

Recuperación del sistema

En el caso de que haya un problema, el administrador tiene que poder establecer rápidamente el diagnóstico e implementar una solución para reparar y reestablecer la situación.

Para ello, hay que conocer bien las diferentes etapas de arranque del sistema, así como las distintas técnicas que permiten modificar los parámetros de arranque o arrancar a partir de un dispositivo de recuperación.

1. Inicio del sistema

Cuando se arranca la máquina en la que está instalado un sistema Linux, un programa inicializa los componentes materiales y después busca un programa de carga del núcleo (*boot loader*).

En las arquitecturas de hardware de tipo PC, este programa puede ser la BIOS o, cada vez más, UEFI, su sustituto más potente y seguro.

a. BIOS

La BIOS (*Basic Input/Output System*) es un programa almacenado en una memoria EEPROM o flash. Es la encargada de configurar y gestionar los componentes de hardware de bajo nivel de la máquina y de arrancar la carga del sistema operativo.

Cuando se acciona el botón de encendido de la máquina, se ejecuta la BIOS. Ésta efectúa tests llamados POST (*Power-On Self Test*) en los componentes de bajo nivel (memoria, procesador, etc.). Después busca un programa de arranque de un núcleo del sistema operativo (*boot loader*), recorriendo para ello sucesivamente los distintos dispositivos, según un orden configurado. En cuanto encuentra un programa en la ubicación específica del dispositivo (el sector de arranque del disco, el MBR, o el primer sector de la partición activa), lo carga en memoria y este se ejecuta.

La arquitectura de la BIOS es antigua y sus límites están impuestos por las primeras máquinas de tipo PC. Está cada vez menos adaptada a las posibilidades del hardware

moderno y da problemas de seguridad.

b. UEFI

El UEFI (*Unified Extensible Firmware Interface*) fue concebido para reemplazar a la BIOS, ofreciendo más funcionalidades y reforzando la seguridad. Administra una interfaz gráfica de alta resolución, la red y el multiboot (elección de la carga de diferentes sistemas operativos).

El UEFI utiliza un esquema de particionado de discos específico, el GPT (*GUID Partition Table*), que permite gestionar discos de gran tamaño con muchas particiones. En esos discos se pueden crear particiones especiales de arranque, llamadas ESP (*EFI System Partition*), en las que se almacenan los elementos necesarios para cargar el núcleo.

Cuando la máquina arranca, el UEFI puede ejecutar el cargador de arranque o cargar directamente el núcleo configurado, a través de una partición ESP. También puede usar el modo de compatibilidad CSM (*Compatibility Support Module*) para arrancar usando el MBR. Puede, también, detectar automáticamente la presencia de un cargador de arranque en un periférico externo o incluso a través de la red (arranque PXE, *Preboot eXecution Environment*).

El UEFI también puede implementar un procedimiento de carga del sistema con seguridad. Esta funcionalidad reposa en la gestión de claves de autenticación que han firmado el núcleo (o los núcleos) dispuestos a ser cargados. En ese caso, solamente se podrán cargar los núcleos oficiales de los desarrolladores.



En el caso de Linux, esto implica que el núcleo tiene que ser el que provee la distribución, lo que puede ocasionar problemas si se quiere personalizar este último.

c. NVMe

Los discos SSD que utilizan la interfaz de comunicación NVMe (*Non-Volatile Memory express*) son particularmente eficientes y, por lo tanto, usados cada vez más. Las

versiones de las distribuciones antiguas no sabían gestionar por defecto la carga del núcleo a partir de discos de este tipo. Había que implementar procedimientos relativamente complejos para que pudieran soportar estas configuraciones. Las versiones más recientes integran directamente estos dispositivos.

Para poder cargar el núcleo a partir de un periférico NVMe, hay que usar UEFI.

2. El gestor de arranque GRUB

El gestor de arranque (*boot loader*) es un programa ejecutable. Una parte de él se encuentra almacenada en el sector de arranque de un disco (MBR, *Master Boot Record*) y su función es encontrar y cargar en memoria el núcleo de un sistema operativo. Por lo tanto, hay que configurarlo para especificarle la ubicación de uno o varios archivos de núcleo y la partición sistema que tiene que montar cuando cargue el núcleo en cuestión.

En los entornos Linux, el gestor de arranque más utilizado es GRUB (*GRand Unified Boot loader*). Existen dos versiones:

- GRUB Legacy (o GRUB 1): es la versión más antigua, bastante usada todavía en producción, pero tiende a ser obsoleta.
- GRUB 2 (ou GRUB): es la versión por defecto de muchas distribuciones recientes.



Hay que conocer bien las dos versiones para obtener la certificación LPIC-2.

a. Configuración de GRUB Legacy

El archivo de configuración de GRUB Legacy es el archivo `/boot/grub/menu.lst` o `/boot/grub/grub.conf`, según las distribuciones.

El archivo está estructurado en secciones, cada una de las cuales constituyen un elemento del menú que será mostrado al inicio de GRUB.

En cada sección figuran la información relativa a la ubicación del núcleo que se cargará, su configuración de carga, la partición de sistema así como la ubicación del archivo `initrd` asociado.

Sintaxis de una sección

```
title EtiquetaMenú
root partición_Núcleo
kernel RutaNúcleo ro root=ParticiónSistema Opciones
initrd RutaInitrd
```

Con

<code>title EtiquetaMenú</code>	Etiqueta de la elección del menú asociado a la sección.
<code>root partición_Núcleo</code>	Partición que contiene el núcleo, en formato (<code>hdX,N</code>), donde <code>X</code> corresponde al número del disco, y <code>N</code> al número de la partición en el disco, los dos empiezan a partir de <code>0</code> .
<code>kernel RutaNúcleo</code>	Ruta del núcleo con respecto a su partición.
<code>ro</code> <code>root=ParticiónSistema</code>	Ruta de la partición del sistema, montada en solo lectura (<code>ro</code>).
<code>Opciones</code>	Parámetros utilizados al cargar el núcleo.
<code>initrd RutaInitrd</code>	Ruta de acceso al archivo de imagen de disco que contiene los módulos necesarios para el arranque.

La línea de opción `default` define el número de la sección que se tiene que cargar por defecto (empieza a partir de `0`):

default NúmeroSección

La línea de opción **timeout** define el tiempo en segundos que se mostrará el menú, antes de lanzar un arranque automático con la sección por defecto:

timeout NúmeroSegundos



Esta antigua versión de GRUB no es compatible con UEFI, excepto si este último utiliza el modo de compatibilidad MBR (CSR, *Compatibility Support Module*).

Ejemplo

Archivo de configuración de GRUB de una distribución CentOS 6:

```
vi /boot/grub/grub.conf
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making changes to this file
# NOTICE: You have a /boot partition. This means that
#   all kernel and initrd paths are relative to /boot/, eg.
#   root (hd0,0)
#   kernel /vmlinuz-version ro root=/dev/mapper/vg_centos6-lv_root
#   initrd /initrd-[generic-]version.img
#boot=/dev/sda
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title CentOS 6 (2.6.32-754.el6.x86_64)
    root (hd0,0)
    kernel /vmlinuz-2.6.32-754.el6.x86_64 ro root=/dev/mapper/
vg_centos6-lv_root rd_NO_LUKS rd_LVM_LV=vg_centos6/lv_swap rd_NO_MD
```

```
LANG=es_ES.UTF-8 SYSFONT=latarcyrheb-sun16 crashkernel=auto rd_LVM_LV=
vg_centos6/lv_root KEYBOARDTYPE=pc KEYTABLE=es-pc rd_NO_DM rhgb quiet
initrd /initramfs-2.6.32-754.el6.x86_64.img
```

b. Configuración de GRUB 2

La configuración de GRUB 2 se hace en el archivo `/boot/grub/grub.cfg`. Este archivo no se debe modificar. Este archivo se genera automáticamente con el comando `update-grub`, que utiliza scripts almacenados en `/etc/grub.d`. Estos scripts buscan los distintos núcleos de sistemas operativos ejecutables y configuran el menú de arranque de GRUB 2 en consecuencia.

Cuando se instala un nuevo núcleo, la actualización del archivo de configuración final se hace gracias a los comandos `update-grub` o `grub-mkconfig`.

Los elementos que pueden ser modificados por el administrador (modo de visualización del menú, tiempo antes del arranque, etc.) tienen que ser especificados en el archivo `/etc/default/grub`.

Ejemplo

Actualización de la configuración GRUB 2 en una distribución Debian 10:

update-grub

Generando un fichero de configuración de grub...

Found background image: /usr/share/images/desktop-base/desktop-grub.png

Encontrada imagen de linux: /boot/vmlinuz-4.19.117-amd64

Encontrada imagen de memoria inicial: /boot/initrd.img-4.19.117-amd64

Encontrada imagen de linux: /boot/vmlinuz-4.19.0-8-amd64

Encontrada imagen de memoria inicial: /boot/initrd.img-4.19.0 how now-8-amd64

hecho

El comando ha buscado todos los núcleos ejecutables para generar el archivo de configuración.

Creación de un archivo de configuración de test con el comando `grub-mkconfig`:

grub-mkconfig -o grub2-test.cfg

Generando un fichero de configuración de grub...

Found background image: /usr/share/images/desktop-base/desktop-grub.png

Encontrada imagen de linux: /boot/vmlinuz-4.19.117-amd64

Encontrada imagen de memoria inicial: /boot/initrd.img-4.19.117-amd64

Encontrada imagen de linux: /boot/vmlinuz-4.19.0-8-amd64

Encontrada imagen de memoria inicial: /boot/initrd.img-4.19.0 how now-8-amd64

hecho

El archivo `/etc/default/grub` contiene la declaración de las variables de configuración para GRUB 2. El administrador puede modificarlo.

Las principales variables son:

<code>GRUB_DEFAULT</code>	Número de la entrada de menú seleccionada por defecto (a partir de 0), o <code>saved</code> (núcleo definido por defecto).
<code>GRUB_TIMEOUT</code>	Duración máxima de la visualización del menú en segundos.
<code>GRUB_CMDLINE_LINUX</code>	Parámetros por defecto usados para lanzar el núcleo Linux.
<code>GRUB_TERMINAL</code>	Si la variable = <code>"console"</code> , menú en modo no gráfico.

Ejemplo

Archivo `/etc/default/grub` en una distribución CentOS 8:

vi /etc/default/grub

`GRUB_TIMEOUT=5`

`GRUB_DISTRIBUTOR="$(sed 's, release .*$,g' /etc/system-release)"`

`GRUB_DEFAULT=saved`

`GRUB_DISABLE_SUBMENU=true`

`GRUB_TERMINAL_OUTPUT="console"`

```
GRUB_CMDLINE_LINUX="crashkernel=auto resume=/dev/mapper/cl-swap
rd.lvm.lv=cl/root rd.lvm.lv=cl/swap rhgb quiet"
GRUB_DISABLE_RECOVERY="true"
GRUB_ENABLE_BLSCFG=true
```

c. Configuración de GRUB 2 en modo UEFI

GRUB 2 es compatible con UEFI. Si la instalación del sistema se ha hecho a partir de un arranque en modo UEFI, GRUB 2 utiliza, en un disco duro particionado en GPT, una partición ESP (tipo de partición ef00), de 35 MB como mínimo, con el flag boot, en la que se ha creado un sistema de archivos FAT32, montado por defecto en `/boot/efi`.

Este sistema de archivos contiene los archivos y directorios necesarios para el arranque del sistema en modo UEFI. Estos son los principales:

- ~ `EFI`: directorio principal.
- ~ `EFI/BOOT`: directorio que contiene los archivos para el arranque UEFI por defecto:

<code>BOOTX64.EFI</code>	Ejecutable de carga UEFI por defecto (para Linux, copia de <code>grubx64.efi</code> o de <code>shimx64.efi</code>).
<code>mmx64.efi</code>	Ejecutable de carga UEFI por defecto, en modo seguro.
<code>fbx64.efi</code>	Ejecutable para el modo de seguridad.

- ~ `EFI/NombreDistrib`: directorio que contiene los archivos para el cargador GRUB 2 de la distribución:

<code>BOOTX64.CSV</code>	Archivo de instalación, para añadir la entrada en NVRAM.
<code>grub.cfg</code>	Archivo de configuración de GRUB 2.
<code>grubx64.efi</code>	Ejecutable GRUB 2 en modo normal.
<code>mmx64.efi</code>	Ejecutable para la firma de la clave en modo de seguridad.
<code>shimx64.efi</code>	Ejecutable para el modo de seguridad.

El comando `efibootmgr` permite visualizar y modificar la configuración de la carga en modo UEFI.

Ejemplo

efibootmgr

```

BootCurrent: 0003
  BootNext: 0002
  BootOrder: 0003,0000,0001,0002
  Timeout: 30 seconds
  Boot0000* CD-ROM Drive(device:FF)
  Boot0001* Hard Drive(Device:80)/HD(Part1,Sig00112233)
  Boot0002* PXE Boot: MAC(00D0B7C15D91)
  Boot0003* Linux

```

El comando muestra los diferentes sistemas que se pueden arrancar. Linux es el sistema actual y el sistema por defecto en el orden de carga (`BootOrder`).

d. Uso de GRUB en modo interactivo

Durante el arranque, GRUB (Legacy o GRUB 2) muestra un menú que propone la carga de uno de los núcleos declarados en el archivo de configuración. El usuario puede seleccionar un núcleo o dejar que se cargue el que está definido por defecto.

En caso de problema durante el arranque, se puede usar GRUB en modo interactivo. Esto permite modificar los parámetros de carga del núcleo, o incluso especificar la ubicación del núcleo de recuperación, en un dispositivo externo.

Para ello, solamente hay que pulsar la tecla `e`, en el momento que se muestra el menú. GRUB se pone en modo de edición. Es posible desplazarse por las líneas de los diferentes menús, y modificarlas presionando de nuevo la tecla `e`. Después de haber hecho la modificación, hay que pulsar la tecla `b` (`boot`) (o `[Ctrl] x` en el caso de GRUB 2) y eso ejecutará la carga del núcleo teniendo en cuenta las especificaciones de la línea editada.

También se puede pulsar la tecla `c` cuando se muestra el menú. GRUB mostrará una línea de comandos y se pueden ejecutar comandos del shell GRUB.

e. Reinstalación desde un sistema que no arranca

Si el gestor de arranque GRUB ya no permite cargar correctamente el núcleo, se puede volver a instalar a partir de un sistema Linux de rescate o restauración cargado desde un dispositivo extraíble que contendrá un sistema Linux (Live-CD o un pendrive USB).

Para GRUB Legacy, se puede utilizar el comando `grub` sin argumento de manera interactiva y permite instalar el gestor de arranque usando el comando `setup` y especificando:

- `root (hdX,N)` : partición que contiene el archivo del núcleo.
- `setup (hdX)` : número del disco donde se instalará GRUB.
- `Quit` : salir del comando `grub`.

La reinstalación de GRUB 2 se hace directamente con el comando `grub-install`.

f. Shell de rescate en el momento del arranque

Si no se puede arrancar el sistema pero el núcleo sí se carga, se puede pedir al núcleo que active un shell básico en lugar de un sistema completo.

En ese caso, el núcleo se inicializa y después ejecuta un shell interactivo, asociado a la cuenta de superusuario, sin necesidad de teclear la contraseña.

Se pueden usar un parámetro al ejecutar el núcleo indicándole que lance un proceso. Para ello, debe editar la línea de carga del núcleo en el menú GRUB y añadir el parámetro `init=/bin/bash`.

Procedimiento:

- ˆ Arrancar físicamente el sistema.
- ˆ Pasar al modo de edición pulsando la tecla `e` desde la línea de carga del núcleo en el menú visualizado.
- ˆ Añadir el argumento `init=/bin/bash` al final de la línea: `kernel archivo_nucleo root=fs_raiz ro init=/bin/bash`
- ˆ Cargar el núcleo pulsando la tecla `b` (o `[Ctrl] X` en el caso de GRUB 2).

El núcleo será inicializado, este monta el sistema de archivos solo lectura y después ejecutará el proceso `bash`, asociado al UID 0 (superusuario). En este momento se puede volver a montar el sistema de archivos como lectura y escritura con el comando:

```
mount -rw -o remount /
```



Este procedimiento permite modificar la contraseña del superusuario, **sin tener que especificar la contraseña actual**. Sin embargo, se puede proteger la edición del menú GRUB con una contraseña específica.

g. Modos `systemd rescue` y `emergency`

Si la distribución Linux usa `systemd`, en el caso de que haya un problema en el arranque del sistema, se puede especificar el objetivo `systemd` que se tiene que activar, poniéndolo como argumento del comando de carga del núcleo.

Para ello, usando la funcionalidad de edición del menú de GRUB, hay que añadir el argumento siguiente al final de la línea de comandos de la carga del núcleo:

`systemd.unit=NombreObjetivo`

Por ejemplo, para arrancar en modo rescate, el argumento sería:

`systemd.unit=rescue`

Y para arrancar en modo de urgencia, el argumento sería:

`systemd.unit=emergency`