

# Grub

## Apprenez à maîtriser Grub !

Déjà, qu'est ce que Grub ?

Derrière ce nom barbare se cache un programme très utile aux utilisateurs de Linux. En vérité Grub est ce qu'on appelle un chargeur d'amorçage, ou (pour les anglophones *bootloader*). L'acronyme signifie **the GRand Unified Bootloader** ou littéralement **le Grand Chargeur Unifié**.

Je sais, ça ne veut rien dire ^^ mais ne vous inquiétez pas, nous allons clarifier ça tout au long de ce tutoriel.

- Note : le tuto était au départ destiné à la première version de Grub, mais une nouvelle version étant apparue entre temps, j'ai dû adapter le tuto pour les faire cohabiter. Ce tutoriel va donc vous proposer de configurer aussi bien Grub que Grub2 (Grub-pc). Ainsi, si vous voyez une mauvaise référence à l'ancienne version ou une manipulation qui n'est plus à jour (malgré le soin de mes petites mains et de celles de toute l'équipe), merci de me le signaler ! ;)

## Histoire et fonctionnement de Grub

Parlons tout d'abords du bien connu projet GNU.

### Sa création

Le projet GNU a été créé par Richard Stallman en 1984 alors qu'il travaillait au MIT (au laboratoire d'intelligence artificielle). Il souhaitait selon ses dires créer un système d'exploitation libre et complet pour "ramener l'esprit de coopération qui prévalait dans la communauté informatique dans les jours anciens".

La mascotte de GNU est un gnou (en raison de sa prononciation anglophone).

Image utilisateur

Le terme GNU est un acronyme récursif pour Gnu's Not Unix (Gnu N'est pas Unix). Il faut savoir qu'au moment de la création du projet GNU, le système UNIX était déjà bien lancé et reconnu par les informaticiens. GNU fut donc créé dans le but d'être compatible avec ce système.

En 1985, Stallman fonde la Free Software Foundation, structure logistique, légale et financière du projet GNU.

La FSF finance le développement du projet GNU et ce sont pour la plupart des communautés étudiantes américaines qui ont rendu ce projet viable. Le projet GNU se développe rapidement si bien que plusieurs sociétés contribuent au projet GNU en revendant ses logiciels, ou en offrant un support technique.

En 1990, GNU dispose de son propre éditeur de texte (Emacs), de son compilateur (GCC) et de la plupart des bibliothèques dont dispose Unix. Mais il manque encore un noyau.

Les systèmes 100% GNU sont des utopies mais il existe certains systèmes s'en approchant (comme [Debian](#)).

La totalité des distributions de Linux sont plus ou moins apparentées au projet GNU au point que Richard Stallman défendait l'appellation *distribution GNU/Linux*.

A ce stade, il manquait encore une pièce du puzzle : le noyau, et c'est ici que Linux fait son apparition.

Hé ! Je suis là pour vous parler de Grub pas de GNU ;) )

---

## L'arrivée de Grub

Grub est un programme de [multiboot](#) libre et gratuit.

Ce genre de programme se nomme **bootloader ou chargeur d'amorçage**. Il permet de choisir entre plusieurs systèmes d'exploitation (OS) sur une seule machine.

Grub a été créé en 1995 par Erich Boleyn à l'université de l'Utah. Avec l'aide de Brian FORD, il a créé un lanceur de multiboot. Erich a compris qu'il aurait plus vite fait d'écrire son propre chargeur plutôt que d'en modifier un autre.

Grub était né !

Erich a ajouté de nombreuses fonctions à GRUB, mais d'autres priorités l'ont empêché de suivre les demandes du nombre croissant d'utilisateurs. En 1999, Gordon Matzigkeit et Yoshinori K. Okuji ont adopté GRUB comme un projet GNU officiel.

Sa principale fonction est de pouvoir supporter la spécification multiboot.  
Ses concepteurs voulait qu'il soit en même temps :

- Simple d'utilisation ;
- Fonctionnel pour les experts et les concepteurs de noyau ;
- Qu'il soit compatible avec les systèmes du moment.

Une chose s'avère être intéressante : Grub dispose d'une interface menu lors de son chargement. On peut y ajouter autant d'entrées que l'on souhaite.

Il n'y a pas de limite, et, de plus on n'est pas forcé de mettre des "liens" vers un système : rien n'empêche au programmeur mégalomane d'insérer une ligne comme "Bienvenue Monsieur."

Grub possède une interface de commande. Elle s'affiche automatiquement si il manque un fichier de configuration ou si il y a une erreur.

Contrairement à [LILO](#), GRUB n'a pas besoin d'être réinstallé pour mettre à jour sa configuration. Une simple commande permet sa mise à jour `sudo update-grub`

GRUB est par ailleurs très flexible : il permet de charger aussi bien des systèmes compatibles avec le multiboot que des systèmes non-compatibles avec cette fonction (comme Microsoft Windows). Il supporte en outre beaucoup de [systèmes de fichiers](#) comme ext3, VFAT ou NTFS. GRUB est également compatible avec le mode [Logical Block Address](#) (anglais et pas très utile ici ^^ ).

---

# Le boot

Venons en au boot.

What ?

Héhé :p

*Citation : Wikipédia*

“ En informatique, le mot boot (apocope du mot anglais bootstrap, nom qui désigne la languette des chaussures pour pouvoir les enfiler plus facilement, voir la partie Historique de l'article pour plus de précisions) désigne la procédure de démarrage d'un ordinateur. En français son usage est un anglicisme, il peut être employé de façon synonyme avec le mot amorçage (du verbe amorcer) de

même que le mot bootable est synonyme d'amorçable. Initialement lorsque les ordinateurs n'avaient pas encore de disque dur, cette procédure nécessitait l'usage d'une disquette souple de 5"1/4 ; toutefois actuellement ce terme est également employé lorsqu'il s'agit d'un autre médium comme un CD, un DVD une clef USB, ou encore un accès réseau.

On distingue :

- \* le « boot à froid » (cold boot), obtenu en allumant la machine, ou en l'éteignant puis en la rallumant ;
- \* du « boot à chaud » (warm boot), ou «re-boot», obtenu en redémarrant. L'option est présente au niveau du système d'exploitation.

Derrière ce pavé se cache une notion fondamentale de l'informatique. Tout d'abord, en anglais, TO BOOT signifie INITIALISER.

En informatique, la procédure de boot est le démarrage de l'ordinateur en "passant la main" à un système d'exploitation.

A savoir : on ne boot pas forcément sur un système d'exploitation, on peut aussi booter sur un CD ou un DVD *ou pour les ordinateurs les plus récents* le boot sur un support USB ou support SD.

---

Que se passe-t-il entre le moment où vous appuyez sur le bouton de mise sous tension de votre ordinateur et le moment où vous voyez le logo de votre OS avec sa barre de chargement ?

Image utilisateur

Pendant le moment où vous voyez les écrans publicitaires avec les marques et logos de votre PC, un programme se charge : le BIOS.

Le BIOS (ou Basic Input Output System) va lire le MBR (Master Boot Record), premier secteur d'un disque dur où se trouve la table des partitions et un programme : le bootloader.

Ce superbe programme, stocké sur la ROM (mémoire morte) ou sur la mémoire vive, va extraire le noyau du système d'exploitation (kernel en anglais) qui fera le lien entre logiciel et matériel et donne sa dernière instruction : l'exécuter.

Et ce bootloader cela peut être Grub.

Voilà pour le début, la partie la plus ennuyeuse est passée. :p

---

# Installation de Grub

Vous allez ici apprendre à installer Grub2 sur votre machine. Depuis la version 2, c'est devenu très facile ! ;)

## ATTENTION !

La partie qui suit comporte certains risques, minimes certes mais pas inexistantes.

Les opérations décrites ici ont été effectuées par des millions de gens mais je ne saurais être tenu responsable des éventuels dommages rencontrés si vous vous écartez hors des sentiers battus.

## Installation

Installer Grub2 n'est pas difficile, il vous suffit d'installer le paquet *grub-pc*, puis de lancer la configuration automatique :

```
$ sudo apt-get install grub-pc
$ sudo grub-install /dev/sdX // Remplacez le X par la lettre de votre disque-
dur dans /dev/ (Exemple: /dev/sda)
$ sudo update-grub
```

Grub-install copiera tous les fichiers dont grub-pc a besoin dans le dossier */boot* et la commande *update-grub* générera un nouveau fichier *grub.cfg* (voir plus loin).

---

## Upgrade depuis la version précédente

La première version de Grub est peut-être déjà installée sur votre machine. Il est alors possible que Grub2 soit automatiquement mis à jour si la version de votre système est récente, mais pour les versions antérieures, vous serez probablement amenés à effectuer les manipulations suivantes :

```
$ sudo apt-get install grub-pc
$ sudo upgrade-from-grub-legacy
```

Durant l'installation, vous devrez configurer le disque sur lequel se trouve la partition */boot*.

Image utilisateur

Attention, n'appuyez surtout pas directement sur ENTRER sans avoir confirmé la sélection du disque, sous peine de corrompre Grub legacy sans installer correctement Grub2. Durant le prochain démarrage, vous allez obtenir une erreur "GRUB Error 15" (il faudra alors utiliser un Live-CD afin de réparer Grub).

Voilà, GRUB est enfin installé.

Si tout s'est bien passé, félicitations à vous. Si ce n'est pas le cas, reportez-vous à la partie "Réparation et Restauration" (en cours de rédaction :-° ) ou en attendant, aux forums ou bien envoyez moi un [MP](#) ;)

---

## Configurez Grub2

Ce chapitre va vous permettre de vous faire la main avec Grub2 et de le configurer à vos envies. A vos claviers !

Suite à un commentaire, je vous signale qu'il y a une différence importante entre Grub et Grub2 dans la numérotation des partitions :

- Sur Grub, la numérotation démarre à 0 pour les disques physiques ET pour les partitions.  
Exemple : `/dev/hda1` ou `/dev/sda1` sera nommé `hd(0,0)` par grub.
- Sur Grub2, elle démarre à 0 pour les disques physiques et à 1 pour le numéro de la partition.  
Exemple : `/dev/hda1` ou `/dev/sda1` sera nommé `hd(0,1)` par grub-pc.

## Mise au point

Voici d'abord ce que la plupart d'entre vous ont (je dis la plupart car cela peut varier selon vos OS installés) :

Image utilisateur

*Un Menu classique de GRUB*

Ce n'est pas très accueillant, je vous l'accorde...

---

# Les fichiers à modifier

## Le fichier /etc/default/grub (debian)

On commence par ouvrir le fameux fichier (en root bien sur) :

```
# gedit /etc/default/grub // Pour Gnome
# kate /etc/default/grub // Pour KDE
```

### Citation

```
## # If you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg.

GRUB_DEFAULT=0
GRUB_HIDDEN_TIMEOUT=0
GRUB_HIDDEN_TIMEOUT_QUIET=true
GRUB_TIMEOUT="3"
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
GRUB_CMDLINE_LINUX=""

# Uncomment to disable graphical terminal (grub-pc only)
#GRUB_TERMINAL=console

# The resolution used on graphical terminal
# note that you can use only modes which your graphic card supports via VBE
# you can see them in real GRUB with the command `vbeinfo'
#GRUB_GFXMODE=640x480

# Uncomment if you don't want GRUB to pass "root=UUID=xxx" parameter to
Linux
#GRUB_DISABLE_LINUX_UUID=true

# Uncomment to disable generation of recovery mode menu entries
#GRUB_DISABLE_LINUX_RECOVERY="true"
```

Dans cette couleur bleue, ce sont les commentaires.

Dans cette couleur rouge, ce sont les parties actives qui seront traitées.

## Le répertoire /etc/grub.d/ (debian)

La commande `ls -al` nous renvoie :

```
total 52
drwxr-xr-x  2 root root  4096 2009-11-09 23:45 .
drwxr-xr-x 169 root root 12288 2009-12-07 17:40 ..
-rwxr-xr-x  1 root root  3296 2009-10-24 02:44 00_header
-rwxr-xr-x  1 root root  1154 2009-10-24 02:31 05_debian_theme
-rwxr-xr-x  1 root root   783 2008-03-04 19:20 06_dell_theme
-rwxr-xr-x  1 root root  3778 2009-10-24 02:44 10_linux
-rwxr-xr-x  1 root root   772 2009-10-23 18:11 20_memtest86+
-rwxr-xr-x  1 root root  5467 2009-10-29 17:21 30_os-prober
-rwxr-xr-x  1 root root   214 2009-10-24 02:44 40_custom
-rw-r--r--  1 root root   483 2009-10-24 02:44 README
```

A l'exception du README, les autres sont des script bash. La génération du fichier grub.cfg va se passer tout simplement en exécutant l'ensemble de ces scripts dans un ordre croissant, l'ordre étant appliqué sur les numéros XX précédent le caractère \_ dans le nom de ces scripts.

- **00\_header** : script permettant la génération de l'en-tête du fichier grub.cfg. Cet en-tête est généré principalement à l'aide des informations que vous avez paramétrées dans le fichier /etc/default/grub.
- **05\_debian\_theme** : script permettant la génération des informations sur le thème graphique du menu de démarrage. À l'heure où j'écris ces lignes, sur Ubuntu 9.10, le thème est minimal : du texte blanc sur un fond noir.
- **10\_linux** : script permettant de générer les entrées correspondant à votre système GNU/Linux hôte.
- **20\_memtest86+** : script permettant de générer les entrées memtest.
- **30\_os-prober** : script permettant de détecter des nouveaux systèmes lors de la génération du fichier grub.cfg et de générer les entrées correspondantes. Il est normalement capable de détecter les systèmes avec un noyau Linux, un noyau Hurd, les systèmes Windows et les systèmes Mac OS X.
- **40\_custom** : script permettant de générer des entrées introduites manuellement dans ce fichier.

Les vérifications memtest86+ sont parfois très utiles, mais sont rarement employées et sont surtout très longues ...

Il est donc possible de les désactiver, mais comment ?

Rappelez-vous :

*Citation : Moi*



A l'exception du README, les autres sont des script bash. La génération du fichier grub.cfg va se passer tout simplement en exécutant l'ensemble de ces scripts dans un ordre croissant, l'ordre étant appliqué sur les numéros XX précédent le caractère \_ dans le nom de ces scripts.

Il suffit donc d'empêcher le script memtest86+ de s'exécuter.

Or, pour empêcher ça, nous avons les chmods, facile : `sudo chmod -x 20_memtest86+`

## Le fichier /boot/grub/grub.cfg

**Alors là, pas touche !** Si vous êtes sur Debian (Ubuntu par exemple), ce fichier est généré automatiquement par Grub après chaque mise à jour de votre part (n'oubliez pas de le faire après les modifications) : `sudo update-grub`

Sinon, vous pouvez le modifier ; le principe reste le même que plus haut (la syntaxe est un peu différente):

- **(hdn,m)** est la partition m du disque n, le numéro du disque commence par zéro (0), les partitions par un (1).
- **set default=n** est l'entrée par défaut, qui démarrera automatiquement après le time-out.
- **set timeout=m** le temps m m est le temps d'attente avant que le choix par défaut ne démarre.
- **menuentry "str" {options} title string 'str'** pour préciser manuellement une entrée du menu et la façon donc elle sera affichée.
- **set root=(hdn,m)** définit la partition root (/), là où est installé le noyau.
- **linux /path ro root=/dev/device initrd /initrd.img** options pour la partition root, si le noyau n'est pas installé sur la partition / .
- **chainloader +1** permet de passer le relais à un autre bootloader (exemple celui de Windows, nécessaire à son démarrage)

---

# Modification des entrées

Ici, ça se complique un peu (non, ne partez pas, j'ai dit un peu ! :p )

La ligne de base

Nous allons éditer la ou les lignes commençant par **GRUB\_DISTRIBUTOR** (je n'en ai qu'une, mais je n'exclus pas la possibilité de systèmes un peu plus exotiques ^^ )

Ces lignes vont définir ce qui sera affiché à l'écran suivant les options choisies.

Choisissez parmi les options suivantes :

*Citation : Doc*

```
## GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian` # donne :  
Ubuntu, Linux 2.6...  
#GRUB_DISTRIBUTOR=`lsb_release -d -s 2> /dev/null || echo Debian` # donne :  
Ubuntu 9.10, Linux 2.6...  
#GRUB_DISTRIBUTOR=`echo -n $(lsb_release -c -s 2> /dev/null || echo Debian)`  
# donne : Ubuntu 9.10 karmic, Linux 2.6....
```

Les autres

Il est aussi possible d'ajouter des entrées "manuellement". Pour cela, éditez le fichier **/ect/grub.d/40\_custom** et ajoutez vos entrées. En effet, tout ce qui est présent dans ce fichier sera recopié tel quel dans le fichier **grub.cfg**.

Exemple:

*Citation : 40\_custom*

```
## BEGIN /etc/grub.d/40_custom ##  
# 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" {  
  set root=(hd0,1)  
  chainloader +1  
}
```

---

# Protégez Grub2 avec un mot de passe

Bon, ici je ne suis pas un pro et le web n'est franchement pas riche en informations, donc malgré tous mes tests, je vous conseille une émulation avant de modifier véritablement ce qui suit.

Protéger automatiquement tous les recovery modes

Le recovery mode est un mode de Linux vous permettant de réparer votre OS en cas de problème et vous offre notamment un shell normal ainsi que plusieurs autres options intéressantes.

Pour le protéger, il vous suffit de dé-commenter cette ligne du fichier **/ect/default/grub** :

*Citation : menu.lst*

```
## #GRUB_DISABLE_LINUX_RECOVERY="true" // Enlevez le #
```

On n'oublie pas de mettre à jour :

```
$ sudo update-grub
```

Ainsi Grub ne générera plus d'entrée recovery dans son menu. Notez cependant que le jour où vous avez un problème, vous risquez fort de devoir booter sur un Live CD pour la re-commenter.

Protéger entièrement le menu

Vous pouvez également protéger avec un mot de passe tout le menu.

Ce [How To](#) m'a été proposé dans les commentaires. Je ne traiterai pas cette partie plus en détail car :

- je n'aime pas vraiment le concept (devoir sans arrêt taper son mot de passe à chaque démarrage...);
- ces explications, bien qu'en anglais, sont très bien rédigées.

---

## Le démarrage

Eh oui, la charrue avant les bœufs...

Reprenons le début du fichier **/etc/default/grub** :

*Citation*

```
GRUB_DEFAULT=0
```

Surement la plus intéressante : elle vous permet de modifier l'entrée de votre menu sur laquelle vous allez booter automatiquement. La valeur peut être un nombre naturel (définissant la position) ou *saved*. Ce dernier indique que l'entrée choisie au dernier démarrage sera la prochaine entrée par défaut. Lorsque cette ligne est commentée, la valeur par défaut est 0.

*Citation*

---

```
GRUB_TIMEOUT="5"
```

Cette ligne précise le nombre de **secondes** à attendre avant de booter sur le *default*. La valeur est soit un nombre naturel (secondes), soit -1. Ce dernier indiquera que le menu doit attendre indéfiniment. Précisons que si la valeur est 0, l'entrée par défaut sera bootée instantanément. Lorsque cette ligne est commentée, la valeur par défaut est 5.

*Citation*

```
“ #GRUB_HIDDEN_TIMEOUT=0
```

Permet de définir si le menu de démarrage doit être affiché ou non. Si ce paramètre est commenté, il sera affiché, sinon, sa valeur est un nombre naturel (secondes). Lequel définit le nombre de secondes à attendre avant de lancer le boot de l'entrée par défaut. Si cette valeur est supérieure à 0, pendant le nombre de secondes définies, vous aurez la possibilité de faire afficher le menu manuellement en appuyant sur Esc par exemple. Notez bien la différence avec le précédent : ici, vous ne verrez pas le menu.

*Citation*

```
“ #GRUB_HIDDEN_TIMEOUT_QUIET=true
```

Permet de définir si un chronomètre doit être affiché durant les GRUB\_HIDDEN\_TIMEOUT secondes définies dans le paramètre précédent. Sa valeur est true ou false. Lorsque cette ligne est commentée, la valeur par défaut est false.

---

## Fond d'écran

Plutôt que de faire un vieux plagia, je préfère vous rediriger vers la doc, bien plus complète sur le sujet.

Le lien est le suivant :

[http://doc.ubuntu-fr.org/grub-pc#exemple\\_fond\\_d\\_ecran](http://doc.ubuntu-fr.org/grub-pc#exemple_fond_d_ecran)

Notez cependant que ce n'est pas quelque chose d'essentiel, et que la plupart d'entre vous choisiront sûrement de ne pas afficher leur menu du tout.

Pas vraiment de conclusion, les choses vues dans ce chapitre sont assez diverses et variées.

Encore une fois, si vous avez rencontré un problème, vous pourrez vous reporter au chapitre sur les erreurs fréquentes (en cours de rédaction) ou bien poster un message sur les forums (pensez à faire une recherche avant, les forums grouillent de sujets sur les erreurs de Grub) ou encore m'envoyez un MP, je me ferai un plaisir de vous aider ! ;)

---

# Configurez Grub-legacy

Dans ce chapitre, vous allez apprendre à configurer Grub (ou encore "Grub legacy", l'ancienne version) pour le rendre plus convivial et à terme plus sécurisé !

## Déterminer l'ordre de boot

Prenez votre menu et définissez le **numéro** auquel va correspondre le système qui vous intéresse de cette façon :

```
Ubuntu, kernel 2.6.17-10-generic
```

```
Ubuntu, kernel 2.6.17-10-generic (recovery mod)
```

```
Ubuntu, memtest86+
```

```
Other operating systems:
```

```
Windows XP Media Center Edition
```

```
Windows NT/2000/XP
```

```
Microsoft Windows XP Embedded
```

Sur mon code :

- **Ubuntu, kernel 2.6.17-10-generic = 0**
- Ubuntu, kernel 2.6.17-10-generic (recovery mode) = 1
- Ubuntu, memtest86+ = 2

- **Other operating systems = 3**
- Windows XP Media Center Edition = 4
- Windows NT/2000/XP = 5
- Microsoft Windows XP Embedded = 6

La première ligne compte pour 0.

Il est important de prendre en compte les lignes d'information telles que "Other operating systems".

Notez bien le nombre qui vous intéresse : celui sur lequel vous désirez booter par défaut.

---

## Méthode 1 : Le default

Une fois revenu sur votre bureau, ouvrez un terminal et entrez-y :

- pour KDE :

```
kdesu kate /boot/grub/menu.lst
```

- pour Gnome :

```
gksudo gedit /boot/grub/menu.lst
```

- pour XFCE :

```
sudo mousepad /boot/grub/menu.lst
```

Si vous n'avez pas compris, on va ouvrir le fichier cible de l'adresse **/boot/grub/menu.lst**.

Effectivement, ce n'est pas très digeste... :(

Le mien est résumé ainsi :

```
default 0

timeout 15

hiddenmenu

title Ubuntu, kernel 2.6.17-10-generic
```

```
root (hd0,4)
kernel /boot/vmlinuz-2.6.17-10-generic root=/dev/sda5 ro quiet splash
initrd /boot/initrd.img-2.6.17-10-generic
quiet
savedefault
boot

title Ubuntu, kernel 2.6.17-10-generic (recovery mode)
root (hd0,4)
kernel /boot/vmlinuz-2.6.17-10-generic root=/dev/sda5 ro single
initrd /boot/initrd.img-2.6.17-10-generic
boot

title Ubuntu, memtest86+
root (hd0,4)
kernel /boot/memtest86+.bin
quiet
boot

title                Other operating systems:
root

title                Windows XP Media Center Edition
root                (hd0,0)
savedefault
makeactive
chainloader          +1
```

Remplacez tout simplement le chiffre suivant *default* par celui qui vous intéresse.

```
default X
```

Où X est le chiffre que vous avez choisi en première partie.

Enregistrez votre fichier, fermez et redémarrez.

Magique :magicien: , le curseur est déjà placé sur le choix que vous vouliez. Laissez passer le temps et Grub bootera sur votre choix par défaut.

C'est tout.

---

## Méthode 2 : savedefault

Il existe une autre méthode afin de déterminer le système par défaut.

Grub vous propose de déterminer lui-même le système sur lequel vous allez booter, en ne choisissant pas n'importe lequel, mais le dernier état sauvegardé.

Question logique : comment sauvegarde-t-on ?

Regardez vos entrées, avec un peu de chance, vous devriez y voir au moins un *savedefault*. Cela signifie que vous allez automatiquement enregistrer ce choix comme choix par défaut pour le boot.

Placez donc *savedefault* après votre choix le plus important et prenez soin de le retirer aux autres. Ensuite, remontez en haut du fichier et éditez encore une fois la ligne *default X* et changez le X par *saved*.

Grub sélectionnera automatiquement le système qu'il aura sauvegardé au préalable ;)

---

## Annexes

Le timeout

Une chose qui peut être intéressante est de pouvoir modifier la ligne :

```
timeout 15
```

Cette ligne permet de régler le temps limite pour faire votre choix.

Il est par défaut de 10 et comme vous pouvez le voir, j'ai mis 15 secondes.

Autre chose : vous pouvez désactiver cette fonction en commentant la ligne (mettre un # devant), ainsi :

Citation : menu.lst

```
## # timeout 10
```

Le gris signifie que la ligne est commentée, GRUB ne bootera plus après un certain temps, ce sera donc à vous de taper Entrée.

**ATTENTION !**, Ne mettez SURTOUT PAS

```
timeout 0
```

Vous n'auriez plus aucun temps de battement avant le démarrage. C'est une erreur très fréquente et qui peut être particulièrement désagréable à réparer.

Imaginons que vous choisissiez de faire booter votre système sur Windows. Vous n'auriez alors plus accès à Linux, et sachant que vous ne pouvez modifier votre menu.lst que depuis ce système, cela vous pose donc un gros problème.

Trop tard ? Ne vous inquiétez pas, ce n'est pas dramatique du tout ;) Faites une recherche sur les forums où envoyez moi un petit MP, ou reportez-vous à la partie concernant les erreurs fréquentes (en cours de rédaction actuellement). Un chapitre fera en effet bientôt la liste des erreurs les plus courantes et expliquera comment les rattraper ;)

Les noms des systèmes

Autre chose : il est possible et facile de modifier les noms des systèmes.

Une partie de mon menu.lst :

```
title Ubuntu, kernel 2.6.17-10-generic
root (hd0,4)
kernel /boot/vmlinuz-2.6.17-10-generic root=/dev/sda5 ro quiet splash
initrd /boot/initrd.img-2.6.17-10-generic
quiet
savedefault
boot
```

Il suffit de modifier cette ligne :

**title Ubuntu, kernel 2.6.17-10-generic**

En :

**title Ubuntu**

Pour changer le nom et avoir un menu de boot plus simple.

La protection des entrées

Et oui ! C'est possible ! Grub vous permet de protéger votre système avec un mot de passe supplémentaire.

Et alors ? J'ai déjà un Mdp pour me connecter sur ma machine, moi !

C'est vrai. Néanmoins, ce que vous ne savez peut être pas, c'est qu'en mode recovery, vous avez accès à beaucoup d'éléments et notamment une console en root sans mot de passe ! C'est plutôt risqué !

Pour protéger Grub par un mot de passe, il vous faut déjà ... un mot de passe.

Grub vous propose par ailleurs de le crypter lui-même en md5 :

```
grub> md5crypt
md5crypt
Password: votre mot de passe
Encrypted: $1$gLhU0/$aW78kHK1QfV3P2b2znUoe/
grub> quit
```

Décommentez ensuite la ligne correspondant dans le menu.lst (**en n'oubliant pas de changer le mot de passe encrypté par le vôtre !!!**)

*Citation*

```
## password [--md5] passwd
# If used in the first section of a menu file, disable all interactive editing
# control (menu entry editor and command-line) and entries protected by the
# command 'lock'
# e.g. password topsecret
password --md5 $1$gLhU0/$aW78kHK1QfV3P2b2znUoe/
# password topsecret
```

Pour entrer dans le système, vous devrez entrer votre mot de passe avant de sélectionner l'entrée. Pour cela, tapez simplement sur **p**.

Maintenant vous avez deux possibilités :

Protéger une seule entrée

C'est assez facile, ajoutez simplement lock à la suite de vos options sur le système voulu :

*Citation*

```
title Ubuntu 9.04, kernel 2.6.28-14-generic (recovery mode)
lock
uuid e77d8ae8-36d8-42d1-891d-2cf89dbf0b0c
kernel /boot/vmlinuz-2.6.28-14-generic root=UUID=e77d8ae8-36d8-42d1-891d-
2cf89dbf0b0c ro single
initrd /boot/initrd.img-2.6.28-14-generic
```

Pour être sûr de ne pas vous planter, testez sur un ancien noyau ou un vieux truc histoire de ne pas rester coincé au menu.

Protéger tous les recovery mode

Encore une fois, cela va être assez rapide.

Remplacez :

*Citation*

```
## # lockalternative=false
```

Par :

*Citation*

```
## # lockalternative=true
```

Ainsi, tous vos recovery modes seront protégés et cela même après mise à jour du noyau ;)

Il est vrai que la première version est plus facile à configurer que la dernière, tout tient en un seul fichier (un peu "mastoc" certes :p ).

---

Revision #1

Created 2026-04-24 13:31:54 UTC by Nico là

Updated 2026-04-24 13:32:15 UTC by Nico là