

# El arranque en GNU/Linux.

Grub fue creado en 1995 por Erich Boleyn mientras intentaba poner en marcha Hurd, la eterna alternativa al nucle Linux que nunca terminó de arrancar. Erich modificó el código de arranque de FreeBSD (una distribución de Linux) para que soportase multiarranque. En 1999 Grub pasó a formar parte de GNU como paquete oficial, se expandió rápidamente, sustituyendo a Lilo, el gestor de arranque por excelencia hasta ese momento. Una de las ventajas que hizo que Grub sustituyese a Lilo, fue que era capaz de acceder al sistema de archivos, cosa que Lilo no podía hacer, ya que localizaba los archivos por posición física en disco (lo que quiere decir que cualquier variación en su posición física de disco impedía su localización).

En 2007, Grub2 se puso en marcha, y su adopción masiva por las principales distribuciones empezó en 2009.

## Proceso de arranque con Grub2

Cuando un ordenador basado en BIOS arranca, lo primero en ejecutarse es el MBC (el Master Boot Code ubicado en la MBR). En concreto el MBC es muy pequeño (446 Bytes), y dado que Grub es de mayor tamaño, El proceso de arranque de grub se hace en fases. La primera fase es precisamente el MBC. El MBC carga a su vez el resto de fases que se encuentran ya en la partición donde esté los archivos de arranque de GNU/Linux.

Una vez que Grub se ha cargado, muestra el contenido del archivo `grub.cfg` donde se encuentran las diferentes opciones de arranque. El usuario termina seleccionando una de las entradas.

Suponiendo que se ha elegido una entrada de GNU/Linux, el proceso de arranque del sistema operativo se hace en varias fases:

1. En primer lugar, el cargador de Grub carga el kernel de linux (archivo `vmlinuz`) y el ramdisk (archivo `initrd`) en memoria. El kernel realiza una inicilización básica (establece las tablas de paginación de memoria, detecta el tipo del CPU y funcionalidad adicional como capacidades de punto flotante), y después descomprime el archivo `initrd`, y lo monta como un sistema de archivos temporal (llamado `ramdisk`).
2. `initrd` contiene programas y módulos del kernel necesarios para cargar drivers para poder inicializar los dispositivos que contienen el sistema de archivos real. Así que el kernel emplea `initrd` para montar un sistema de archivos virtual en memoria, y poder así cargar en memoria el software y los controladores necesarios para poder montar el sistema de archivos real y hacer funcionar correctamente el hardware. Una vez que el proceso ha terminado, el kernel sustituye (con las herramientas disponibles en memoria) el sistema de archivos temporal por el real, e `initrd` es borrado de memoria.
3. Una vez que el kernel ha montado el sistema de archivos real, pasa el control al programa `/sbin/init`.
4. `Init` (o bien `upstart` o bien `systemd`) consigue que todo funcione como debe, una vez que

el kernel está totalmente en funcionamiento, estableciendo el espacio de usuario. Esto incluye la comprobación y montaje de sistemas de archivos, la puesta en marcha los servicios de usuario necesarios y cambiar al entorno de usuario cuando el inicio del sistema se ha completado. `Init` y `upstart` toman como parámetro el `runlevel`, conocido como nivel de ejecución, que tiene un valor entre 1 y 6, y que determina qué servicios se pondrán en funcionamiento. `Systemd` reúne dichos servicios con otra denominación: en lugar de `runlevel`, los llama `target`, aunque el concepto es el mismo.

## Trabajando con Grub2

Centos 7 incluye Grub2 como gestor de arranque. Grub2 lee su configuración desde el archivo `/boot/grub2/grub.cfg`. Este archivo contiene los menús que se muestran durante el inicio. A diferencia de la versión de Grub anterior, este archivo (`grub.cfg`) no está pensado para ser editado a mano, ya que se genera con la aplicación `/usr/sbin/grub2-mkconfig`, partiendo de:

- los kernels Linux ubicados en `/boot/`,
- las plantillas ubicadas en `/etc/grub.d/`,
- y las configuraciones presentes en `/etc/default/grub`

Cada vez que se instala un nuevo kernel, el archivo de configuración `grub.cfg` se actualiza automáticamente.

Cualquier cambio realizado en `/etc/default/grub` requiere reconstruir `grub.cfg`, mediante la ejecución de `grub2-mkconfig`, del siguiente modo:

```
# grub2-mkconfig -o /boot/grub2/grub.cfg
```

Si abrimos el archivo `grub.cfg` lo veremos lleno de entradas de tipo `menuentry`. Cada una de estas entradas, representa una opción del menú que Grub2 muestra al encender el ordenador.

## Configurar el menú de Grub2

Grub2 es capaz por sí mismo de localizar los sistemas operativos en el ordenador donde se ejecuta, y añadirlos al menú de inicio. Sin embargo el usuario puede querer crear una nueva entrada, o tener las entradas en un orden específico.

Grub2 usa una serie de scripts para crear el menú de unicio, que están en `/etc/grub.d/`. Se trata de los siguientes archivos.

- **00\_header** - carga la configuración de Grub2 desde el archivo `/etc/default/grub`.
- **01\_users** - se crea cuando se asigna una contraseña al gestor de arranque.
- **10\_linux** - localiza los kernels en la partición por defecto de CentOS
- **30\_os-prober** - crea entradas para los sistemas operativos encontrados en otras particiones.
- **40\_custom** - en esta plantilla se pueden añadir entradas adicionales.

# Cambiar la entrada por defecto de Grub

Por defecto, la variable `GRUB_DEFAULT` dentro del archivo `/etc/default/grub` tiene el valor `saved`. Este valor se utiliza para indicar a Grub2 que cargue la entrada indicada en el archivo `/boot/grub2/grubenv`. Para que la opción por defecto del menú sea otra, debemos usar el comando `grub2-set-default` que actualizará el archivo `/boot/grub2/grubenv`. Por ejemplo:

```
# grub2-set-default 2
```

NOTA: el número 2 indica la tercera entrada, ya que empezamos por 0.

Cualquier cambio requiere la reconstrucción de `grub.cfg` del siguiente modo:

```
# grub2-mkconfig -o /boot/grub2/grub.cfg
```

## Reinstalar Grub2

La reinstalación de Grub2 puede ser útil para reparar problemas relacionados con una instalación defectuosa de Grub2, archivos perdidos, etc. Otras opciones para reinstalar Grub2 pueden ser:

- Actualizar de Grub a Grub2
- Desaparición de Grub2 debido a que otro gestor de arranque lo ha destruido
- Añadir la información de arranque a otro disco diferente

Cuando se usa el comando `grub2-install` la información de arranque se actualiza. Observa el siguiente ejemplo:

```
# grub2-install /dev/sda
```

Este método se puede usar mientras los archivos de configuración de Grub2 están íntegros (`/etc/grub.d/*` y `/etc/sysconfig/grub`). Pero si dichos archivos están corruptos, contienen entradas erróneas, etc., no funcionará. Para ello podemos usar otro procedimiento: borrar y reinstalar Grub2.

## Borrar y reinstalar Grub2

El siguiente método borra todos los archivos de configuración de Grub2, y después devuelve la configuración de Grub2 a la configuración por defecto. Este método resuelve problemas causados por archivos corruptos y configuraciones incorrectas. Seguimos el siguiente procedimiento:

```
# rm /etc/grub.d/* # rm /etc/sysconfig/grub
# yum reinstall grub2-tools
# grub2-mkconfig -o /boot/grub2/grub.cfg
```

# Algunos comentarios sobre UEFI

Aparentemente, durante la instalación de Linux, no hay ninguna diferencia entre un sistema BIOS y otro UEFI. En éste último no habrá necesidad de crear una partición extendida, ya que UEFI soporta 128 particiones. A diferencia de BIOS, UEFI no instala parte de Grub en la MBR y el resto en el sector de la partición de arranque. En lugar de esto coloca los archivos necesarios (el binario de Grub compilado como aplicación UEFI, el archivo de configuración grub.cfg y los módulos de Grub) en la partición ESP (Efi System Partition). A continuación se muestran algunos comandos útiles para el trabajo sobre Grub en UEFI:

## Para reconstruir la configuración de Grub

```
# grub2-mkconfig -o /boot/efi/EFI/ubuntu/grub.cfg
```

## Para reinstalar los binarios de Grub en la ESP

```
# mkdir /boot/efi  
# mount /dev/sda1 /boot/efi → suponiendo que la ESP está en /dev/sda1  
# sudo grub-install --target=x86_64-efi --efi-directory=/boot/efi --bootloader-id=nuevo-grub --recheck  
→ bootloader-id da nombre al directorio donde se almacenará grub
```

Referencias:

<http://www.mauriciomatamala.net/SO/arranque-grub.php>