

Tutoriel de création d'une image pour Framakey Linux Mint personnalisée

Table des matières

I But des manipulations.....	2
II Préparation du PC.....	2
III Avant de semer, il faut préparer la terre.....	2
IV Labourons tout ça (préparation de l'image).....	3
IV.1 Création de l'image.....	3
IV.2 Partitionnement de l'image.....	3
IV.3 Installation du MBR de syslinux pour le boot BIOS.....	5
IV.4 Installation de rEFInd pour le boot UEFI.....	5
IV.5 Clôture de l'image.....	5
IV.6 Configuration des 2 systèmes de démarrage (BIOS/UEFI).....	5
IV.6.1 Paramétrage de rEFInd.....	5
IV.6.2 Paramétrage de syslinux.....	6
IV.7 Mettre tout ça au chaud.....	6
V Plantons, mon bon monsieur, plantons !.....	6
V.1 Intégration des fichiers « live ».....	6
V.2 Création du fichier persistant.....	7
VI Mettez de l'engrais et arrosez.....	7
VI.1 Les machines virtuelles, c'est bon mangez-en !.....	7
VI.2 Personnalisation de Linux Mint.....	7
VI.3 Nettoyage de casper-rw.....	8
VI.4 Union des modifications au système de base.....	8
VII Enlevons les mauvaises herbes.....	9
VIII La polyculture, c'est top.....	10
IX Pour aller plus loin.....	11

I But des manipulations

Le but des manipulations suivantes est de créer une Framakey associée à une distribution GNU/Linux. Il pourra s'agir d'une Linux Mint mais aussi d'une Tails ou autre.

L'objectif est de pouvoir partir d'un Live CD d'une distribution, d'y apporter des personnalisations et de recréer un système de fichiers incluant ces dernières.

L'idée est aussi d'assurer le boot via un système indépendant de la distribution utilisée, ce qui permettra, entre autres, de pouvoir utiliser la même méthode pour différentes distros.

Pour ce faire, on utilisera rEFInd pour gérer le boot UEFI et syslinux pour le boot BIOS.

Les essais sur une Linux Mint ayant montré qu'il n'était pas possible d'accéder à la partition contenant le système squashfs du Live en écriture par défaut, ni aux partitions contenant les fichiers de persistance (argh, comprenez que j'y ai perdu beaucoup de temps), il faudra réaliser une modification un peu plus profonde que prévue (modification de casper).

Ce que ce tutoriel n'est pas : une méthode pour se créer un Live USB personnalisé. Si c'est que vous cherchez à faire, privilégiez des outils comme UnetBootin ou LiliUSBCreator avec un fichier persistant.

II Préparation du PC

Vous aurez besoin des outils suivants :

Outil utilisé	Nom du RPM (Fedora)	Nom du DEB (Ubuntu/ Linux Mint)
dd	coreutils	
losetup	utils-linux	mount
udisksctl	udisks2	
gparted	gparted	
rEFInd	http://www.rodsbooks.com/refind/getting.html prendre la version binaire zippée	
syslinux/extlinux	là j'ai la flemme prenez tout ce qui concerne syslinux et extlinux	
Virtualbox	Euh, vous trouverez bien tout seul ?	
fatattr	http://rpm.pbone.net/index.php3/stat/4/idpl/23373945/dir/centos_6/com/fatattr-1.0.1-3.1.x86_64.rpm.html	fatattr
zerofree	zerofree	
funionfs / aufs	funionfs	aufs-tools
squashfs	squashfs-tools	

Droits nécessaires :

Mettez votre utilisateur dans le groupe 'disk' de manière TEMPORAIRE (parce que c'est risqué).

Sous Linux Mint (et vraisemblablement Ubuntu), cela n'a pas l'air de suffire, /dev/loop-control n'appartenant pas au groupe 'disk'. Débrouillez vous avec des sudo.

III Avant de semer, il faut préparer la terre

Ceci n'est aucunement obligatoire mais recommandé.

Créez un dossier de travail contenant les sous-dossiers kionvrien.

Perso, ça donne ça :

FK_Linux

images (contiendra les images disques non compressées)

images_7z (les mêmes mais compressées pour téléversement)

resultat (contiendra les fichiers squashfs et autres, mis à jour)

squashfs (point de montage pour l'union)

refind-bin-0.10.1 (contenu de l'archive téléchargée plus haut, la version peut varier)

IV Labourons tout ça (préparation de l'image)

Plutôt que de travailler sur une clé, on va utiliser une image disque car beaucoup plus rapide (bon c'est aussi plus chiant, je l'avoue :p).

Rien ne vous empêche de faire la préparation directement sur une clé USB.

Et pour vous y aider, vous pouvez même utiliser directement l'image disque compressée de 8Go disponible à l'adresse suivante : https://files.framakey.org/testing/packages/FKMint/image_live_travail_base8G.img.7z
Vous pouvez utiliser la dite image pour sauter toute cette partie de préparation de l'image même si vous envisagez de travailler sur une image plutôt qu'une clé.

L'image en question contient aussi les fichiers de configuration de rEFInd et syslinux pour une Linux Mint et une Tails.

IV.1 Création de l'image

Se placer dans le dossier de travail.

```
$ cd /home/user/FK_Linux/images
```

Créer une image de 8Go avec dd.

```
$ dd if=/dev/zero of=image_live_travail.img bs=1MD count=8000
```

Monter l'image dans un loop (périphérique boucle)

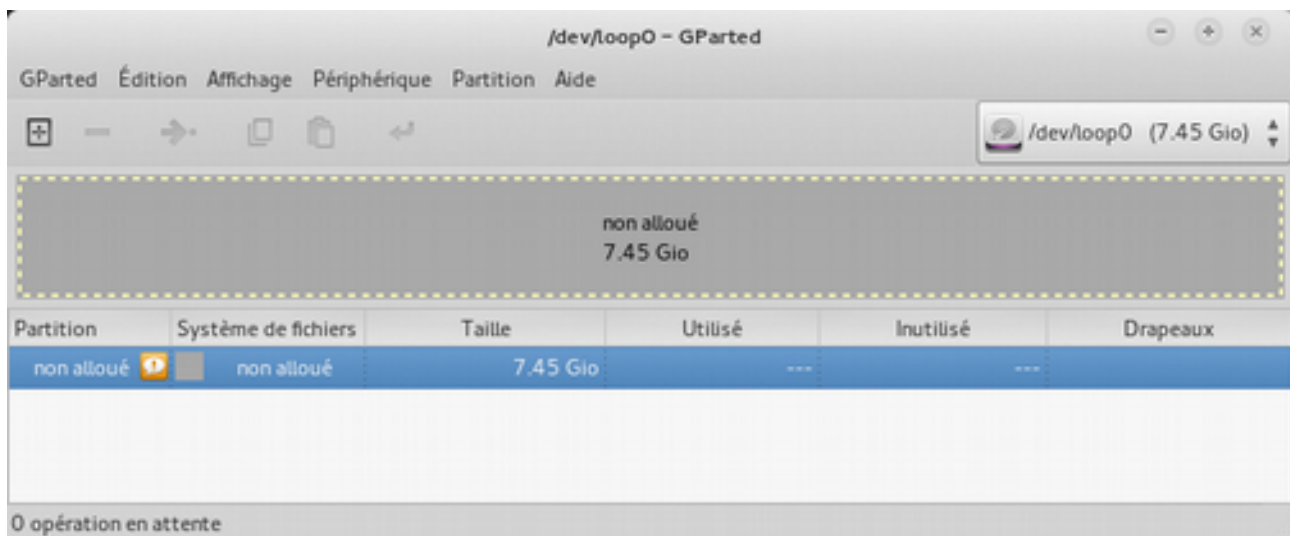
```
$ losetup -f --show -P image_live_travail.img
```

Cela vous retournera le loop utilisé pour info.

IV.2 Partitionnement de l'image

Lancer **gparted** sur le loop créé

```
# gparted /dev/loop0
```



Créer une table de partition **msdos**



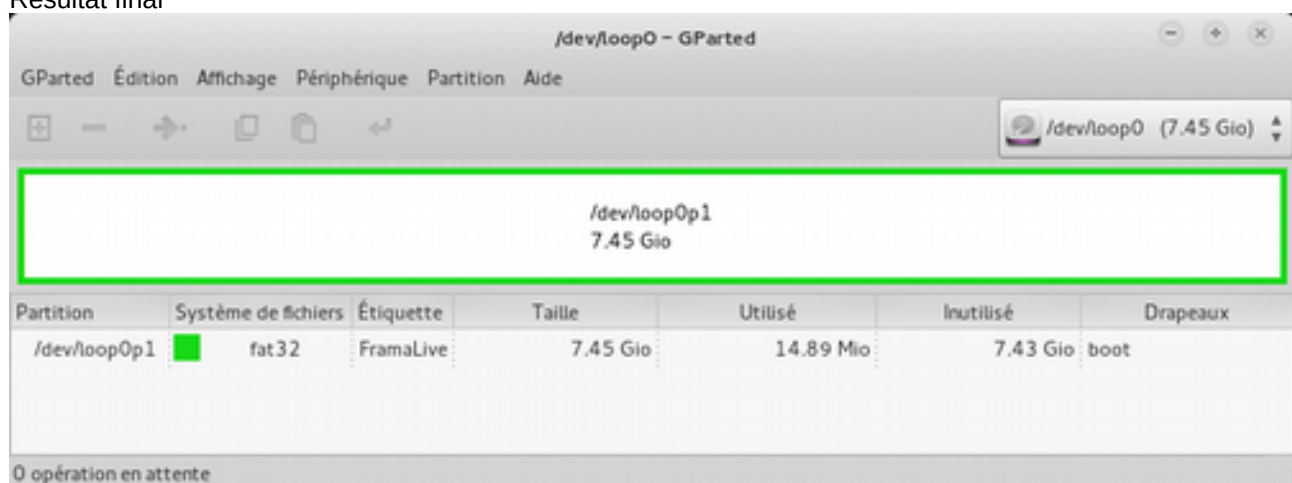
Créer une partition FAT32 avec une étiquette **FramaLive** et occupant tout l'espace disponible (défaut).



Placer un drapeau **boot** sur cette dernière.



Résultat final



Sortir de **gparted**

```
$ ls /dev/loop*
```

devrait vous montrer l'apparition de loop0p1, la partition que l'on vient de créer.

IV.3 Installation du MBR de syslinux pour le boot BIOS

L'installation du mbr (adaptez le chemin du fichier mbr.bin suivant votre cas) se fait sur le périphérique

```
$ dd conv=notrunc bs=440 count=1 if=/usr/share/syslinux/mbr.bin of=/dev/loop0
```

À priori, cela n'est utile qu'avec extlinux (utilisé lors de précédents essais), avec syslinux c'est automatique.

IV.4 Installation de rEFInd pour le boot UEFI

À partir du répertoire de rEFInd, lancer l'installation sur /dev/loop0p1 (correspondant à la partition Framakey)

```
# ./refind-install --usedefault /dev/loop0p1 --alldrivers
```

IV.5 Clôture de l'image

démonter le loop

```
$ losetup -d /dev/loop0
```

On privilégiera **udisksctl** à **losetup** par la suite car il gère comme un grand les points de montage et les partitions (loop*p*).

IV.6 Configuration des 2 systèmes de démarrage (BIOS/UEFI)

Remontons notre image

```
$ udisksctl loop-setup -f image_live_travail.img
```

Miracle, ça crée automatiquement les points de montage (dans /media/user pour Ubuntu/Mint ou /run/media/user pour Fedora) et ça les monte (icône FramaLive dans le gestionnaire de fichiers).

IV.6.1 Paramétrage de rEFInd

Cela se fait au travers du fichier **/EFI/BOOT/refind.conf** du périphérique **FramaLive**, ce dernier est assez bien documenté. L'ajout d'une entrée se fait selon le schéma simple suivant (nécessite un noyau ≥ 3.3.x, cf <http://www.rodsbooks.com/refind/linux.html#efistub>)

```
menuentry "Linux Mint Live" {
    icon EFI/BOOT/icons/os_linuxmint.png
    volume FramaLive
    loader /casper/vmlinuz
    initrd /casper/initrd.lz
    options "file=/cdrom/preseed/linuxmint.seed boot=casper quiet"
    splash --"
}
```

Éventuellement ajouter une bannière personnalisée et/ou des icônes spécifiques.

Créer un fichier **.hidden** à la racine du montage contenant les dossiers à cacher, cela évitera les fausses manipulations et permet de présenter quelque chose de plus propre. N'est utilisé que sous Linux.

Exemple de contenu (masque tout ce qui concerne EFI, syslinux, Linux Mint et Tails)

```
casper
dists
EFI
pool
preseed
syslinux
Recycled
ldlinux.sys
MD5SUMS
README.diskdefines
live
```

IV.6.2 Paramétrage de syslinux

Créer un dossier syslinux dans **FramaLive**, il contiendra les fichiers de configuration et de boot.

Syslinux nécessite que la cible soit démontée pour l'installation (alors qu'extlinux impose qu'elle soit montée). Il faut donc démonter la partition/périphérique **FramaLive** dans le gestionnaire de fichiers, cela démontera et supprimera automatiquement le loop correspondant (magie udiskctl, gênante dans ce cas là). Remonter l'image dans un loop (sans monter la partition qui est dedans)

```
$ losetup -f --show -P image_live_travail.img
```

Installer **syslinux** sur la partition du loop indiqué (loop0 normalement) :

```
# syslinux -i /dev/loop0p1 -d /syslinux
```

Dans le gestionnaire de fichiers, cliquer sur le périphérique **FramaLive** pour le monter.

Copier les modules syslinux nécessaires à un menu graphique à partir de **/usr/share/syslinux** vers le dossier **syslinux** de **FramaLive** :

- libcom32.c32 ;
- libmenu.c32 ;
- libutil.c32 ;
- vesamenu.c32.

Y créer le fichier **syslinux.cfg** qui contiendra la configuration du menu et éventuellement le splash.

IV.7 Mettre tout ça au chaud

Cette étape est facultative mais permet de gagner du temps si vous devez refaire une image.

Dans le gestionnaire de fichiers, démonter la partition/périphérique **FramaLive**, cela démontera automatiquement le loop correspondant (magie udiskctl, cool dans ce cas là).

Compresser **image_live_travail.img** avec votre format préféré pour garder l'image de base au chaud, ça vous évitera de refaire toutes les manips à chaque fois.

En plus ça prend peu de place (<2.5Mo) puisque à part nos fichiers il n'y a que des 0 (/dev/zero).

Pour finir proprement, effacer le loop utilisé :

```
$ udiskctl loop-delete -b /dev/loop0
```

V Plantons, mon bon monsieur, plantons !

V.1 Intégration des fichiers « live »

Récupérer le dernier Live CD de Linux Mint en date :

- <http://www.linuxmint.com/download.php> (prendre la version Cinnamon, no codecs, 64-bit)

Le placer dans le dossier images (cf plus haut).

Si nécessaire, remonter l'image

```
$ udiskctl loop-setup -f image_live_travail.img
```

Monter ensuite l'ISO

```
$ udiskctl loop-setup -f linuxmint-17.3-cinnamon-nocodecs-64bit.iso -r
```

L'option -r permet de s'assurer que le système de fichiers est monté en lecture seule

Copier le contenu de l'ISO sur FramaLive sauf ce qui concerne le boot (dossiers boot, EFI, isolinux). Cela devrait inclure les fichiers et dossiers suivants :

- casper ;
- dists ;
- pool ;
- preseed ;
- .disk, **dossier caché à la racine de l'ISO** ;
- MD5SUMS ;
- README.diskdefines.

L'ISO devrait être dispo dans `/run/media/user/Linux Mint 17.3 Cinnamon 64-bit`.
La partition Framalive devrait être `/run/media/user/Framalive`.
Les chemins sont à adapter en fonction de votre distro et de votre utilisateur.

Démonter l'ISO via le gestionnaire de fichiers, `udisksctl` se chargera du reste.

V.2 Création du fichier persistant

Créer un fichier nommé `casper-rw` sur `/run/media/user/Framalive`

```
$ dd if=/dev/zero of=casper-rw bs=1M count=2000
```

Ne pas hésiter à mettre un gros `casper-rw`, on arrive vite en bout de course.

Le formater en `ext2` avec une étiquette.

```
$ mkfs.ext2 -F -L casper-rw casper-rw
```

VI Mettez de l'engrais et arrosez.

VI.1 Les machines virtuelles, c'est bon mangez-en !

Plutôt que de courir partout après un PC, on va se contenter d'utiliser `VirtualBox`.

Créer un `vmdk` pour accéder en RAW au `loop` qui nous intéresse

```
$ VBoxManage internalcommands createrawvmdk -filename image_live_travail.vmdk  
-rawdisk /dev/loop0
```

Créer une machine virtuelle `kivabien` :

- Debian 64
- 1.5Go de RAM
- 32Mo vidéo
- EFI activé
- presse-papiers bidirectionnel

Définir le `vmdk` précédent (`image_live_travail.vmdk`) comme disque `SATA0`.

Et c'est parti, lancez la machine !!!

VI.2 Personnalisation de Linux Mint

Booster la machine virtuelle, cela doit lancer le Live Linux Mint en mode persistant.
Les commandes suivantes sont bien évidemment à lancer dans le système « live ».

Ajouter un clavier français via le `Control Center` > `Hardware / Keyboard`, si vous le souhaitez.

Optionnellement, ajouter un dépôt pour une version récente du client `ownCloud` (en anglais seulement) :

```
$ echo "deb  
http://download.opensuse.org/repositories/isv:ownCloud:/desktop/Ubuntu_14  
.04/ /" | sudo tee -a /etc/apt/sources.list.d/owncloud.list  
$ wget http://download.opensuse.org/repositories/isv:ownCloud:desktop/Ubuntu_14.0  
4/Release.key -O - | sudo apt-key add -
```

Installer les packs de langues, le client `ownCloud` et mettre à jour `Firefox` & `Thunderbird` :

```
$ sudo apt-get update  
$ sudo apt-get install firefox firefox-locale-fr thunderbird thunderbird-  
locale-fr language-pack-fr language-pack-gnome-fr owncloud-client
```

Lancer la configuration des langues via le `Control Center` > `Preferences / Languages`.

Choisir `France UTF8` pour `Language & Region`.

Cliquer sur le bouton "Apply system-wide".

Cliquer sur "Install/Remove langages".

Sélectionner `French, France` et cliquer sur le bouton "Install language pack".

Modifier le clavier en remplaçant 'us' par 'fr' :

```
$ sudo gedit /etc/default/keyboard
```

Les étapes suivantes sont là par sécurité mais je ne suis pas sûr qu'elles soient utiles.

Se déconnecter.

Laisser la reconnexion auto faire son boulot.

Valider ou pas le renommage des dossiers (peu importe)

Patcher casper et mettre à jour l'initrd. Le but est de faire en sorte que la framakey soit montée en lecture-écriture pour tous les utilisateurs, et non pas seulement pour root.

```
$ sudo gedit /usr/share/initramfs-tools/scripts/casper
```

Modifier les lignes 619 à 622 :

```
if is_supported_fs ${fstype}; then
    devuid=$(blkid -o value -s UUID "$devname")
    [ -n "$devuid" ] && grep -qs "\<$devuid\>" $tried && continue
    mount -t ${fstype} -o ro,noatime "${devname}" $mountpoint || continue
```

par :

```
if is_supported_fs ${fstype}; then
    mount_opts="ro,noatime"
    devuid=$(blkid -o value -s UUID "$devname")
    [ -n "$devuid" ] && grep -qs "\<$devuid\>" $tried && continue
    if [ "${fstype}" = "vfat" -o "${fstype}" = "ntfs" ]; then
        mount_opts="rw,noatime"
        extra_mount_opts=",fmask=111,dmask=0"
    fi
    mount -t ${fstype} -o ${mount_opts}${extra_mount_opts} "${devname}"
    $mountpoint || continue
```

Cela appliquera les masques fmask=111 et dmask=0 au montage de la partition Framalive sur /cdrom, permettant ainsi à tous les utilisateurs de lire-écrire dessus.

Recréer l'initrd :

```
$ sudo /usr/share/casper/casper-update-initramfs -u
```

Les avertissements (cryptsetup ou locale) n'ont pas d'influence donc pas de panique.

Éteindre la machine virtuelle.

VI.3 Nettoyage de casper-rw

Se placer dans le répertoire de montage de l'utilisateur et monter casper-rw :

```
$ cd /run/media/user/
$ udisksctl loop-setup -f Framalive/casper-rw
```

Et oui, on peut monter une image contenue dans un montage. Mieux que les poupées russes !

Dans **casper-rw** (utilité d'avoir mis une étiquette lors de la création, ça crée un point de montage facile à repérer), effacer tout sauf les dossiers **etc**, **usr** et **var**.

Note : suite au passage à Linux Mint 17.3, il faut utiliser le contenu du dossier upper comme racine

Dans **etc**, ne garder que :

- /etc/apt/trusted.gpg
- /etc/apt/sources.list.d/owncloud.list
- /etc/owncloud
- /etc/default/locale
- /etc/default/keyboard

Garder tout **usr SAUF**

- /usr/share/applications/ubiquity.desktop (il m'a agacé celui-là pour le trouver).

Dans **var/cache**, ne garder que /var/cache/apt/*.bin (2 fichiers), supprimer le reste.

Supprimer **var/spool**.

Garder tout **var/lib**.

Dans **var/log**, garder :

- alternatives.log
- dpkg.log
- lastlog
- mintsytem.log
- apt/* (dossier)

VI.4 Union des modifications au système de base

Les actions suivantes sont dérivées de ce tutoriel : <http://jan.varho.org/?p=254>

Le contenu du fichier casper-rw est toujours accessible dans `/run/media/user/casper-rw`.

Monter en lecture seule le système de fichiers compressé squashfs du live.

```
$ udisksctl loop-setup -f Framalive/casper/filesystem.squashfs -r
```

Il se monte dans le dossier disk.

Créer une union de dossiers.

Sous Fedora, utiliser funionfs (nécessite d'être root) :

```
# mount -t fuse -o dirs=disk=RO:casper-rw/upper -o allow_other funionfs#NONE  
/home/user/FK_Linux/squashfs/
```

Sous Ubuntu/Mint, utiliser aufs (nécessite sudo) :

```
$ sudo mount -t aufs -o br:casper-rw/upper:disk none  
/home/user/FK_Linux/squashfs/
```

Recréer le système de fichier squashfs.

```
# mksquashfs /home/user/FK_Linux/squashfs/  
/home/user/FK_Linux/resultat/filesystem.squashfs
```

Recréer le manifest.

```
# chroot /home/user/FK_Linux/squashfs/ dpkg-query -W  
/home/user/FK_Linux/resultat/filesystem.manifest
```

Démonter le tout, dans l'ordre inverse du montage :

```
# umount /home/user/FK_Linux/squashfs/
```

ou

```
$ sudo umount /home/user/FK_Linux/squashfs/
```

Puis, via le gestionnaire de fichiers :

- disk ;
- casper-rw.

Remonter le nouveau filesystem.squashfs pour mettre à jour le fichier filesystem.size.

```
$ udisksctl loop-setup -f /home/user/FK_Linux/resultat/filesystem.squashfs -r
```

Recalculer la taille (nécessite d'être root ou sudo pour accéder à la totalité du contenu).

```
# printf $(du -sx --block-size=1 /run/media/user/disk/ | cut -f1) >  
/home/user/FK_Linux/resultat/filesystem.size
```

Puis le démonter via le gestionnaire de fichiers.

Effacer le fichier `Framalive/casper/filesystem.squashfs` pour éviter un message d'erreur d'espace disque insuffisant.

Transférer les 3 fichiers filesystem.* du dossier `resultat` vers `Framalive/casper/`.

VII Enlevons les mauvaises herbes

Il reste des traces à nettoyer aussi bien pour le fonctionnement normal que pour l'objectif qui est de créer une image la plus petite possible pour un transfert par le Net.

Recréer le fichier casper-rw.

```
$ dd if=/dev/zero of=/run/media/user/FramaLive/casper-rw bs=1M count=2000
```

Puis le reformater.

```
$ mkfs.ext2 -F -L casper-rw /run/media/user/FramaLive/casper-rw
```

Optionnellement, créer un fichier home-rw pour séparer la configuration utilisateur de la configuration système (enregistrée dans casper-rw).

```
$ dd if=/dev/zero of=/run/media/user/FramaLive/home-rw bs=1M count=512
```

Le formater en ext2.

```
$ mkfs.ext2 -F -L home-rw /run/media/user/FramaLive/home-rw
```

La partie Linux Mint est propre, le fichier casper-rw aussi et ils seront compressés au maximum.

VIII La polyculture, c'est top

La partie GNU/Linux est terminée, reste à s'occuper de la partie Framakey.

Le plus simple est d'avoir une Framakey fonctionnelle, il suffit alors de recopier ses fichiers vers **/run/media/user/FramaLive**.

Si ce n'est pas le cas, vous aurez besoin de **wine**, ou de Windows et de **ImDisk Virtual Disk Driver** (permet de monter une image sous windows comme losetup).

ImDisk est dispo là : <http://www.ltr-data.se/opencode.html/#ImDisk>

Récupérer la dernière version de la Framakey : <https://framakey.org/Pack/Pack20Base>

La décompresser dans **FramaLive** avec le gestionnaire d'archives, prendre tout ce qui est dans **\$OUTDIR** sauf **\$PLUGINS**.

Ouvrir un terminal, se placer dans le dossier de **SynApps** et lancer **SynApps**.

```
$ cd /run/media/user/FramaLive/Framakey/SynAppsPortable/App/SynApps/  
$ python main.py
```

Installer les paquets désirés. Pour information, voici la liste des paquets que j'ai utilisé :

7-zipPortable	GreenshotPortable	NVDAPortable
AudacityPortable	HandbrakePortable	ownCloudPortable
CDexPortable	InfraRecorderPortable	PengupopPortable
ClavicomNGPortable	InkscapePortable	PidginPortable
CoolplayerPortable	JitsiPortable	PokerTHPortable
FileZillaPortable	KeePassPortable	SudokuPortable
FramabirdPortable	KeyNotePortable	SumatraPDFPortable
FramafoxPortable	KittyPortable	VirtualMagnifyingGlassPortable
FramakeyToolboxPortable	LibreOfficePortable	VLCPortable
FreeOTFEPortable	MeldPortable	
GIMPPortable	Notepad++Portable	

Vérifier s'il y a des mises à jour disponibles pour les paquets déjà installés. Si besoin, les mettre à jour puis quitter **SynApps**.

Avec **wine**, lancer **FramaUpdate** pour mettre à jour les menus.

Masquer les fichiers et dossiers inutiles pour l'utilisateur final (ex : le dossier EFI).

```
$ cd /run/media/user/FramaLive/  
$ fatattr +h EFI
```

Pour « zéroifier » la partition FAT32 sans être coincé par les limitations de ce système de fichiers, il va falloir utiliser un petit script que voici :

```
#!/bin/bash  
# Write files filled with zero-bits to the directory passed as argument.  
# The script stops, when the filesystem is full.  
# Don't use on your root-filesystem.  
  
if [ $# -ne 1 ]; then  
echo "Usage: $(basename $0) directory-for-zero-bit-files" 1>&2  
exit 234  
fi
```

```

declare -r zeroFilename='deleteme'
declare targetDirectory="$1"

# Remove all trailing slashes, if any.
targetDirectory=$(echo $targetDirectory | sed "s/\/*$//")

counter=0
writeResult='File size limit exceeded'

while [ -n "$(echo "$writeResult" | fgrep 'File size limit exceeded')" -o -n "$(echo "$writeResult" | fgrep 'File too large')" ]; do
  ((counter++))
  writeResult=$(LANG=C dd if=/dev/zero of="${targetDirectory}/${zeroFilename}.${counter}" 2>&1)
  echo $writeResult
done

while [ $counter -gt 0 ]; do
  rm -f "${targetDirectory}/${zeroFilename}.${counter}"
  ((counter--))
done

exit

```

À enregistrer sous un nom quelconque, perso je l'ai appelé **zerowipedir.sh**.

Le lancer sur la partition **FramaLive** :

```
$ ./zerowipedir.sh /run/media/user/FramaLive/
```

C'est fini !

Démonter **FramaLive**.

Il ne reste qu'à compresser **image_live_travail.img** et à l'uploader.

IX Pour aller plus loin

Il est possible d'ajouter un Live Tails très simplement :

- télécharger Tails => <https://tails.boum.org/download/index.fr.html> ;
- monter l'ISO
- copier le dossier live de l'ISO vers Framalive
- activer les menus préconfigurés dans Framakey/EFI/BOOT/refind.conf (en commentant la ligne *disabled* de l'entrée de menu idoine) et Framalive/extlinux/extlinux.conf (en décommentant l'entrée de menu)

La copie de l'image vers une clé USB devrait la laisser bootable. Mais si vous devez redéfinir les partitions, par exemple pour utiliser une clé de 16Go au lieu de 8, je ne garantis pas que ça fonctionne. Au pire, il suffira de relancer l'installation du bootloader syslinux plus éventuellement la copie du mbr.bin.

Pour le boot UEFI, il n'y a rien à faire.

Ce document est long, mais c'est parce que j'ai essayé de décrire le plus précisément possible toutes les étapes. En pratique, il ne faut pas très longtemps pour faire toutes les manipulations. Ce sont principalement les opérations de copie et de « zéroification » qui prennent du temps puisque ce sont celles qui font apparaître les limites du support physique.

À titre de comparaison, j'ai atteint facilement des vitesses d'écriture de l'ordre de 40Mio/s avec une image (sauf pour les innombrables petits fichiers de la Framakey) alors que j'ai souvent stagné entre 4 et 8 Mio/s en travaillant directement sur une clé USB ou une carte SD.

Les opérations de compression (création du squashfs, compression de l'image) peuvent également prendre du temps. Sur mon PC, il m'a fallu 10-12 minutes pour la création du filesystem.squashfs et 15-20 minutes pour la compression finale. Pour cette dernière, il est inutile de choisir une compression « ultra », cela ne fait que rallonger le temps d'exécution sans rien apporter au final, les fichiers présents dans l'image étant déjà très souvent compressés.