

GRUB

Como es el proceso de arranque

1- El sistema carga y ejecuta el gestor de arranque. Las especificaciones de este proceso dependen de la arquitectura del sistema. Por ejemplo:

- BIOS en sistemas basados en x86 ejecutan una primera etapa del gestor de arranque desde el MBR del disco duro primario, el cual, a su vez, carga un gestor de arranque adicional, GRUB.
- UEFI en sistemas basados en x86 montan una partición de sistema EFI que contiene una versión del gestor de arranque de GRUB. El gestor de arranque EFI carga y ejecuta GRUB como una aplicación de EFI.

2- El gestor de arranque carga el kernel en memoria, la cual a su vez carga los módulos necesarios y monta la partición root para sólo-lectura.

3- El kernel transfiere el control del proceso de arranque al programa `/sbin/init`.

4- El programa `/sbin/init` carga todos los servicios y herramientas de espacio del usuario y monta todas las particiones listadas en `/etc/fstab`.

5- Se le presenta al usuario una pantalla de inicio de conexión para el sistema Linux recién iniciado.

GRUB 2

GRUB 2(GNU GRand Unified Bootloader) permite al usuario seleccionar el sistema operativo o el kernel a ser cargado al momento del inicio.

El archivo de configuración de GRUB 2, `/boot/grub2/grub.cfg` es generado:

- Durante la instalación del S.O.
- Al invocar `/usr/bin/grub2-mkconfig`
- por `grubby` cada vez que se instala un kernel

Al utilizar `grub2-mkconfig` el archivo es generado de acuerdo a la plantilla ubicada en `/etc/grub.d/` y a la configuración almacenada en el archivo `/etc/default/grub`.

Por este motivo no se debe modificar el archivo `grub.cfg` ya que los cambios se perderán cada vez que se ejecute `grub2-mkconfig`.

Las operaciones sobre `grub.cfg` que normalmente se realizan ante la eliminación o instalación de un nuevo kernel se deben hacer mediante `grubby`.

Nombres de dispositivos en GRUB 2

Cuando se refiera a un dispositivo específico con GRUB, hágalo mediante el formato siguiente (observe que los paréntesis y las comas son muy importantes en la sintaxis):

```
(<tipo-de-dispositivo><número-de-dispositivo-bios>,<número-de-partición>)
```

El `<tipo-de-dispositivo>` especifica el tipo de dispositivo desde el cual GRUB arranca. Las opciones más comunes son `hd` para un disco.

El `<número de dispositivo bios>` es el número del dispositivo de BIOS. El disco duro IDE primario es 0 y un disco duro IDE secundario es 1. Esta sintaxis es casi equivalente a la que utiliza el kernel por dispositivos. Por ejemplo, la `a` en `hda` para el kernel es análoga al 0 en `hd0` para GRUB, la `b` en `hdb` es análoga al 1 en `hd1`, y así sucesivamente.

<partition-number> Especifica el número de la partición en un dispositivo. Al igual que <bios-device-number>, la mayoría de los tipos de particiones se enumeran a partir de 0. Sin embargo, las particiones BSD se especifican mediante letras, con a correspondiente a 0, b correspondiente a 1, y así sucesivamente.

Por ejemplo, si un sistema tiene más de un disco duro, GRUB se refiere al primer disco duro como (hd0) y al segundo como (hd1). De la misma manera, GRUB se refiere a la primera partición en el primer disco como (hd0,0) y se refiere a la tercera partición en el segundo disco duro como (hd1,2).

Ejemplo de una entrada de grub2

```
menuentry 'CentOS Linux (3.10.0-693.el7.x86_64) 7 (Core)'
--class centos --class gnu-linux --class gnu
--class os --unrestricted $menuentry_id_option
'gnulinux-3.10.0-693.el7.x86_64-advanced-a0de2b66-ac69-452d-a560-f8649349f3ed' {
    load_video
    set gfxpayload=keep
    insmod gzio
    insmod part_msdos
    insmod xfs
    set root='hd0,msdos1'
    if [ x$feature_platform_search_hint = xy ]; then
        search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos1
    --hint-efi=hd0,msdos1
    --hint-baremetal=ahci0,msdos1 --hint='hd0,msdos1'
    13ae7bb0-94fa-4731-be30-6554bffa839
    else
        search --no-floppy --fs-uuid --set=root
    13ae7bb0-94fa-4731-be30-6554bffa839
    fi
    linux16 /vmlinuz-3.10.0-693.el7.x86_64 root=/dev/mapper/centos-root
    ro crashkernel=auto
    rd.lvm.lv=centos/root rd.lvm.lv=centos/swap rhgb quiet
    initrd16 /initramfs-3.10.0-693.el7.x86_64.img
}
```

De lo anterior podemos destacar:

- Nombre de la entrada
- Módulos que se cargan (insmod)
- Root filesystem
- Kernel a cargar
- Initramfs

ACTIVIDAD 1

- Compruebe la versión del kernel que se está ejecutando usando el comando `uname -a`
- Identifique en la salida previa, la arquitectura
- Analice otros parámetros viendo `man uname`

Cambios temporales

Al momento de inicio, cuando se presenta el menú de grub2, podemos presionar la tecla "e" y de ese modo ingresar al menú de edición de grub. Todos los cambios que realicemos durarán hasta que reiniciemos el equipo.

Cambios permanentes con grubby

La herramienta grubby puede utilizar para leer la información de grub2, y generar el nuevo archivo /boot/grub2/grub.cfg, quedando de este modo de manera persistente los mismos.

- Para visualizar todos los kernels disponibles, ejecutamos

```
[root@localhost ~]# grubby --info=ALL
index=0
kernel=/boot/vmlinuz-3.10.0-693.21.1.el7.x86_64
args="ro crashkernel=auto rd.lvm.lv=centos/root rd.lvm.lv=centos/swap rhgb
quiet LANG=es_AR.UTF-8"
root=/dev/mapper/centos-root
initrd=/boot/initramfs-3.10.0-693.21.1.el7.x86_64.img
title=CentOS Linux (3.10.0-693.21.1.el7.x86_64) 7 (Core)
index=1
kernel=/boot/vmlinuz-3.10.0-693.el7.x86_64
args="ro crashkernel=auto rd.lvm.lv=centos/root rd.lvm.lv=centos/swap rhgb
LANG=es_AR.UTF-8 quiet"
root=/dev/mapper/centos-root
initrd=/boot/initramfs-3.10.0-693.el7.x86_64.img
title=CentOS Linux (3.10.0-693.el7.x86_64) 7 (Core)
index=2
kernel=/boot/vmlinuz-0-rescue-8f345dae63df40e39b2469ca7e7d8be9
args="ro crashkernel=auto rd.lvm.lv=centos/root rd.lvm.lv=centos/swap rhgb quiet"
root=/dev/mapper/centos-root
initrd=/boot/initramfs-0-rescue-8f345dae63df40e39b2469ca7e7d8be9.img
title=CentOS Linux (0-rescue-8f345dae63df40e39b2469ca7e7d8be9) 7 (Core)
index=3
non linux entry
```

Si queremos ver las opciones de una entrada en particular, le pasamos el kernel en cuestión

```
[root@localhost ~]# grubby --info=/boot/vmlinuz-3.10.0-693.21.1.el7.x86_64
index=0
kernel=/boot/vmlinuz-3.10.0-693.21.1.el7.x86_64
args="ro crashkernel=auto rd.lvm.lv=centos/root rd.lvm.lv=centos/swap rhgb quiet
LANG=es_AR.UTF-8"
root=/dev/mapper/centos-root
initrd=/boot/initramfs-3.10.0-693.21.1.el7.x86_64.img
title=CentOS Linux (3.10.0-693.21.1.el7.x86_64) 7 (Core)
```

Si queremos ver cual es el kernel que bootea por defecto

```
[root@localhost ~]# grubby --default-index
0
```

Si queremos ver que kernel es

```
# grubby --default-kernel
/boot/vmlinuz-3.10.0-693.21.1.el7.x86_64
```

ACTIVIDAD 2

- Liste los kernels instalados en su sistema usando grubby
- Corrobore cual es el kernel que inicia por defecto y su índice

- Liste los archivos que se encuentran en `/boot` para corroborar los kernels disponibles

Si queremos cambiar los argumentos de booteo

```
# grubby --remove-args "quiet" --update-kernel /boot/vmlinuz-3.10.0-693.el7.x86_64
```

Para ver un detalle completo de los mensajes de booteo elimine `rhgb quiet`, para ver los mensajes estándar de booteo deje solamente `quiet`.

Si queremos agregar un argumento de booteo

```
# grubby --args "quiet" --update-kernel /boot/vmlinuz-3.10.0-693.el7.x86_64
```

Si queremos actualizar todos los kernels, agregando o sacando argumentos

```
# grubby --update-kernel=ALL --args=console=ttyS0,115200 --remove-args="quiet"
```

si queremos cambiar la entrada de booteo por defecto

```
# grubby --set-default-index=0
```

ACTIVIDAD 3

- Pruebe el efecto que tiene quitar el argumento `quiet` y `rhgb` (reinicie el sistema en cada cambio)
- Modifique el kernel que se inicia por defecto por alguno de los disponibles
- Modifique los argumentos del kernel en forma temporal, desde los comandos durante el booteo
- Algunos parámetros globales de grub2 se modifican en el archivo `/etc/default/grub`. Cambie el valor del `GRUB_TIMEOUT` y luego ejecute

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

Como bootear el sistema si el archivo grub.cfg no existe

Si por error borramos `/boot/grub2/grub.cfg`, el sistema no iniciara, pero por suerte grub cuenta con una consola para la ejecución de los comando necesarios. Desde esta consola debemos consignarle los siguientes parámetros:

- `rootfs`
- `kernel`
- `initramfs`

Comandos útiles en la consola

En el menú de grub al inicio es posible ingresar comandos presionando la tecla `c``

- `ls` nos muestra los dispositivos que encontró y sus particiones
- `linux16` nos permite especificar el kernel a utilizar (recordemos que al kernel se le debe pasar como parametro cual es el `rootfs`, que en el caso de Centos es por defecto `/dev/mapper/centos-root`)
- `initrd16` nos permite cargar el archivo `initramfs` a utilizar.

Ejemplo paso a paso de recuperación:

ACTIVIDAD 4

- Borre el archivo `/boot/grub2/grub.cfg`
- Reinicie y en la consola de grub escriba

```
set root=(hd0,msdos1)
linux16 /vmlinuz-3.10.0-693.el7.x86_64 root=/dev/mapper/centos-root
initrd16 /initramfs-3.10.0-693.el7.x86_64.img
boot
```

Con eso conseguira bootear nuevamente el sistema, por lo que solo restará luego ejecutar `grub2-mkconfig` para que se vuelva generar dicho archivo

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

Recuperar el grub si se ha borrado el registro del MBR

En caso de que se haya borrado el registro MBR, el grub no podra arrancar y no contaremos con la consola anterior. En estos casos debemos inicar el sistema con un CD de rescate, como el de instalación de Centos, y seleccionar la opción de rescate.

Para volver a tener la opción de bootear windows, debemos agregar la siguiente líneas en el archivo `/etc/grub.d/40_custom`

```
menuentry "Windows 7" {
    set root=(hd0,3)
    chainloader +1
}
```

Luego ejecutamos

```
grub2-mkconfig --output=/boot/grub2/grub.cfg
```

Referencias

- Red Hat Enterprise Linux 7 System Administrator's Guide, cap. 25 (pág. 539).
- Red Hat Enterprise Linux 6 Guía de instalación, Apéndice F.
- [WikiCentos](#)
- [DocsFedora](#)