoft Windows XP Home Edition
root (hd0,0)
savedefault
makeactive
chainloader +1
```

Figura 3: Las secciones de Windows en menu.lst requieren un juego de campos especiales, porque GRUB no puede arrancar este sistema directamente.

el kernel se encuentra en otra partición distinta en la que están los ficheros de configuración de GRUB, la partición precede a la ruta del directorio. El kernel puede ser modificado por cualquier opción del kernel. Las entradas suelen añadir *ro* (read-only) y *quiet* al final. A menudo aparece una entrada separada para el mismo kernel en el GRUB con la opción del kernel *single*, la cual arranca un sistema de línea de comandos básico que el usuario *root* puede usar para reparaciones.

- *inited*: Donde encontramos los sistemas de ficheros temporales usados durante el arranque. Su ruta es habitualmente la misma que la del kernel.

Para una sección Windows (Figura 3), los dos primeros campos son *title* y *root*, igual que para GNU/Linux, luego:

- *savedefault*: Conserva la sección predeterminada original y evita que la entrada de Windows se convierta en predeterminada. Si Windows es la entrada por defecto, podemos omitir este campo.
- *makeactive*: Establece la partición Windows como activa.
- *chainloader +1*: Le dice a GRUB que transfiera el control al cargador de arranque de Windows en el primer sector de la partición especificada como *root*.

Para añadir una sección como separador, todo cuanto se necesita es un título.

Además de *lock*, el cual se usa cuando GRUB está protegido por contraseñas, pueden añadirse docenas de opciones distintas a una sección según se requiera [2]. Para diferentes sistemas operativos, también debemos especificar un conjunto específico de campos [3]. Sin embargo, en la mayoría de los casos, estas instrucciones serán suficientes.

Edición desde la Línea de Comandos de GRUB

En lugar de editar *menu.lst*, también podemos utilizar la línea de comandos de GRUB (Figura 4). La mayoría de la gente usa la línea de comandos de GRUB principalmente para rescatar operaciones, aunque

```
GNU GRUB version 0.97 (640K lower / 3072K upper memory)

[ Minimal BASH-like line editing is supported. For
the first word, TAB lists possible command
completions. Anywhere else TAB lists the possible
completions of a device/filename. ]

grub> |
```

Figura 4: Podemos iniciar GRUB en el menú de arranque o desde la línea de comandos como usuario *root*.

también es útil para comprobar una configuración.

Para abrir la línea de comandos de GRUB pulsamos la tecla *e* cuando aparece el menú al encender nuestro ordenador. Alternativamente, introducimos *grub* desde un terminal Bash, de un disco duro. Si deseamos iniciar una línea de comandos de GRUB desde un Live CD, tendremos que montar discos duros y posiblemente usar primero *chroot*. En todos estos casos, nuestra acción nos llevará hasta un prompt en el que podremos leer *grub* > .

En su forma más simple, la línea de comandos de GRUB es útil para arrancar un kernel o un sistema operativo. Simplemente

```
grub> find /boot/grub/stage1
(hd0,1)
```

Figura 5: La línea de comandos de GRUB dispone de algunos comandos como Bash. Aquí, el comando *find* encontrará la partición en la cual se instala GRUB buscando Stage 1.

introduciendo los mismos campos que definen un elemento en el menú *menu.lst*, uno a la vez. Sin embargo, podemos omitir *title*, ya que no tenemos la necesidad de distinguir la entrada de las otras en GRUB.

Si no estamos seguros de que podamos arrancar o si necesitamos reparar GRUB, la línea de comandos tiene un conjunto de varias docenas de comandos que podemos ver introduciendo *help*. Algunos de estos comandos son los mismos que en Bash, como *cat*, el cual muestra ficheros, o *find*, que los localiza (Figura 5). De modo similar, podemos usar *md5crypt* para encriptar una contraseña y *halt* o *reboot* para apagar el ordenador. Podemos regresar a Bash desde la línea de comandos de GRUB introduciendo *quit*.

Adicionalmente, la línea de comandos de GRUB tiene otras funcionalidades similares a las de Bash, incluyendo un historial en el que podemos buscar hacia delante y hacia atrás y un autocompletado con el tabulador. Esta última puede ser especialmente útil cuando estamos reuniendo información para reparar una máquina. Por ejemplo, introduciendo *root* (*hd* y pulsando el tabulador muestra todas las particiones arrancables en la máquina. Si sólo hay un disco duro presente podemos utilizar *root* (*hdo0*, más el autocompletado para obtener una

lista de las particiones y sus sistemas de ficheros. De modo similar, una vez que conocemos o que tenemos configurado la partición *root* de GRUB, podemos introducir después *kernel /boot/vmlinuz* y pulsar luego el tabulador para listar todos los kernels posibles.

Provistos con esta información, podemos utilizar *root* para establecer la localización de GRUB, y a continuación *setup* para reinstalar GRUB. Después de todo esto, abrimos *menu.lst* en un editor de texto para hacer todos los cambios necesarios a los secciones de manera que podamos arrancar todos los kernels posibles y sistemas operativos presentes.

Otras Alternativas y Cargadores de Arranque

La línea de comandos y *menu.lst* son solamente los modos más comunes de interactuar con GRUB. Si tenemos problemas con alguno de estos métodos podemos usar el comando *update-grub* para añadir entradas para actualizaciones instalables al final del menú *menu.lst* – aunque puede que tengamos que editar los resultados. De forma parecida, podemos usar *grub-install* para instalar o reinstalar GRUB.

Otra opción es usar el cargador de arranque. A pesar de que no puedo recomendar el uso de un cargador de arranque Windows con GNU/Linux porque su complejidad hace que GRUB parezca sencillo, es buena idea considerar el Linux Loader, o LILO [4], el gestor de arranque que GRUB sustituyó en la mayoría de las distribuciones hace 11 años.

Y si ninguna de estas opciones nos van bien, o si deseamos estar en la cresta de la ola en la materia, entonces aconsejo leer el artículo sobre GRUB 2 el mes que viene. ■

RECURSOS

- [1] Manual GNU GRUB – Color: http://www.gnu.org/software/grub/manual/html_node/color.html
- [2] Lista de comandos: http://www.gnu.org/software/grub/manual/html_node/Command_002dline-and-menu-entry-commands.html#Command_002dline-and-menu-entry-commands
- [3] Notas sobre SOs específicos: http://www.gnu.org/software/grub/manual/html_node/OS_002dspecific-notes.html#OS_002dspecific-notes
- [4] Configuración de LILO: <http://www.control-escape.com/linux/lilo-cfg.html>