

# Grub2

## Sumario

- 1 Introducción a GRUB 2
- 2 Estrutura de ficheiros
  - ◆ 2.1 Directorio /etc/grub.d
- 3 Funcionamento
- 4 Engadir un novo script de GRUB
  - ◆ 4.1 Iniciar Windows XP
  - ◆ 4.2 Iniciar Windows 7
  - ◆ 4.3 Iniciar Windows 8
  - ◆ 4.4 Iniciar Ubuntu 12.04
- 5 Personalizar o menú de Grub2
  - ◆ 5.1 Engadir unha imaxe de fondo
  - ◆ 5.2 Configurar un Menú Personalizado
- 6 Reparar GRUB 2 borrado
- 7 **Importante** Se borramos o MBR dun sistema que tiña instalado o GRUB na partición do sistema, aínda que recuperes o GRUB, na partición, o equipo non iniciará. Isto pasa porque se borra o código da área do MBR necesaria para bootear o sistema operativo. Para arranxar este problema podes facer dúas cousas, recuperar o GRUB no MBR do disco, ou utilizar a opción de TestDisk MBR Code que fai o equivalente ao comando MS-DOS "fdisk /MBR".
- 8 **Reparar GRUB dende consola GRUB** Seguir o seguinte enlace tendo en conta que o arquivo **linux.mod** pode estar gardado en outro directorio do disco.
- 9 Burg
- 10 Enlaces interesantes

## Introdución a GRUB 2

GNU GRUB (GRand Unified Bootloader) é un xestor de arranque capaz de arrancar unha gran variedade de sistemas operativos, tanto libres como propietarios.

GRUB 2 é un sucesor de GRUB. Foi reescrito completamente para proporcionar ao usuario maior flexibilidade e rendemento.

## Estrutura de ficheiros

Os ficheiros que utiliza GRUB2 son:

- **/boot/grub/grub.cfg:**
  - Contén a información do menú de GRUB2. É o substituto do ficheiro **menu.lst** de GRUB.
  - Non debe editarse manualmente xa que se crea automaticamente en base a varios scripts ao executar o comando **update-grub** como root.
  - Sobreescíbese nalgúns actualizacións de GRUB2.
- **/etc/default/grub:** é o ficheiro de configuración dos menús de GRUB 2. Neste ficheiro configúranse, entre outras cousas, o tempo que espera GRUB a que usuario escollo unha entrada, o sistema operativo que se inicia por defecto, etc.
- **/etc/grub.d:** directorio que contén os scripts de GRUB2. Estes scripts son os bloques a partir dos cales se constrúe o ficheiro **grub.cfg**. Cando se executa **update-grub**, lense os scripts nunha secuencia determinada e créase o ficheiro **grub.cfg**.

## Directorio /etc/grub.d

Este é un exemplo dos arquivos que contén o directorio **/etc/grub.d**:

```

/etc/grub.d$ls -la
total 72
drwxr-xr-x  2 root root  4096 2011-10-12 16:30 .
drwxr-xr-x 140 root root 12288 2012-03-22 20:06 ..
-rwxr-xr-x  1 root root  6698 2011-10-01 14:40 00_header
-rwxr-xr-x  1 root root  5522 2011-10-01 14:14 05_debian_theme
-rwxr-xr-x  1 root root  7269 2011-10-01 14:40 10_linux
-rwxr-xr-x  1 root root  6344 2011-10-01 14:40 20_linux_xen
-rwxr-xr-x  1 root root  1588 2011-05-03 00:53 20_memtest86+
-rwxr-xr-x  1 root root  7545 2011-10-01 14:40 30_os-prober
-rwxr-xr-x  1 root root   214 2011-10-01 14:40 40_custom
-rwxr-xr-x  1 root root    95 2011-10-01 14:40 41_custom
-rw-r--r--  1 root root   483 2011-10-01 14:40 README
/etc/grub.d$

```

Os ficheiros que contén son:

- **00\_header**: é o script que carga a configuración de GRUB de `/etc/default/grub`.
- **05\_debian\_theme**: define a imaxe de fondo, cores do texto, etc.
- **10\_linux**: identifica os Kernels do sistema operativo en uso e engade as entradas de menú correspondentes.
- **20\_memtest86+**: busca o ficheiro `/boot/memtest86+.bin` e inclúe a utilidade memtest como unha opción no menú de arranque.
- **30\_os-prober**: é o script que escanea os discos duros buscando outros sistemas operativos e engádeos ao menú de arranque.
- **40\_custom**: é un modelo ou plantilla que se pode usar para engadir entradas adicionais no menú de arranque.

A numeración dos scripts define a precedencia, isto quere dicir que `10_linux` será executado antes de `20_memtest86+`. Ademais, as entradas do menú de arranque están ordenadas en base a esta numeración.

Os scripts tampouco deben editarse, excepto o script **40\_custom** que serve para engadir novas entradas ao menú.

Os scripts deben ter activado o **permiso de execución** (bit x). Se o bit x está desactivado, o script non se executa.

## Funcionamento

O funcionamento de GRUB 2 é moi sinxelo. O ficheiro `/etc/default/grub` contén a configuración; o directorio `/etc/grub.d/` contén scripts con información do menú de GRUB e scripts de arranque de sistemas operativos. Cando se executa o comando **update-grub** como root, execútanse os scripts e xunto coa información de `/etc/default/grub` créase o ficheiro `/boot/grub/grub.cfg`.

## Engadir un novo script de GRUB

Para engadir unha nova entrada ao GRUB hai que crear un ficheiro `/etc/grub.d/XX_nomeScript`, onde **XX** é número de dúas cifras. Este número establece a orde na que se colocarán as entradas do menú de GRUB.

A continuación móstranse unha serie de exemplos de GRUB para iniciar diversos sistemas operativos. Os exemplos foron executados nunha máquina virtual con varios discos duros conectados. Para facilitar as realización de probas, en cada disco duro estaba instalado un sistema operativo diferente.

### Iniciar Windows XP

Supoñemos que temos unha máquina virtual onde no segundo disco duro IDE está instalado Windows XP. Para engadir unha entrada ao grub para iniciar este sistema operativo, debemos realizar o seguinte:

Creamos o ficheiro `/etc/grub.d/51_winXP` co seguinte código:

```

#!/bin/sh
exec tail -n +3 $0
# This file provides an easy way to add custom menu entries.  Simply type the
# menu entries you want to add after this comment.  Be careful not to change
# the 'exec tail' line above.
menuentry "Windows XP - MANUAL" {
    set root='(hd1,msdos1)'

```

```

drivemap -s hd0 hdl
chainloader +1
}

```

- As liñas que comezan polo carácter # son comentarios, polo tanto non se procesan.
- As cinco primeiras liñas da imaxe pertencen á cabeceira do script plantilla.
- **menuentry**: define unha nova entrada no menú de GRUB. A cadea que vai entre comillas é o texto que aparecerá no menú de inicio de GRUB.
- **set root='(hd1,msdos1)'** establece cal é a partición de arranque para esa entrada do menú.

-**hd1** refírese ao segundo disco duro (os discos comezan a numerarse en 0)

-**msdos1** fai referencia á primeira partición. As particións comezan a numerarse en 1. Utilízase msdos1 para referirse á primeira partición dun disco duro MBR e gpt1 para referirse á primeira partición dun disco duro GPT.

- **drivemap -s hd0 hd1**: se Windows XP non está no primeiro disco duro, o sistema operativo non se iniciará. Esta liña permite enganar a Windows para que crea que está no primeiro disco.
- **chainloader +1**: utilízase con sistemas que non poden ser cargados directamente, como Windows. Como o seu nome indica, GRUB pasa o control a outro cargador de arranque, situado no dispositivo que indica a entrada do menú.

Ao executar o comando **update-grub** como root, o script **/etc/grub.d/30\_os-prober** xa detectaría automaticamente o novo sistema operativo Windows XP e engadiría a seguinte entrada ao ficheiro **/boot/grub/grub.cfg**. Así xa non sería necesario engadir manualmente a entrada para Windows XP.

```

menuentry "Microsoft Windows XP Professional (on /dev/sdb1)" --class windows --class os {
    insmod part_msdos
    insmod ntfs
    set root='(hd1,msdos1)'
    search --no-floppy --fs-uuid=rootsa280C5D780C5A269
    drivemap -s $(root)
    chainloader +1
}

```

## Iniciar Windows 7

Supoñemos que temos unha máquina virtual onde no segundo disco duro SATA está instalado Windows 7. Para engadir unha entrada ao grub para iniciar este sistema operativo, debemos realizar o seguinte:

Creamos o ficheiro **/etc/grub.d/52\_win7** co código que se mostra a continuación. Este ficheiro non inclúe ningún comando novo con respecto aos comandos utilizados na entrada de menú de Windows XP.

```

#!/bin/sh
exec tail -n +3 $0
# This file provides an easy way to add custom menu entries.  Simply type the
# menu entries you want to add after this comment.  Be careful not to change
# the 'exec tail' line above.
menuentry "Windows 7 - MANUAL" {
    set root='(hd1,msdos1)'
    chainloader +1
}

```

A entrada que se engadiría automaticamente ao ficheiro **/boot/grub/grub.cfg** ao executar o comando **update-grub** sería:

```

menuentry 'Windows 7 (loader) (on /dev/sdb1)' --class windows --class os $menuentry_id_option 'osprober-chain-86C86114C86103B1' {
    insmod part_msdos
    insmod ntfs
    set root='hd1,msdos1'
    if [ x${feature_platform_search_hint} = xy ]; then
        search --no-floppy --fs-uuid=rootsa280C5D780C5A269 --set=root
    else
        search --no-floppy --fs-uuid=rootsa280C5D780C5A269 --set=root
    fi
    chainloader +1
}

```

## Iniciar Windows 8

Supoñemos que temos unha máquina virtual onde no terceiro disco duro SATA está instalado Windows 8.1. Para engadir unha entrada ao grub para iniciar este sistema operativo, debemos realizar o seguinte:

Creamos o ficheiro **/etc/grub.d/53\_win8** co código que se mostra a continuación. Este ficheiro non inclúe ningún comando novo con respecto aos comandos utilizados na entrada de menú de Windows 7.

```
#!/bin/sh
exec tail -n +3 $0
# This file provides an easy way to add custom menu entries.  Simply type the
# menu entries you want to add after this comment.  Be careful not to change
# the 'exec tail' line above.
menuentry "Windows 8 - MANUAL" {
    set root='(hd2,msdos1)'
    chainloader +1
}
```

A entrada que se engadiría automaticamente ao ficheiro **/boot/grub/grub.cfg** ao executar o comando **update-grub** sería:

```
menuentry 'Windows 8 (loader) (on /dev/sdc1)' --class windows --class os $menuentry_id_option 'osprober-chain-D226B60D26B5F299' {
    insmod part_msdos
    insmod ntfs
    set root='hd2,msdos1'
    if [ x$feature_platform_search_hint = xy ]; then
        search --no-floppy --fs-uuid --set hint-bios=hd2,msdos1 --hint-efi=hd2,msdos1 --hint-baremetal=ahci2,msdos1  D226B60D26B5F299
    else
        search --no-floppy --fs-uuid --set root=D226B60D26B5F299
    fi
    # NOTA: Esta entrada non funciona se non se comenta a liña coa instrucción drivemap
    # drivemap -s (hd0) ${root}
    chainloader +1
}
```

**NOTA:** Se probamos a entrada engadida automaticamente, comprobaremos que Windows 8 **non inicia**. Para que inicie debemos **eliminar ou comentar a liña coa instrucción *drivemap***. A instrucción *drivemap* utilízase para mapear o disco de Windows como se fose o primeiro. Esta instrucción utilizábase na entrada de Windows XP, pois senón o sistema operativo non se iniciaba. Con Windows 8 xa non é necesario.

## Iniciar Ubuntu 12.04

Supoñemos que temos unha máquina virtual onde no cuarto disco duro SATA está instalado Ubuntu 12.04. Para engadir unha entrada ao grub para iniciar este sistema operativo, debemos realizar o seguinte:

Creamos o ficheiro **/etc/grub.d/54\_ubuntu1304** co código que se mostra a continuación. Neste exemplo só aparecen dúas liñas novas con respecto aos ficheiros anteriores:

```
#!/bin/sh
exec tail -n +3 $0
# This file provides an easy way to add custom menu entries.  Simply type the
# menu entries you want to add after this comment.  Be careful not to change
# the 'exec tail' line above.
menuentry 'Ubuntu 12.04 - MANUAL'{
    set root='(hd3,msdos1)'
    linux /boot/vmlinuz-3.2.0-23-generic root=/dev/sdd1
    initrd /boot/initrd.img-3.2.0-23-generic
}
```

As dúas últimas liñas da entrada do menú (**linux** e **initrd**) son os parámetros de arranque que necesita unha distribución de Linux. Nelas especificase a ubicación do kernel e do initrd na partición adecuada

A entrada que se engadiría automaticamente ao ficheiro **/boot/grub/grub.cfg** ao executar o comando **update-grub** sería:

```
menuentry 'Ubuntu 12.04 LTS (12.04)' --class gnu-linux --class gnu --class os $menuentry_id_option 'osprober-gnulinux-simple-3956adf9' {
    insmod part_msdos
    insmod ext2
    set root='hd3,msdos1'
```

```

if [ x${feature_platform_search_hint} = xy ]; then
    search --no-floppy --fs-uuid=${id} --set=hd3,msdos1 --hint-efi=hd3,msdos1 --hint-baremetal=ahci3,msdos1 3956adf9-32d3-47e7-9b43-ab813b543712
else
    search --no-floppy --fs-uuid=${id} --set=3956adf9-32d3-47e7-9b43-ab813b543712
fi
linux /boot/vmlinuz-3.2.0-23-generic root=UUID=3956adf9-32d3-47e7-9b43-ab813b543712 ro quiet splash $vt_handoff
initrd /boot/initrd.img-3.2.0-23-generic
}

```

## Personalizar o menú de Grub2

Pódese personalizar o menú de Grub2 dende o ficheiro **/etc/default/grub**. Neste ficheiro configúrase, por exemplo, a entrada seleccionada por defecto, o tempo de espera por defecto, etc.

A continuación descríbense algúns dos parámetros de configuración:

- **GRUB\_DEFAULT=0** selecciona a entrada por defecto en función da posición que ocupa no menú. As entradas empezan a numerarse en 0.
- **GRUB\_TIMEOUT=10** define o tempo de espera predeterminado en 10 segundos.
  - Se poñemos o valor **-1** o menú de Grub mostrarase indefinidamente ata que o usuario escolla unha opción.
  - se poñemos o valor **0** o sistema iníciase inmediatamente sen mostrar o menú.
- **GRUB\_DISABLE\_LINUX\_RECOVERY="true"**. Se quitamos o carácter **#** do inicio desta liña, as entradas de linux en modo de recuperación non aparecerán no menú.

Se queremos que arranque por defecto o último sistema operativo iniciado, deben modificarse as seguintes liñas do ficheiro:

- **GRUB\_DEFAULT=saved**
- **GRUB\_SAVEDEFAULT=true**

Recordar que as entradas do menú créanse en función dos scripts existentes no directorio **/etc/grub.d**. Se queremos que non se execute un script, debemos quitarlle o **permiso de execución** (bit x). Por exemplo, se queremos que no menú non aparezan as entradas de **memtest** debemos quitar o permiso de execución ao ficheiro **/etc/grub.d/20\_memtest86+**.

No caso de **Debian 9**, a utilidade **memtest86+** non está instalada por defecto e polo tanto xa non aparece no menú de arranque de GRUB2. Para engadir esta entrada no menú de arranque só hai que instalar o **paquete "memtest86+"**. Se queremos que as entradas desaparecan só temos que realizar os pasos descritos no parágrafo anterior.

**NOTA:** Despois de facer os cambios necesarios para personalizar o Grub2, hai que executar o comando **update-grub**.

## Engadir unha imaxe de fondo

GRUB 2 pode usar imaxes de fondo en PNG, JPG/JPEG ou TGA. Hai diversas formas de engadir unha imaxe de fondo ao GRUB:

- Engadir a seguinte liña ao ficheiro **/etc/default/grub** indicando a ruta onde se garda a imaxe

```
GRUB_BACKGROUND=/ruta/nomeImaxe
```

ou

- Colocar unha imaxe apropiada no directorio **/boot/grub**.

Despois, hai que actualizar o GRUB.

```
sudo update-grub
```

Pódese atopar máis información nesta [ligazón](#).

## Configurar un Menú Personalizado

Para facer un menú personalizado, só coas entradas de sistemas operativos que nos interesen, e non con todos os sistemas operativos que o comando **update-grub** atope, o único que debemos facer é:

- 1.- Configurar como NON executable o arquivo **30\_os-prober** existente no directorio **/etc/grub.d**, isto fai que non faga unha búsqueda automática de sistemas operativos.
- 2.- Configurar, tamén, como NON executables, todos os scripts do directorio **/etc/grub.d**, menos os das entradas que nos interesen a nós (engadidas "a man") e, ademáis, tamén deixar sempre como executables **00\_header**, **05\_debian\_theme** e **10\_linux**.
- 3.- Configurar o arquivo **/etc/default/grub** ao noso gusto.
- 4.- Por último, executar o comando **update-grub** como **root**.

## Reparar GRUB 2 borrado

Por moitas causas pode ocorrer que se borre o GRUB. Nós, para facer esta práctica vamos a borrar GRUB borrando o MBR empregando o comando **dd** e logo recuperalo.

Arrancamos o equipo e dende un Terminal executamos o comando:

```
# Se queremos borrar todo o MBR incluíndo a táboa de particións:
sudo dd if=/dev/zero of=/dev/sda bs=512 count=1

# Se queremos borrar o MBR pero deixar intacta ás táboas de particións:
sudo dd if=/dev/zero of=/dev/sda bs=446 count=1

# Estes exemplos son para borrar o MBR do disco sda
# Se queres saber antes cales son os discos que tes no sistema
# recorda que podes empregar o comando:
fdisk -l
```

- Se borramos as particións podemos recuperalas do xeito que se indica no seguinte [enlace](#).

- Para recuperar o GRUB utilizaremos un Live CD. Emprega un Live CD x86 ou 64bits dependendo de se o sistema a recuperar é dun tipo ou doutro.

```
# Buscamos cal é o disco e a partición do sistema
# O normal é que sexa o sda1 ou o hda1
sudo fdisk -l

# Montamos todo.
sudo mount /dev/sda1 /mnt

# Reinstalamos grub
sudo grub-install --boot-directory=/mnt/boot /dev/sda

# Para versións antigas (Ubuntu 10.04), o comando usado debe ser o seguinte:
sudo grub-install --root-directory=/mnt /dev/sda
```

- Para recuperar o GRUB de **Debian 9** utilizaremos un Live CD de Debian.

```
# Buscamos cal é o disco e a partición do sistema
# O normal é que sexa o sda1 ou o hda1
sudo fdisk -l

# Montamos a partición donde se atopa o directorio /boot. Neste caso, na partición /dev/sda1
sudo mount /dev/sda1 /mnt

# Instalamos grub2-common
sudo apt-get update
sudo apt-get install grub2-common

# Executamos as seguintes instrucións
cd /tmp
sudo apt-get download grub-pc-bin
dpkg-deb -R grub-pc-bin* grub/
sudo mv grub/usr/lib/grub/i386-pc/ /usr/lib/grub/

# Reinstalamos grub
sudo grub-install --boot-directory=/mnt/boot /dev/sda
```

Se o anterior non funciona, debemos executar o seguinte:

```
# Buscamos cal é o disco e a partición do sistema
```

```
# O normal é que sexa o sdal ou o hda1
sudo fdisk -l

# Montamos todo.
# mount --bind permítenos montar parte dun directorio xa montado.
sudo mount /dev/sda1 /mnt
sudo mount --bind /dev /mnt/dev
sudo mount --bind /dev/pts /mnt/dev/pts
sudo mount --bind /proc /mnt/proc
sudo mount --bind /sys /mnt/sys

# Agora podemos facer xa o chroot:
sudo chroot /mnt
#### Para facer chroot dende o systemrescuecd:
#### % chroot /mnt /bin/bash

# Por último cargamos o GRUB no MBR:
sudo grub-install --recheck /dev/sda

#### Ou cargamos o GRUB na partición do sistema:
#### % sudo grub-install --force /dev/sda1

# Reiniciamos e cando arranca o sistema (non o LiveCD),
# podemos axustar no menú do GRUB manualmente ou de xeito automático:
sudo update-grub
```



## Importante

Se borramos o MBR dun sistema que tiña instalado o GRUB na partición do sistema, aínda que recuperes o GRUB, na partición, o equipo non iniciará. Isto pasa porque se borra o código da área do MBR necesaria para bootear o sistema operativo. Para arranxar este problema podes facer dúas cousas, recuperar o GRUB no MBR do disco, ou utilizar a opción de [TestDisk MBR Code](#) que fai o equivalente ao comando MS-DOS "fdisk /MBR".



## Reparar GRUB dende consola GRUB

Seguir o seguinte [enlace](#) tendo en conta que o arquivo **linux.mod** pode estar gardado en outro directorio do disco.

## Burg

[Burg](#) (Brand-new Universal loader from Grub) é un menú animado para GRUB2.

Burg ten un funcionamento e unha [estrutura de ficheiros](#) equivalente á de GRUB2:

- **/boot/burg/burg.cfg**
- **/etc/default/burg**
- **/etc/burg.d**

```
# Para instalar Burg hai que engadir primeiro os repositorios
# Para Ubuntu 10.10 e anteriores
sudo add-apt-repository ppa:bean123ch/burg
# Para Ubuntu 11.04
sudo add-apt-repository ppa:n-muench/burg

#Actualizar
sudo apt-get update

# Instalar Burg
# Nas ventás de configuración debes escoller o disco duro onde queres que se instale Burg.
# O resto das opcións que aparezan non é necesario cambialas
sudo apt-get install burg
```

```
# Instalar os temas
sudo apt-get install burg-emu burg-pc burg-themes-common burg-common burg-themes

# Para ver os temas sen ter que reiniciar
# Pulsa F1 para mostrar a axuda e F2 para cambiar o tema
sudo burg-emu -D

# Para actualizar o arranque despois dalgún cambio
sudo update-burg
```

Se durante o proceso de instalación non se seleccionou o disco duro no que se ía instalar burg, pódese facer utilizando o seguinte comando:

```
#Para instalar burg no primeiro disco duro
sudo burg-install /dev/sda
```

## Enlaces interesantes

- [Tutorial sobre GRUB-2](#)
- [Grub2 - Community Ubuntu Documentation](#)
- [GNU GRUB2 Manual](#)

--Cristina París