

Création d'une clé USB personnalisée pour un multiboot Linux

Préambule

De nombreux outils permettent de rendre une clé USB bootable tels que Rufus, Balena Etcher, UNetBootin... et certains permettent de créer des multiboot USB tels que Multisystem, Ventoy, Easy2boot...

Préparation de la clé USB

Formater et monter la clé USB

1. Afin de ne pas faire d'erreur, il est nécessaire de cibler le bon support de stockage. Ainsi, il est nécessaire de lister les supports présents via le terminal :

```
lsblk
```

Ceci retourne une liste des périphériques de stockage de masse. Exemple :

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPPOINTS
sda	8:0	0	931,5G	0	disk	
└─sda1	8:1	0	931,5G	0	part	/mnt/997703ee-871d-4254-8133-65c205a019b7
sdb	8:16	1	14,8G	0	disk	
└─sdb1	8:17	1	7,5G	0	part	
nvme0n1	259:0	0	238,5G	0	disk	
└─nvme0n1p1	259:1	0	1M	0	part	
└─nvme0n1p2	259:2	0	513M	0	part	/boot/efi
└─nvme0n1p3	259:3	0	238G	0	part	/

Ici, la clé usb est détectée en sdb. Elle contient une partition sdb1.

Pour rappel :

- Le préfixe "sd" signifie "SCSI disk" et est utilisé pour tous les types de disques, y compris SATA, SSD, et USB.
- La lettre qui suit "sd" indique l'ordre de détection du disque par le système :
 - sda : premier disque détecté
 - sdb : deuxième disque détecté
 - sdc : troisième disque détecté, et ainsi de suite
- Les chiffres après la lettre représentent les partitions sur le disque :
 - sda1 : première partition du premier disque
 - sda2 : deuxième partition du premier disque
 - sdb1 : première partition du deuxième disque, etc.

2. Démonter la clé USB si elle est montée :

```
sudo umount /dev/sdX*
```

Important

Remplacer X par la lettre correspondant à la clé USB. le symbole * permet de cibler toutes les partitions susceptibles d'être montées.

3. Formater la clé USB en créant une table de partition MBR :

```
sudo fdisk /dev/sdX
```

Dans fdisk, taper 'o' pour créer une nouvelle table de partition GPT, puis 'w' pour écrire les changements.

Important

Penser à installer parted si ce n'est pas fait :

```
sudo apt update
sudo apt upgrade -y
sudo apt install parted -y
```

4. Créer les partitions nécessaires (BIOS, EFI, DATA) :

```
sudo parted /dev/sdX
```

Dans parted, exécuter les commandes suivantes :

```
mklabel msdos
mkpart primary fat32 1MiB 100%
set 1 boot on
quit
```

Installer Grub sur le support

1. Monter la partition :

```
sudo mkdir /mnt/usb
sudo mount /dev/sdX1 /mnt/usb
```

2. Installer GRUB pour le mode BIOS :

```
sudo grub-install --target=i386-pc --boot-directory=/mnt/usb/boot --recheck /dev/sdX
```

3. Installer GRUB pour le mode UEFI :

```
sudo grub-install --target=x86_64-efi --efi-directory=/mnt/usb --boot-directory=/mnt/usb/boot --removable --recheck /dev/sdX
```

La clé USB possède à présent tous les dossiers et fichiers nécessaires pour un démarrage.

Créer votre propre thème

Il est possible de créer son propre menu de manière relativement simple. Les opérations suivantes sont facultatives mais il est bon de savoir faire ce genre de personnalisation afin de brander ses propres clé USB avec le nom et le logo d'une marque, d'une organisation...

Création du dossier des thèmes

1. Utiliser le commande mkdir comme suit :

```
sudo mkdir /mnt/usb/boot/grub/themes  
/mnt/usb/boot/grub/themes/#Le_nom_de_votre_theme
```

Important

Remplacer #Le_nom_de_votre_theme par le nom que vous décidez de donner à votre thème

Créer ou choisir une image de fond

Chaque image de fond doivent être dans le format défini du menu. Deux choix s'offrent à nous :

- 640x480 pixels [Exemples d'images sur le site wall2mob.com](https://wall2mob.com)
- 1024x768 [Exemples d'images sur le site wallpaper.mob.org](https://wallpaper.mob.org)

Il est bien évidemment possible de créer son propre fond d'écran avec un logiciel (de préférence de dessin vectoriel) comme Inkscape (Linux et BSD) ou Illustrator (Windows et MacOS).

2. Copier l'image dans le répertoire de votre thème :

```
sudo cp /chemin/de/votre/image /mnt/usb/boot/grub/themes/#le_nom_de_votre_theme
```

Convertir la police de caractère de son choix

Créer sa propre police de caractère se révèle souvent très fastidieux mais les polices de caractère libres de droit courent le web. Plusieurs sites permettent de choisir une police qui nous convient. Quelques exemples :

- [DaFont](https://dafont.com) (Une véritable institution)
- [FontSquirrel](https://fontquirrel.com) (Simple et rapide)
- [FontSpace](https://fontspace.com) (Modernité et élégance)
- [Awwwards](https://www.awwwards.com) (Le site des webdesigners)

Important

Afin de simplifier la suite de ce tutoriel, nous choisirons une police de caractère au format TTF (True type font)

Nous choisissons de convertir la police Ubuntu.ttf

1. Installez la police de caractère :

```
sudo apt-get install ttf-ubuntu-font-family && sudo fc-cache -fv
```

Si vous obtenez un message d'erreur, il est possible qu'elle soit déjà présente sur votre système. Pour le vérifier, vous devrez taper la commande suivante :

```
ls /usr/share/fonts/truetype/ubuntu/
```

📌 Important

Si vous ne la trouvez pas à cet emplacement, essayez de la trouver dans un des emplacements suivants :

1. `/usr/share/fonts/truetype/`
2. `~/local/share/fonts/` (dans votre répertoire personnel)
3. `~/fonts/` (un ancien emplacement, toujours utilisé sur certains systèmes)

2. Copiez la police dans le bon répertoire de notre clé USB. Par exemple :

```
sudo cp /usr/share/fonts/truetype/ubuntu/Ubuntu-B.ttf  
/mnt/usb/boot/grub/themes/votre_theme
```

3. Convertir la police du format TTF à PF2

```
sudo grub-mkfont -s 20 -o /mnt/usb/boot/grub/themes/votre_theme/Ubuntu-B.pf2 -s  
24 /mnt/usb/boot/grub/themes/votre_theme/Ubuntu-B.ttf
```

📌 Note

Les fichiers .pf2 sont des polices bitmap spécialement formatées pour GRUB.

L'option -s 20 définit la taille de la police en sortie. Cette option est facultative.

Penser à supprimer la police ttf du dossier après conversion `sudo rm`
`/chemin/police.ttf`

À présent, nous avons deux éléments visuels fondamentaux mais il nous faut aussi configurer l'agencement des éléments de notre thème.

Créer et adapter le fichier de configuration de son thème

1. Créer le fichier :

```
sudo nano /mnt/usb/boot/grub/themes/votre_theme/theme.txt
```

2. Écrire le fichier comme dans l'exemple suivant :

```
# Définition de l'image de fond  
desktop-image: "splash.png"  
  
# Configuration du titre (vide)  
title-text: ""
```

```
title-font: "Ubuntu-B.pf2"

# Configuration du terminal
terminal-font: "Terminus Regular 14"
terminal-box: "terminal_box_*.png"
terminal-border: "0"

# Configuration du menu
+ boot_menu {
    left = 15%
    top = 20%
    width = 70%
    height = 60%
    item_font = "Ubuntu-B.pf2"
    item_color = "#cccccc"
    selected_item_color = "#ffffff"
    icon_width = 32
    icon_height = 32
    icon_spacing = 20
    item_height = 24
    item_padding = 5
    item_spacing = 10
}

# Titre
+ label {
    top = 8%
    left = 0
    width = 100%
    height = 40
    text = "Clé multiboot de La Capsule"
    font = "Ubuntu-B.pf2"
    color = "#ffffff"
    align = "center"
}

# Pied de page
+ label {
    top = 95%
    left = 0
    width = 100%
    height = 20
    text = "Sélectionner avec Entrée"
    font = "Ubuntu-B.pf2"
    color = "#ffffff"
    align = "center"
}

# Configuration de la barre de progression
+ progress_bar {
    id = "__timeout__"
    left = 15%
    top = 85%
    width = 70%
    height = 10
    fg_color = "#3465a4"
```

```
bg_color = "#ffffff"
border_color = "#cccccc"
}
```

Fonctionnement du menu de démarrage

Création de la configuration

1. Créer un fichier de configuration comme suit :

```
sudo nano /mnt/usb/boot/grub/grub.cfg
```

2. Tapez le texte suivant en prenant en compte le nom de votre theme :

```
set timeout=60
set default=0
set gfxmode=auto

insmod loopback
insmod iso9660
insmod linux
insmod all_video
insmod gfxterm
insmod terminal
insmod font
terminal_output gfxterm

if [ "${grub_platform}" = "efi" ]; then
    # Configuration pour le mode UEFI
    loadfont /boot/grub/themes/LaCapsule/Ubuntu-B.ttf
    set theme=/boot/grub/themes/LaCapsule/theme.txt
    set gfxmode=1920x1080
else
    # Configuration pour le mode BIOS Legacy
    insmod png
    loadfont /boot/grub/themes/LaCapsule/Ubuntu-B.ttf
    set theme=/boot/grub/themes/LaCapsule/theme.txt
    set gfxpayload=800x600x16
fi
```

Nous allons expliquer chaque ligne du texte ci-dessus :

- set timeout=60 # **Définit le délai d'attente avant le démarrage automatique (en secondes)**
- set default=0 # **Définit l'entrée par défaut du menu de démarrage (0 = première entrée)**
- set gfxmode=auto # **Configure le mode graphique automatiquement**
- insmod gfxterm # **Charge le module pour le terminal graphique**
- insmod terminal # **Charge le module pour le terminal**
- terminal_output gfxterm # **Configure la sortie du terminal en mode graphique**
- set gfxmode=1024x768 # **Définit une résolution graphique spécifique (1024x768)**
- set theme=/boot/grub/themes... # **Définit le thème à utiliser pour le menu GRUB**

- loadfont /boot/grub/themes/LaCapsule/Ubuntu-B.pf2 # Définit la police personnalisée

Importer un iso dans notre clé USB

1. Télécharger un fichier iso depuis internet

Rendez-vous à l'adresse suivante :

<https://clonezilla.org/downloads/download.php?branch=stable>

Téléchargez l'iso de la dernière version stable de clonezilla en sélectionnant iso au point 2 "file type" et attendre la fin du téléchargement.

2. Créer un dossier pour stocker les fichiers iso sur la clé usb :

```
sudo mkdir /mnt/usb/iso
```

3. Copier l'iso depuis le dossier des téléchargements vers la clé USB :

```
sudo cp /home/$USER/Téléchargements/clonezilla-live-*-amd64.iso /mnt/usb/iso
```

⚠ Important

Il est important de vérifier le chemin car si la distribution Linux est dans une autre langue, il se peut que le dossier Téléchargements soit nommé autrement, exemple "Downloads".

Intégrer l'ISO dans le menu

1. Ouvrir de nouveau le fichier de configuration :

```
sudo nano /mnt/usb/boot/grub/grub.cfg
```

2. Ajouter le texte suivant en fin de fichier en prenant en compte votre version de clonezilla :

```
menuentry "CLONEZILLA" {  
  set isofile="/iso/clonezilla-live-3.2.0-5-amd64.iso"  
  loopback loop $isofile  
  linux (loop)/live/vmlinuz boot=live union=overlay username=user config components  
  quiet noswap nolocales eed=on nomodeset ocs_live_run=\"ocs-live-general\"  
  ocs_live_extra_param=\"\" keyboard-layouts= ocs_live_batch=\"no\" locales=  
  vga=788 ip=frommedia nosplash toram=live,syslinux,EFI findiso=$isofile  
  initrd (loop)/live/initrd.img  
}
```

Nous allons expliquer chaque argument du texte ci-dessus :

- menuentry "CLONEZILLA" { # Définit le nom de l'entrée dans le menu GRUB
- set isofile="/clonezilla-live-2.8.1-12-amd64.iso" # Le chemin vers l'ISO de Clonezilla
- loopback loop \$isofile # Monte l'image ISO en tant que périphérique loop
- linux (loop)/live/vmlinuz \ # Charge le noyau Linux depuis l'image ISO montée
- boot=live \ # Démarre en mode "live"

- union=overlay \ **# Utilise un système de fichiers en superposition**
- username=user \ **# Définit le nom d'utilisateur par défaut**
- config \ **# Charge la configuration par défaut**
- components quiet \ **# Active le mode silencieux pour réduire les messages de démarrage**
- noswap \ **# Désactive l'utilisation de la partition swap**
- nolocales \ **# Désactive le chargement des locales**
- edd=on \ **# Active la prise en charge EDD (Enhanced Disk Drive)**
- nomodeset \ **# Désactive le réglage automatique du mode vidéo**
- ocs_live_run="ocs-live-general" \ **# Spécifie le script Clonezilla à exécuter**
- ocs_live_extra_param="" \ **# Réserve un espace pour des paramètres supplémentaires**
- keyboard-layouts= \ **# Laisse la disposition du clavier par défaut**
- ocs_live_batch="no" \ **# Désactive le mode batch de Clonezilla**
- locales= \ **# Laisse les locales par défaut**
- vga=788 \ **# Définit le mode vidéo (résolution 800x600)**
- ip=frommedia \ **# Configure l'IP à partir du média de démarrage**
- nosplash \ **# Désactive l'écran de démarrage graphique**
- toram=live,sylinux,EFI \ **# Charge en RAM les composants spécifiés**
- findiso=\$isofile **# Indique à GRUB où trouver l'image ISO**
- initrd (loop)/live/initrd.img **# Charge l'initrd (système de fichiers initial en RAM) depuis l'image ISO**

Sauvegarder la nouvelle configuration avec Ctrl+o appuyer sur Entrée et quitter avec Ctrl+x.

3. Démonter la clé USB :

```
sudo umount /mnt/usb
```

Il ne reste plus qu'à tester la clé USB afin de valider le fait que votre menu est personnalisé et que Clonezilla est bootable.

Importer un iso dans notre clé USB

Comme nous l'avons vu précédemment, il est possible d'importer un fichier iso sur notre clé usb et de lui créer une entrée de menu, permettant à l'utilisateur de booter dessus en choisissant l'entrée.

Nous allons reproduire l'opération avec un fichier iso plus conséquent que celui de clonezilla.

Warning

Il sera nécessaire de remonter la clé USB comme précédemment grâce à la commande mount.

Télécharger un fichier iso

1. Rendez-vous sur le site de Distrowatch : <https://distrowatch.com/>

Pour l'exercice, nous choisirons de [télécharger Rhino Linux](#) qui est basé sur Ubuntu. Une fois le fichier complètement téléchargé, il suffit de le copier dans le dossier "/iso" de notre clé USB et d'ajouter les entrées de menu correctes.

Afin de définir correctement les entrées de menu à ajouter dans le fichier de configuration, il est essentiel de comprendre la structure du live CD (iso). Pour se faire, il est impératif d'inspecter sa structure en le montant et listant ses fichiers.

Important

Dans notre exemple, nous allons monter le fichier en amont pour vérifier que les routes vers les fichiers sont correctes.

```
sudo mkdir /mnt/iso
sudo mount -o loop '/mnt/usb/iso/Rhino-Linux-2025.1-amd64.iso' /mnt/iso
ls /mnt/iso/casper
```

Nous pouvons voir que le kernel se trouve être dans /casper/vmlinuz et le fichier initramfs dans /casper/initrd.lz et non dans /live comme pour notre iso de clonezilla.

2. Nous allons donc insérer notre nouvelle entrée de menu afin de permettre le démarrage du live cd :

```
sudo nano /mnt/usb/boot/grub/grub.cfg
```

3. Nous insérons ensuite notre nouvelle entrée en fin de fichier :

```
menuentry "RHINO" {
    set isofile="/iso/Rhino-Linux-2025.1-amd64.iso"
    loopback loop $isofile
    linux (loop)/casper/vmlinuz boot=casper iso-scan/filename=$isofile noprompt
    noejct
    initrd (loop)/casper/initrd.lz
}
```

4. Nous mettons ensuite à jour le GRUB :

```
sudo update-grub /mnt/usb/boot/grub/grub.cfg
```

Il ne reste plus qu'à tester notre clé USB multiboot afin de valider sa conformité en démarrant depuis un poste.

Importer un fichier ISO de distribution Linux "custom"

Dernière étape pour notre clé USB multiboot, l'import des distributions Linux personnalisées. ([voir le tutoriel](#))

Comme précédemment, il suffira de monter la clé USB et de copier l'ISO dans le dossier /iso ou de copier le contenu de l'ISO dans un dossier au nom explicite. Pour cet exercice, nous copierons le contenu de l'ISO dans un dossier.

Pour se faire, nous devons monter l'ISO et copier son contenu dans un dossier créé en amont. Ceci peut se faire de la manière suivante :

```
sudo mount /dev/sdX1 /mnt/usb # Montage de la clé USB
sudo mount -o /chemin_de_notre_ISO/nom_de_celui-ci.iso /mnt/iso # Montage de l'ISO
sudo mkdir /mnt/usb/nom_de_notre_dossier # Création du dossier sur la clé
sudo cp -r /mnt/iso /mnt/usb/nom_de_notre_dossier # Copie du contenu de l'ISO sur la clé
```

Ceci fait, il sera nécessaire de vérifier le contenu du dossier pour repérer les bonnes routes, comme précédemment dans l'ISO de Rhino Linux.

Une fois les routes clairement repérées, nous pourrions ajouter une nouvelle entrée de menu à notre fichier de configuration :

```
sudo nano /mnt/usb/boot/grub/grub.cfg
```

En y ajoutant les lignes correspondantes aux entrées réelles comme dans cet exemple :

```
menuentry "Ma Distribution personnalisée" {
    insmod linux
    linux /nom_de_notre_dossier/boot/vmlinuz
    initrd /nom_de_notre_dossier/boot/initrfs.img
}
```

Une fois ceci accompli, nous pouvons à nouveau mettre à jour le GRUB et démonter la clé USB :

```
sudo update-grub /mnt/usb/boot/grub/grub.cfg
sudo umount /mnt/usb
```

Nous pouvons à présent effectuer le test en démarrant depuis un poste.