

## GRUB2 (BIOS)

Els gestors d'arranc (com ara el Grub2) no estan instal·lats íntegrament al MBR perquè aquest només ocupa 512 bytes: això fa que gran part del Grub2 es trobi guardat dins de la carpeta /boot del sistema Linux instal·lat amb ell (la qual pot pertànyer a la mateixa partició que aquest sistema Linux o bé estar muntada en una partició a part). És per això que la majoria de vegades que haguem de configurar el Grub2, ho haurem de fer treballant sobre aquesta partició en comptes de dins del MBR (que conté un codi estable).

Tota la informació que necessiteu del Grub2 que no trobeu en aquests apunts es pot consultar aquí:  
<http://www.gnu.org/software/grub/manual/grub.html>

### **\*Opcions importants del fitxer de configuració /etc/default/grub:**

**GRUB\_DEFAULT= 0** Amb el valor 0 l'entrada per defecte serà la primera del menú ; amb el valor 1 la segona, i així.

**GRUB\_TIMEOUT=10** Número de segons d'espera fins executar l'entrada per defecte. Si el seu valor és -1, el temps d'espera és infinit.

**GRUB\_CMDLINE\_LINUX="opcions"** Paràmetres passats a tots els kernels Linux del menú

**GRUB\_CMDLINE\_LINUX\_DEFAULT="opcions"** Paràmetres a afegir als indicats a la directiva anterior només pels kernels Linux que no siguin de tipus “recovery mode”. Aquí és un bon lloc per exemple per afegir les entrades “quiet” ó “splash”, per exemple.

**GRUB\_DISABLE\_LINUX\_RECOVERY="true"** Aquesta directiva està comentada per defecte, així que per a què es tingui en compte primer s'ha de descomentar. Si el seu valor és *true* l'opció de “recovery mode” no apareixerà al menú.

*NOTA: Encara que aquesta línia no sigui visible, encara es podria entrar al “recovery mode” si seleccionem una entrada estàndard i pulsem “e”: això obrirà la possibilitat d'editar l'entrada seleccionada per aquest arranc; escrivint concretament la paraula “single” al final de la línia que comença per “linux” (i pulsant CTRL+X per arrancar) entrariem al “recovery mode”.*

**GRUB\_DISTRIBUTOR="nom"** Nom de l'entrada del menú corresponent al sistema actual. Normalment aquest nom s'obté executant alguna comanda que retorna un nom identificatiu del sistema (a sistemes Ubuntu, s'utilitza la comanda *lsb\_release -is* i a Fedora alguna equivalent a *cut -f1 -d" " /etc/system-release*)

**GRUB\_HIDDEN\_TIMEOUT=0** Número de segons que han de passar abans de mostrar el menú (a no ser que es premi una tecla qualsevol, moment en el qual es mostraria el menú immediatament). A partir de què es mostri el menú, començaria a contar el temps especificat a GRUB\_TIMEOUT.

**GRUB\_HIDDEN\_MENU\_QUIET=true/false** En el cas de què GRUB\_HIDDEN\_TIMEOUT valgui més de 0, si aquesta directiva val *false* mostrarà el compte enrera fins arribar a mostrar el menú, i si val *true* no mostrarà res

**GRUB\_GFXMODE=ampladaxalçada** Resolució a la què es veurà el menú. Els valors possibles es poden consultar executant una comanda interna del shell de Grub (al qual es pot accedir pulsant “c” quan aparegui el menú) anomenada *vbeinfo* (o també *videoinfo*)

**GRUB\_BACKGROUND="/boot/grub/imatge.png"** Ruta absoluta de la foto PNG que farà de fons de pantalla. La imatge es mostrarà redimensionada si cal.

*NOTA: A Ubuntu existeix un paquet anomenat “grub2-splashimages” que inclou una sèrie d'imatges de fons que s'instal·len a /usr/share/images/grub.*

**GRUB\_THEME=nom\_tema** Tema estètic (ubicat a /boot/grub/themes) a utilitzar. Tant als repositoris d'Ubuntu com els de Fedora hi ha disponible el tema “grub-theme-starfield”. Si volem crear nosaltres el nostre propi tema, devem consultar la documentació oficial per saber com fer-ho (nosaltres no hi entrarem):  
[https://www.gnu.org/software/grub/manual/html\\_node/Theme-file-format.html](https://www.gnu.org/software/grub/manual/html_node/Theme-file-format.html)

Per saber més sobre aquestes opcions (i moltes d'altres no exposades aquí), recomano llegir la documentació oficial. El seu índex és aquest: [http://www.gnu.org/software/grub/manual/html\\_node/Simple-configuration.html](http://www.gnu.org/software/grub/manual/html_node/Simple-configuration.html).

Després de realitzar qualsevol canvi en aquest arxiu, devem executar sempre la comanda *update-grub2* per a què els canvis es tinguin en compte. Aquesta comanda el que fa és regenerar automàticament el fitxer real de configuració del Grub, que és /boot/grub/grub.cfg. ¿Per què hi ha dos fitxers de configuració? Perquè és recomanable no editar mai aquest últim fitxer directament, ja que els canvis fets d'aquesta manera es perden en una següent execució

d'*update-grub2* , la qual pot donar-se amb una actualització del Grub, o amb la instal·lació d'un kernel nou, etc...però fent els canvis a */etc/default/grub* evitem aquests problemes.

**NOTA:** En realitat, *update-grub2* és un alias de la comanda *grub2-mkconfig -o /boot/grub/grub.cfg*

#### **\*Scripts presents a */etc/grub.d*:**

En realitat, */boot/grub/grub.cfg* no conté la informació especificada a */etc/default/grub* sinó la configuració establerta a una sèrie d'scripts presents dins de la carpeta */etc/grub.d* (que llegeixen el contingut de */etc/default/grub*). Cada cop que s'executa *update-grub2* són executats aquests scripts automàticament per tal de generar el */boot/grub/grub.cfg* corresponent. Aquests scripts s'executen en ordre seqüencial; és per això que el seu nom comença per números, l'ordre dels quals determina la seva execució (i de retruc, la ubicació dins del menú del Grub2 de l'entrada corresponent). Els arxius més importants són:

**00\_header** Llegeix */etc/default/grub* i carrega les configuració allà escrita

**05\_debian\_theme** Estableix els colors del text i els temes del menú -incloent les possibles imatges de fons-. Aquest script només es troba a Ubuntu; a Fedora l'hem d'escriure nosaltres dins de *40\_custom* o similar (veure més avall)

**10\_linux** Busca kernels Linux a la partició actual i els inclou com a ítems del menú (entrades "menuentry")

**20\_memtest86+** Inclou el programa */boot/memtest86+.bin* com un ítem més del menú. Aquest programa realitza un test dels mòduls de memòria RAM i es pot executar directament des del Grub sense necessitat d'arrencar cap sistema operatiu.

**NOTA:** Si no es vol que aparegui aquesta opció (així com qualsevol altra), simplement s'ha de treure el permís d'execució a aquest script (*chmod -x 20\_memtest86+*) i tornar a executar *update-grub2*.

**30\_os-prober** Busca altres sistemes operatius (Linux instal·lats a altres particions, Windows, etc) i els inclou com a ítems del menú (entrades "menuentry")

**40\_custom** Arxiu (inicialment buit) que serveix com a plantilla per incloure entrades de menú personalitzades definides per l'usuari.

**NOTA:** En realitat, es pot utilitzar qualsevol altre fitxer creat des de zero si el seu nom comença per un número igual o superior a 40, està ubicat dins de */etc/grub.d* i té permisos d'execució. El número amb el que comença el fitxer indicarà la posició relativa dins de les entrades visibles del menú.

#### **\*TRUC: Canviar els colors del menú:**

En el cas que estem a Ubuntu, hem d'editar l'script *05\_debian\_theme* i buscar les línies *set menu\_color\_normal* i *set menu\_color\_highlight*. La primera estableix el color de les entrades no seleccionades (en el format *color\_texte/color\_fons*) i la segona estableix el color de l'entrada seleccionada (en el mateix format). Els colors possibles són: black, dark-gray, light-gray, white, brown, yellow, red, light-red, blue, light-blue, green, light-green, cyan, light-cyan, magenta i light-magenta. A l'script també existeix la línia *set color\_normal*, que serveix per establir el color (de lletra i fons) dels elements del menú que no són les entrades (el text informatiu de sota, etc).

En el cas de Fedora, com que no existeix l'arxiu *05\_debian\_theme*, hem de crear un nou fitxer anomenat (per exemple) *06\_fedora\_theme* i escriure-hi el següent contingut:

```
#!/bin/sh -e
cat << EOF
    set menu_color_normal=blue/green
    set menu_color_highlight=white/dark-gray
    set color_normal=blue/green
EOF
```

**NOTA:** La línia inicial "*cat <<EOF*" i la final "*EOF*" serveixen per indicar on comença i on acaba el text que s'inclourà tal qual dins de */boot/grub/grub.cfg*

#### **\*TRUC: Afegir una entrada personalitzada al menú:**

Per afegir una entrada (això s'ha de fer al final de l'arxiu *40\_custom*) es poden consultar les entrades "menuentry" presents a */boot/grub/grub.cfg* per observar la sintaxis apropiada, ja que tenen la mateixa. Per exemple, per afegir una entrada personalitzada que executi un kernel d'Ubuntu a *sda3*, s'hauria d'escriure el següent:

**NOTA:** El Grub utilitza una nomenclatura pròpia pels discos durs, així: "*sda*" per ell és "*hd0*", "*sdb*" per ell és "*hd1*", etc. I les particions són així: "*sda1*" és "*hd0,1*" (o també "*hd0,msdos1*"), "*sdb2*" és "*hd1,2*" (o també "*hd0,msdos2*"), etc.

```

menuentry "Ubuntu chupiguay" {
    set root=(hd0,3)
    linux /boot/vmlinuz root=/dev/sda3 ...
    initrd /boot/initrd.img
}

```

Bàsicament, cada entrada es compon d'una secció “menuentry” indicant el nom visible de l'entrada, dins de la qual establim la partició on es troba el kernel a arrencar (directiva *set root*), la ruta dins d'aquesta on es troba aquest kernel (directiva *linux*), així com també la ruta del fitxer *initrd* (directiva *initrd*)

En el cas de ser una entrada de Windows, existirà la directiva *chainloader (hdX,X)+1*, la qual indica que Grub2 delega l'arranc de Windows al seu propi gestor d'arranc, el qual estarà instal·lat en el byte número 1 -el primer- de la seva partició particular. A més, s'han de carregar els mòduls dels Grub (que són com plugins addicionals, per entendre'ns) anomenats “chain” (per a què aquest mecanisme de delegació funcioni) i “ntfs” (per a què Grub2 pugui llegir el contingut de particions NTFS). Un exemple concret funcional seria:

```

menuentry "Windows 8" {
    set root=(hd0,1)
    insmod chain
    insmod ntfs
    chainloader (hd0,1)+1
}

```

#### **\*TRUC: Afegir una entrada personalitzada que arrenqui directament un arxiu iso:**

Grub2 té la possibilitat d'arrencar directament un kernel+initrd gravat dins una iso ElTorito guardada en un lloc accessible, sense que calgui instal·lar-los en cap partició prèviament. Això es fa mitjançant la directiva “*loopback*”, que serveix per crear una referència a l'arxiu iso que s'utilitzarà. Les directives *linux* i *initrd* es segueixen utilitzant per indicar la ruta dins de la iso on es troba el kernel i l'arxiu *initrd*, respectivament. Un exemple és el següent, el qual arrenca un Ubuntu, la iso de la qual està guardada a sda5:

```

menuentry "Lubuntu Live" {
    set root=(hd0,5)
    loopback pepito /ruta/lubuntu.iso
    linux (pepito)/casper/vmlinuz.efi iso-scan/filename=/ruta/lubuntu.iso ...
    initrd (pepito)/casper/initrd.lz
}

```

**NOTA:** El paràmetre *iso-scan* és imprescindible per a què el kernel sàpiga que tot el sistema es troba dins de la iso i no pas dins d'un CD o una partició real. Els punts suspensius indiquen que s'hauran d'afegir altres paràmetres per acabar de configurar l'arranc, paràmetres que s'han d'obtenir de l'arxiu de configuració de l'Isolinux present a la pròpia iso del Lubuntu (com per exemple: *boot=casper, quiet, splash, file=/ruta/fitxer.preseed,...*)

**NOTA:** Si no es vol escriure la ruta de l'arxiu explícitament en molts llocs del menuentry (per evitar haver de canviar-la a mà en un futur) es pot definir al principi una variable així, per exemple: *set isofile=/ruta/lubuntu.iso* i després escriure *\${isofile}* allà on hi hauria la ruta dins les línies *loopback* o *linux*

**NOTA:** Altres distribucions poden tenir altres paràmetres equivalents a *iso-scan/filename* (com ara *findiso, isoloop,...*). Per veure'n uns quants exemples, recomano visitar <https://github.com/thias/glim/tree/master/grub2>

#### **\*TRUC: Eliminar entrades del menú:**

Hi ha varies maneres que eviten editar l'arxiu *grub.cfg* directament:

\*Desinstal·lar els paquets corresponents als kernels a eliminar (a Ubuntu aquests paquets tenen el nom “*linux-image-3.xx-generic*” y “*linux-headers-3.xx*” i a Fedora “*kernel-3.xx*”). Això farà que s'esborrin automàticament del menú del Grub (perquè s'executa sol un *update-grub2*). Si no els volem desinstal·lar perquè simplement volem que no surtin al Grub, un truc és moure aquest kernel (i el seu fitxer *initrd* associat) de la carpeta */boot* a una altra qualsevol i executar llavors *update-grub2*: d'aquesta forma Grub no el podrà trobar i no l'afegirà.

\*Treure el permís d'execució als scripts presents a */etc/grub.d* corresponents a les entrades a eliminar i executar *update-grub*

\*Recordar que les entrades de tipus “*recovery mode*” es poden eliminar editant l'opció corresponent del fitxer */etc/default/grub*

**\*TRUC: Protegir amb contrasenya l'execució i edició de les entrades del menú:**

El primer que hem de fer és generar una contrasenya encriptada (encara que ho podem fer sense encriptar també) amb la comanda `grub-mkpasswd-pbkdf2` : ens preguntarà la contrasenya a encriptar i ens mostrarà el resultat per pantalla. Sense perdre de vista aquesta informació, crearem un fitxer EXECUTABLE anomenat `01_password` on especificarem un nom d'usuari inventat (que no té res a veure amb els usuaris de cap sistema) i la contrasenya generada al pas anterior, que estarà associada a aquest usuari, així:

```
#!/bin/sh -e
echo set superusers="pepito"
echo password_pbkdf2 pepito grub.pbkdf2.sha512.10000.743275C021A9EF460...
```

**NOTA:** Podríem haver creat més usuaris simplement indicant a `set superusers` el seu nom un rera l'altre separats per comes i escrivint altres línies `password_pbkdf2` associant una contrasenya diferent per cadascun d'aquests usuaris. Això serviria per tenir diferents parelles usuari-contrasenya (totalment equivalents).

A partir d'aquí, ja tindríem totes les entrades del menú protegides contra execució i edició. Si només volguéssim protegir una entrada contra edició però que no calgués utilitzar cap contrasenya per executar-la, hauríem de copiar l'entrada "menuentry" corresponent de `/boot/grub/grub.cfg` a `40_custom` i afegir el paràmetre `--unrestricted` just abans de `{`.

Recordar, en qualsevol cas, d'executar finalment `update-grub2` per tal de què aquests canvis tinguin efecte.

**\*TRUC: Recuperar un Grub preexistent després d'haver-lo matxacat (per una instal·lació de Windows, per exemple):**

Iniciem la màquina des d'un LiveCD corresponent al sistema matxacat i:

- 1.-Muntem la partició on es troba el Grub2 a recuperar (suposarem que és `/dev/sda1`). Això ho podem fer amb les comandes: `mkdir hola` i `mount /dev/sda1 hola`
- 2.-Instal·lem el Grub2 corresponent a aquest sistema Linux amb la comanda `grub-install --boot-directory=hola/boot /dev/sda`

**NOTA:** Fixar-se com en aquesta comanda s'especifica un nom de disc en comptes de partició perquè estem instal·lant part del Grub2 al MBR (concretament, el fitxer binari `/boot/grub/xxx/boot.img`), i fixar-se també que necessitem indicar en quina carpeta instal·larem la resta del Grub2 mitjançant el paràmetre `--boot-directory`

- 3.-Reiniciem i entrem al sistema Linux recuperat, executant per acabar la comanda `update-grub2`. D'aquesta manera trobarem la resta de sistemes que poden existir als discs.

**NOTA:** La comanda `grub-install` també es pot fer servir per instal·lar Grub2 en comptes de Syslinux en un llapis Usb. Simplement caldria executar: `grub-install --boot-directory=/mnt/usb/boot /dev/sdX`

**\*TRUC: Modificar la configuració del Grub2 des d'un LiveCD (quan cap entrada ens deixa arrancar en local):**

Iniciem la màquina des d'un LiveCD i:

- 1.-Muntem la partició on es troba el Grub2 a recuperar (suposarem que és `/dev/sda1`). Això ho podem fer amb les comandes `mkdir hola` i `mount /dev/sda1 hola`
- 2.-Modifiquem els arxius de configuració necessaris (`/etc/default/grub`, `/etc/grub.d/*...`)
- 3.-El pas anterior no és suficient perquè necessitem executar l'`update-grub2`. El problema és que si ho fem tal qual, ho estariem fent respecte el sistema Live (que de fet, no utilitza el Grub2 sinó Isolinux) i ho hem de fer respecte els sistemes instal·lats al disc dur. Per solucionar això hem de crear un chroot d'aquest sistema i executar aquesta comanda allà. Per tant devem fer:

```
mount --bind /dev hola/dev
mount --bind /sys hola/sys
mount --bind /proc hola/proc
chroot hola
...(si és necessari) grub-install /dev/sda
update-grub2
exit
umount hola/dev hola/sys hola/proc
```

**\*TRUC: Crear i utilitzar CDs de reparació del Grub2:**

En un sistema funcional amb el Grub2 ja instal·lat, cal executar la comanda: `grub-mkrescue -o nomfitxer.iso`. Fent això es crea una imatge iso autoarrencable ElTorito (concretament, el mètode és gravant l'arxiu `/boot/grub/xxx/cdboot.img` amb l'eina `xorriso`) que podrem cremar en un CD per rescatar el Grub2 en cas de fallada en un futur (gràcies a la seva consola incorporada, de la qual parlarem al següent truc).

Si desitgéssim afegir més entrades al menú d'aquest CD per aprofitar-lo millor només cal crear en una carpeta

qualsevol en el nostre sistema (l'anomenarem “pepita”) un fitxer semblant al `/boot/grub/grub.cfg` amb el text de l'entrada “menuentry” a afegir al menú del CD i copiar a l'arrel de “pepita” el kernel i l'`initrd` extres a arrencar. Finalment, hem d'incloure tot plegat dins de la iso així: `grub-mkrescue -o nomfitxer.iso pepita`

De totes formes, existeixen programes que ens permeten reparar el Grub2 (i moltes coses més) d'una forma realment senzilla. Un és Boot Repair (<http://sourceforge.net/p/boot-repair/home/Home>), el qual ve en forma d'iso autoarrencable i un altre (més complet) és Rescatux (<http://www.supergrubdisk.org/rescatux>)

**\*TRUC: Utilitzar la consola del Grub2 en temps real (només quan el menú és visible):**

Serveix per manipular el comportament del Grub2 de la forma que vulguem. Per això hem de pulsar la tecla “c” mentre el menú sigui visible. Entrarem en un shell particular (amb teclat en anglès) des d'on es podran utilitzar un conjunt de comandes, algunes de les quals són:

`help` : el seu significat és evident  
`set root=(hdX,Y)` : indica quina és la partició on es treballarà a partir de llavors (per accedir a fitxers)  
`linux /ruta/kernel [ paràmetres ]` : carrega el kernel especificat dins la partició indicada a `set root`  
`initrd /ruta/fitxer/initrd` : carrega el fitxer `initrd` dins la partició indicada a `set root`  
`boot` : arrenca el sistema especificat en les línies “linux” i “initrd”  
`set pager=1` : activa un paginador (similar a *more*) per quan la sortida de les comandes sigui massa llarga  
`set menu_color_normal=red/blue` : canvia el color de les entrades del menú en temps real. També podríem utilitzar `set menu_color_highlight`  
`insmod nommòdul` : carrega un mòdul concret (como “ext2”, “ntfs”, “chain”, ...)  
`vbeinfo/videoinfo` : mostra les resolucions gràfiques admittides pel Grub2  
`cat /ruta/fitxer` : evident. La ruta pot anar precedida de la notació (hdX,X) per indicar una partició concreta  
`halt` : apaga la màquina  
`ls` : mostra quins dispositius (discos i particions) són reconeguts pels diferents mòduls de Grub2. També es pot afegir una ruta absoluta d'una carpeta a la comanda per observar el seu contingut.  
**NOTA:** En el cas de què la carpeta `/boot` estigui muntada en una partició diferent a la del sistema, la “/” no representa l'arrel d'aquest sistema sinó aquesta carpeta `/boot`

Per sortir de la consola i retornar al menú original, cal premer ESC.

Trobareu totes les comandes de la shell interna de Grub2 a la documentació oficial. Concretament, a: [https://www.gnu.org/software/grub/manual/html\\_node/Command\\_002dline-and-menu-entry-commands.html](https://www.gnu.org/software/grub/manual/html_node/Command_002dline-and-menu-entry-commands.html)

**\*TRUC: Editar les línies del menú en temps real:**

Serveix per modificar el comportament de l'entrada seleccionada (per exemple, per afegir o treure paràmetres del kernel escollit). Per això hem de pulsar la tecla “e” mentre el menú sigui visible. Per arrencar l'entrada modificada, cal premer F10. Per sortir de l'edició de l'entrada i retornar al menú original com si no hagués passat res, premer ESC.