

Debian live avec live-build

- Objet : création de votre distribution personnelle basée sur Debian.
- Niveau requis :
[débutant, avisé](#)
- Débutant, à savoir : [Utiliser GNU/Linux en ligne de commande, tout commence là !](#) 😊
- Suivi :
 - Création par [deuchdeb](#) le 30/08/2011
 - Mise à jour par [arpinix](#) le 20/09/2024
 - Vérifié le ...
 - Testé par
- Commentaires sur le forum : [C'est ici](#)¹⁾

Introduction

Live-build est un programme qui permet de créer un Live à partir d'un système développé en chroot (le système dans le système chroot= change root).

Un Live installable est un système embarqué utilisable sur CDRom/DVD/USB et installable sur tout support de stockage (clé USB, carte SD, disque dur interne). Il se compose d'un **squashfs** (une grosse archive) qui contient l'intégralité du système, un **bootloader** ([GRUB](#)) pour pouvoir démarrer et accessoirement un **installateur** (ce sera la cas dans notre exemple).

Le bootloader démarre et lance la décompression du squashfs afin de le rendre utilisable. Dès lors, le Live inséré se comporte comme un système d'exploitation classique.

Dans ce wiki, nous présenterons les options de base ainsi qu'un exemple complet de construction d'un live.

Installation

```
apt update && apt install live-build
```

Le programme live-build se compose de 3 commandes principales à lancer depuis le répertoire de construction du Live :

- `lb config` : configurer les options de votre système live.
- `lb build` : fabriquer votre système live à l'aide des dossiers et fichiers construits avec `lb config`.
- `lb clean` : nettoyer les dossiers et les fichiers créés avec `lb build` pour pouvoir repartir à neuf et construire un nouveau système live à partir de zéro.

Principe de construction du Live

Pour construire un Live, placez-vous dans votre répertoire de travail (`$HOME/ma_debian_perso` par exemple). Vous pouvez alors lancer les commandes `lb config` les unes après les autres afin de

configurer le Live ou utiliser un **script de configuration** (à créer dans `$HOME/ma_debian_perso/auto/config`) qui rassemble toutes les options. Un exemple complet de script est fourni plus bas.

Une fois votre Live configuré, lancez la commande `lb build` en root. Cette commande va générer un fichier ISO ou IMG transférable sur CD/DVD/USB pour une utilisation en Live.

Rappel : procédure de transfert de l'ISO sur USB

Le transfert se fera via le Terminal, c'est la méthode recommandée. Toutefois, une [méthode graphique](#) est disponible sur ce wiki. Vous pouvez aussi utiliser l'application multi-plateforme [Etcher](#).

Pour commencer, branchez votre clé USB, puis lancez un terminal en mode administrateur "root". Nous allons identifier la clé USB à utiliser grâce à la commande `blkid` qui vous donnera un résultat du type :

```
blkid
/dev/sda1: LABEL="system" UUID="3d378712-1b6e-4f66-b9e8-2a6673c62199"
TYPE="ext4"
/dev/sda5: UUID="65bdec62-8d0e-49ca-b70b-c99340e4ee5e" TYPE="swap"
/dev/sdb1: UUID="F9B8-E691" TYPE="vfat"
```

Ici, notre clé est identifiée comme `UUID="F9B8-E691"`, est formatée en "vfat" et contient la partition `sdb1`. Notez bien ce `sdb1` pour ne pas, par erreur, effacer une partition de votre disque dur interne (ici `sda1`).

Placez-vous dans le dossier contenant votre ISO (à adapter à votre configuration) :

```
cd $HOME/ma_debian_perso/
```

Transférer le contenu de l'ISO sur la clé USB grâce à la commande "dd". Prenez bien soin de nommer la clé USB "sdb" et pas `sdb1`, car c'est le disque qui compte, pas la partition et changez "ma_debian.iso" par le nom de votre Live :

```
dd if=ma_debian.iso of=/dev/sdb bs=4M status=progress; sync
```

Le temps de transfert sur votre clé USB dépend de la taille de l'ISO et du taux de transfert de votre port USB. Cette opération peut durer 10 à 15 minutes (la progression de la copie s'affiche dans le terminal). Le terminal vous "rendra la main" une fois le transfert terminé.

Présentation des principales options de configuration

La commande `lb config` permet de configurer les options principales du Live, l'architecture, les paquets intégrés, la destination du Live (DVD, USB) etc. Pour une liste exhaustive des options disponibles, n'hésitez pas à consulter le man `man lb config`.

Voici quelques options à utiliser :

- Construire un Live en amd64 (par défaut, live-build construira un Live sous l'architecture du système hôte) :

```
lb config --architecture amd64
```

- Construire un live basé sur Debian Bookworm :

```
lb config --distribution "bookworm"
```

- Intégrer les dépôts mentionnés :

```
lb config --archive-areas "main contrib non-free non-free-firmware"
```

- Construire un Live au format polyvalent utilisable sur DVD ou USB (pour une liste des formats, man `lb config`) :

```
lb config --binary-images "iso-hybrid"
```

- Prendre en compte les paquets recommandés lors de la construction. Mettre sur "false" produira un Live plus léger mais il faudra bien vérifier les dépendances lors du choix des paquets à intégrer :

```
lb config --apt-recommends "true"
```

- Intégrer un installateur. L'option `live` installera la distribution perso configurée présente sur le Live, `none` supprimera la possibilité d'installer le système, `netinst` ajoutera [l'installateur par le réseau](#) classique :

```
lb config --debian-installer "live"
```

Plus d'options et d'explications dans l'exemple à suivre...

Exemple de construction d'un Live Debian personnalisé

Pour illustrer ce wiki, nous allons construire un live Debian personnalisé utilisant XFCE comme environnement, accompagné d'une série d'applications choisies pour une utilisation basique. Cet exemple se base sur la construction de [Dfiso](#) dont les sources sont [disponibles sur GIT](#).

Mise en place de l'environnement de travail

Pour commencer, nous allons mettre en place un répertoire de travail et y placer les dossiers et fichiers nécessaires à la construction d'un Live. Les commandes qui suivent sont à adapter à votre système et aux souhaits pour votre future distribution personnelle : pensez à vérifier les commandes avant de les exécuter.

Création du répertoire de travail et mise en place des principaux dossiers :

```
mkdir -p $HOME/ma_debian_perso/auto $HOME/ma_debian_perso/config
cd $HOME/ma_debian_perso/auto
cp /usr/share/doc/live-build/examples/auto/* ./
```

Deux dossiers principaux sont créés dans votre répertoire de travail :

- Le dossier **auto** contient les 3 scripts appelés par live-build :
 - auto/build : le script de construction qui lit le script de config
 - auto/clean : le script de nettoyage du dossier de build
 - auto/config : le script principal définissant les options de constructions
- Le dossier **config** accueillera la personnalisation de votre système.

Configuration générale du Live

La configuration du Live se situe dans le script `$HOME/ma_debian_perso/auto/config`. Pour l'exemple, voici un script dérivé de celui utilisé pour construire [DFiso](#), la Debian préconfigurée par Debian-Facile :

config

```
#!/bin/sh

set -e

lb config noauto \
  --mode "debian" \
  --system "live" \
  --distribution "bookworm" \
  --architecture "amd64" \
  --archive-areas "main contrib non-free non-free-firmware" \
  --security "true" \
  --updates "true" \
  --backports "false" \
  --binary-images "iso-hybrid" \
  --bootloaders "syslinux grub-efi" \
  --apt-indices "true" \
  --apt-recommends "true" \
  --apt-secure "true" \
  --apt-source-archives "false" \
  --linux-package "linux-image linux-headers" \
  --debian-installer "live" \
  --debian-installer-distribution "bookworm" \
  --debian-installer-gui "true" \
  --firmware-binary "true" \
  --firmware-chroot "true" \
  --iso-publisher "Projet DFiso; https://debian-facile.org/dflinux; contact@debian-facile.org" \
  --memtest "none" \
  --win32-loader "false" \
  --clean \
  --debug \
  --verbose \
  --source "false" \
  "${@}
```

Ce script rassemble les options pour `lb config`, il sera appelé lors du lancement de la commande

lb build.

Explications des options présentées :

- --system "live" : pour indiquer que le système généré sera utilisable en session "live"
- --distribution "bookworm" : pour indiquer la distribution sur laquelle se base le Live
- --architecture "amd64" : pour indiquer l'architecture du futur Live
- --archive-areas "main contrib non-free non-free-firmware" : pour indiquer les dépôts actifs lors de la construction
- --security "true" : pour indiquer l'utilisation des dépôts "security"
- --updates "true" : pour intégrer les dépôts "updates" dans les sources du live
- --backports "false" : pour ne pas utiliser les dépôts "backports"
- --binary-images "iso-hybrid" : pour générer une ISO polyvalente utilisable sur CD/DVD/USB
- --bootloaders "syslinux grub-efi" : pour indiquer les 2 bootloaders en cas d'utilisation sur un système BIOS (syslinux) ou UEFI (grub-efi)
- --apt-indices "true" : pour activer les informations apt
- --apt-recommends "true" : pour utiliser les paquets recommandés par apt
- --apt-secure "true" : pour vérifier les signatures des paquets utilisés
- --apt-source-archives "false" : pour ne pas intégrer les "deb-src" dans les dépôts du système
- --linux-package "linux-image linux-headers" : pour identifier le noyau utilisé et si les "headers" du noyau doivent être intégrés
- --debian-installer "live" : pour installer le système préconfiguré lors de l'installation
- --debian-installer-distribution "bookworm" : pour indiquer la distribution de l'installateur
- --debian-installer-gui "true" : pour indiquer l'utilisation d'un installateur graphique
- --firmware-binary "true" : pour ajouter les micrologiciels dans le Live
- --firmware-chroot "true" : pour ajouter les micrologiciels dans le système installé
- --iso-publisher "Projet DFiso; <https://debian-facile.org/dflinux>; contact@debian-facile.org" : pour identifier le constructeur de l'ISO
- --memtest "none" : pour ne pas intégrer "memtest"
- --win32-loader "false" : pour ne pas intégrer le lanceur microsoft
- --clean : pour nettoyer les dossiers vides lors de la construction
- --debug : pour générer les plus d'infos possible pour les 'log'
- --verbose : pour augmenter les retours sur le 'log'
- --source "false" : pour ne pas générer les sources complètes du build (avec les sources de tous les paquets utilisés)

Les deux autres scripts présents dans auto/, "build" et "clean" sont courts et basiques :

Le script de build indique le nom du fichier de 'log' à consulter après construction :

build

```
#!/bin/sh

set -e

lb build noauto "${@}" 2>&1 | tee build.log
```

Le script de nettoyage indique les dossiers et fichiers à supprimer lors de la commande `lb clean` (à lancer en root) :

clean

```
#!/bin/sh

set -e

lb clean noauto "${@}"

rm -f config/binary config/bootstrap config/chroot config/common
config/source
rm -f build.log
```

Choix des paquets à installer

Comme vous pouvez le constater, le script principal du build `auto/build` ne mentionne aucun paquet hormis le noyau "linux-image" et les "linux-headers". Les paquets sont installés directement depuis une liste, depuis un dossier dédié ou depuis un script additionnel placé dans les "hooks". Nous allons détailler chaque méthode.

Ajout de paquets .deb depuis une liste

C'est la méthode par défaut. Il suffit de placer une liste de paquets dans le fichier `config/packages-lists/maliste.chroot` ("maliste" ou ce que vous voulez tant que vous placez ".chroot" en extension) pour qu'elle soit automatiquement prise en compte par live-build. Cette liste est lue par apt : vous n'avez pas à vous soucier des dépendances, elles seront automatiquement ajoutées. Les paquets doivent être séparés par un espace, les commentaires (#) sont autorisés.

Voici la liste des paquets pour DFiso :

maliste.chroot

```
## dfiso packages ##
#base
task-french-desktop task-french lsb-release acpi accountsservice user-
setup bash bash-completion command-not-found
#login
lightdm lightdm-gtk-greeter lightdm-gtk-greeter-settings mugshot
#desktop & xfce4-apps
xfce4 xfce4-terminal xfce4-notifyd xfce4-settings xfce4-taskmanager
xarchiver xfburn xfce4-goodies xfce4-power-manager tlp
#filer xtra
thunar-archive-plugin tumbler-plugins-extra
#network
hexchat firefox-esr firefox-esr-l10n-fr webext-https-everywhere webext-
```

```
ublock-origin-firefox thunderbird thunderbird-l10n-fr
transmission-gtk network-manager-gnome curl wpasupplicant
#office
libreoffice libreoffice-gtk3 gnote gnome-calculator mousepad atril
hplip-gui fbreader
cups-pdf system-config-printer simple-scan xournal pdfarranger
#media
vlc asunder lame alsa-utils alsa-tools rhythmbox gstreamer1.0-plugins-
bad gstreamer1.0-plugins-ugly
pavucontrol sox mpg123 vorbis-tools flac wavpack
#graphics
ristretto shotwell gimp cheese
#games
aisleriot crack-attack four-in-a-row gnome-2048 gnome-chess gnome-mines
gnome-nibbles gnome-sudoku quadrapassel gnome-mahjongg
#system
vrms synaptic gdebi gnome-system-tools dkms gnome-disk-utility gdisk
dosfstools ntfs-3g mtools e2fsprogs gnome-software
#flatpak
flatpak gnome-software-plugin-flatpak
#tools
catfish zenity zip p7zip-full most baobab dvd+rw-tools xdotool eject
deja-dup blueman hardinfo
#misc
gvfs gvfs-fuse gvfs-backends x11-utils x11-apps numlockx ntp
#theme
gnome-brave-icon-theme arc-theme numix-gtk-theme numix-icon-theme
numix-icon-theme-circle
#qt
qt5-style-plugins
#nonfree
alsa-firmware-loaders amd64-microcode atmel-firmware bluez-firmware
dahdi-firmware-nonfree firmware-linux firmware-linux-nonfree
firmware-misc-nonfree firmware-amd-graphics firmware-atheros firmware-
ath9k-htc firmware-b43-installer firmware-b43legacy-installer
firmware-bnx2 firmware-bnx2x firmware-brcm80211 firmware-cavium
firmware-ipw2x00 firmware-iwlwifi firmware-intel-sound
firmware-intelwimax firmware-ivtv firmware-libertas firmware-myricom
firmware-netronome firmware-netxen firmware-qlogic firmware-realtek
firmware-samsung firmware-siano firmware-ti-connectivity firmware-
zd1211 hdmi2usb-fx2-firmware intel-microcode iucode-tool
ttf-mscorefonts-installer ttf-xfree86-nonfree unrar
```

Pour éviter les doublons, vous pouvez vérifier la liste des dépendances de vos paquets depuis le gestionnaire de paquets Synaptic en graphique ou depuis un terminal avec apt-rdepends:

```
apt update && apt install apt-rdepends
```

```
apt-rdepends mon_paquet
```

Ajout de paquets .deb depuis le dossier dédié

Un autre moyen d'ajouter des paquets à votre liveCD est de littéralement les "coller" dedans. En effet, tous les paquets de type debian (.deb) placés dans le dossier `config/packages/` seront installés dans le Live.

Attention : cette procédure n'est pas effectuée par `apt` mais par `dpkg`. Il n'y a donc pas de prise en charge des dépendances : prenez soin de les ajouter dans le même dossier, ou de les ajouter dans la liste des paquets à installer (voir section précédente sur les `package-lists`).

Ajout de paquets .deb depuis les "hooks"

Les scripts du type "`monscript.chroot`" placés dans le dossier `config/hooks/normal/` seront exécutés dans le chroot avant la compression en `squashfs`. Vous pouvez donc faire quasiment ce que vous voulez via ces scripts, dont l'ajout de paquet. Un exemple avec l'installation et la configuration de `libdvd-pkg` qui permettra la lecture des DVD commerciaux sur votre futur système :

[libdvdcss2.chroot](#)

```
#!/bin/sh

set -e

# installation de libdvd-pkg
apt install libdvd-pkg

# compilation de libdvdcss2 à partir des sources debian
export DEBIAN_FRONTEND=noninteractive
dpkg-reconfigure libdvd-pkg
```

Personnalisation

Nous avons vu que la construction d'un Live s'effectuait dans un chroot, un sous-système dans votre système. Nous avons vu que vous pouviez agir sur ce chroot grâce aux options du `live-build`, mais également depuis les scripts `hooks` qui modifient le chroot de l'intérieur avant sa compression dans le `squashfs`.

Nous allons voir comment remplacer ou ajouter des fichiers dans le chroot avant même le processus de construction.

Personnalisation du système

La modification directe du système se fait via le dossier `config/includes.chroot`. Tout ce qui sera dans ce dossier se retrouvera dans le chroot, donc dans le `squashfs`, donc dans le Live.

L'arborescence et les droits des ajouts seront conservés lors de la construction.

Vous voulez votre `theme_perso` disponible dans le Live ? Il suffit de créer un dossier `config/includes.chroot/usr/share/themes/` et d'y coller votre dossier `theme_perso`.

Vous voulez vos fonds d'écran favoris dans le Live ? Il suffit de créer un dossier `config/includes.chroot/usr/share/backgrounds/` et d'y coller vos images. etc.

Personnalisation de l'utilisateur

La modification de la configuration de l'utilisateur se fait via le dossier `config/includes.chroot/etc/skel/` aka "skeleton" : tout ce que vous collez dans ce dossier se retrouvera dans le `$HOME/user/` de votre Live et dans le `$HOME/mon_login/` de votre système installé. Très pratique pour ajouter votre `~/ .bashrc` préparés aux petits oignons : collez-le dans `config/includes.chroot/etc/skel/.bashrc`.

De la même façon, vous pouvez ajouter de la documentation pour le futur utilisateur dans `config/includes.chroot/etc/skel/Documents/ma_doc.pdf`

Limites à la personnalisation

Si vous ajoutez dans le dossier `config/includes.chroot/`, un élément qui appartient déjà à un paquet, il sera écrasé par le paquet original lors du build. Par exemple, si vous voulez coller votre version de "ffmpeg" dans `config/includes.chroot/usr/bin/ffmpeg`, il sera écrasé par le vrai "ffmpeg" lors de la prochaine mise à jour. Vous pouvez coller le vôtre dans `config/includes.chroot/usr/local/bin/ffmpeg` : il ne sera pas écrasé et vous pourrez vous en servir sans soucis depuis un alias approprié,

Construction du Live

Vous avez désormais un dossier de build prêt à l'emploi pour construire une Debian personnalisée. Une fois vos fichiers configurés et vos dossiers complétés, vous pouvez lancer la construction de votre Live depuis le répertoire de travail :

```
cd $HOME/ma_debian_perso
```

```
lb build
```

Vous pouvez aller boire un thé ou retaper votre chaise de jardin... la construction d'un Live dépend de votre système et des paquets à installer et peut prendre entre 20 minutes et 2 heures...

Au final, un fichier `$HOME/ma_debian_perso/live-image-amd64.hybrid.iso` sera généré : Félicitations, c'est votre Live ! Il ne reste plus qu'à le renommer et à la transférer sur un DVD ou une clé USB.

Tests et corrections

Pour la phase de test, je vous conseille d'installer votre distro sur une machine virtuelle afin de pouvoir noter vos dernières commandes qui seront à répercuter dans le dossier de build.

Conseils pour un bon debug :

- Testez chaque application en condition réelle : ne faites pas qu'ouvrir votre navigateur, naviguez avec. N'ouvrez pas simplement gimp, éditez, enregistrez et imprimez une image.
- Lancez les applications depuis un terminal afin d'observer les erreurs éventuelles et notez les modifications qui s'imposent (ajout d'un gtk-engine pour le rendu graphique, d'une librairie manquante...)
- Lancez un update/upgrade et vérifiez les paquets recommandés non-installés pour détecter d'éventuels petits outils manquants
- Côté préférences utilisateur : je vous conseille de peaufiner vos réglages personnels, puis d'archiver tout ça pour la dernière étape et la construction finale.

Modification du build > build > tests > répercussion des modifications > nettoyage > build ... ce processus devra être répété pour parvenir à la distro de vos rêves.

Finalisation de l'image ISO

Vous avez un Live exploitable et personnalisé, il reste à peaufiner l'environnement de démarrage et l'installateur. Ça se passe dans le dossier `config/includes.binary/`. Dans notre exemple (rappel: l'exemple de ce wiki utilise [les fichiers de DFiso](#)), le Live généré sera en français, utilise un clavier "fr" et fixe le nom de l'utilisateur (humain) et le nom d'hôte (DFiso) : à adapter à votre configuration.

Configuration du menu de lancement BIOS

Le fond d'écran et le texte qui s'affichent au lancement du Live en version BIOS se configurent dans le dossier `config/includes.binary/isolinux`.

```
cd ~/ma_debian_perso/config/includes.binary/isolinux
ls
install.cfg  live.cfg  menu.cfg  splash.png  stdmenu.cfg
```

Le fond d'écran se nomme `splash.png` et doit faire 640×480 px en PNG. Pour les fichiers texte, voici ceux de DFiso :

- `config/includes.binary/isolinux/menu.cfg` : titre du menu de démarrage et liste des fichiers appelés

[menu.cfg](#)

```
menu hshift 0
menu width 82

menu title Debian 11 par debian-facile : testez ou installez ...
include stdmenu.cfg
include live.cfg
include install.cfg
menu clear
```

- `config/includes.binary/isolinux/stdmenu.cfg` : configuration graphique du menu et label de fin

stdmenu.cfg

```
menu background splash.png
menu color title      * #FFFFFF *
menu color border     * #00000000 #00000000 none
menu color sel        * #ffffff #5258b8b8 *
menu color hotsel     1;7;37;40 #ffffff #5258b8b8 *
menu color tabmsg     * #c1ccdccc #00000000 *
menu color help       37;40 #ffdddd00 #00000000 none
menu vshift 12
menu rows 10
menu helpmsgrow 15
# The command line must be at least one line from the bottom.
menu cmdlinerow 16
menu timeoutrow 16
menu tabmsgrow 18
menu tabmsg ENTER pour lancer -- TAB pour editer
```

- config/includes.binary/isolinux/live.cfg : les entrées de lancement de la session live

live.cfg

```
label live
    menu label Tester Debian 11
    menu default
    linux /live/vmlinuz
    initrd /live/initrd.img
    append boot=live components quiet splash acpi_osi=Linux
    username=humain hostname=DFiso locales=fr_FR.UTF-8 keyboard-layouts=fr
    timezone=Europe/Paris

label live-failsafe
    menu label Tester Debian 11 (mode sans echec)
    linux /live/vmlinuz
    initrd /live/initrd.img
    append boot=live components noapic noapm nodma nomce nolapic
    nomodeset nosmp nosplash vga=normal username=humain hostname=DFiso
    locales=fr_FR.UTF-8 keyboard-layouts=fr timezone=Europe/Paris
```

- config/includes.binary/isolinux/install.cfg : les entrées d'installation du Live

install.cfg

```
label classic
    menu label Installation classique
    linux /install/gtk/vmlinuz
    initrd /install/gtk/initrd.gz
    append video=vesa:ywrap,mtrr vga=788 locale=fr_FR.UTF-8 keyboard-
configuration/layoutcode=fr keyboard-configuration/variantcode=latin9
```

```
keyboard-layout=fr file=/cdrom/install/df-preseed.cfg -- quiet

label expert
    menu label Installation en mode expert
    linux /install/vmlinuz
    initrd /install/initrd.gz
    append vga=788 priority=low
```

Notez que ce dernier fichier appelle le `df-preseed.cfg` décrit dans la section [“Pré-configuration de l'installateur”](#).

Configuration du menu de lancement UEFI

Le fond d'écran et le texte qui s'affichent au lancement du Live en version UEFI se configurent dans le dossier `config/includes.binary/boot/grub/`.

```
cd ~/ma_debian_perso/config/includes.binary/boot/grub
ls
grub.cfg live-theme splash.png
cd live-theme && ls
theme.txt
```

Le fond d'écran se nomme `splash.png` et doit faire 1024×768 px en PNG. Pour les fichiers texte, voici ceux de DFiso :

- `config/includes.binary/boot/grub/grub.cfg` : configuration des entrées du menu de GRUB

[grub.cfg](#)

```
set default=0

loadfont $prefix/dejavu-bold-16.pf2
loadfont $prefix/dejavu-bold-14.pf2
loadfont $prefix/unicode.pf2
set gfxmode=auto
insmod all_video
insmod gfxterm
insmod png

set theme=/boot/grub/live-theme/theme.txt
set menu_color_normal=cyan/blue
set menu_color_highlight=white/blue

terminal_output gfxterm

insmod play
play 960 440 1 0 4 440 1
```

```
# Live boot
menuentry "Tester Debian 11" {
    linux    /live/vmlinuz boot=live components quiet splash
    acpi_osi=Linux username=humain hostname=DFiso locales=fr_FR.UTF-8
    keyboard-layouts=fr timezone=Europe/Paris
    initrd   /live/initrd.img
}
menuentry "Tester Debian 11 (mode sans échec)" {
    linux    /live/vmlinuz boot=live components memtest noapic noapm
    nodma nomce nolapic nomodeset nosmp nosplash vga=normal username=humain
    hostname=DFiso locales=fr_FR.UTF-8 keyboard-layouts=fr
    timezone=Europe/Paris
    initrd   /live/initrd.img
}

menuentry "Installation classique" {
    linux    /install/gtk/vmlinuz video=vesa:ywrap,mtrr vga=788 quiet
    locale=fr_FR.UTF-8 keyboard-configuration/layoutcode=fr keyboard-
    configuration/variantcode=latin9 keyboard-layouts=fr
    file=/cdrom/install/df-preseed.cfg
    initrd   /install/gtk/initrd.gz
}

menuentry "Installation en mode expert" {
    linux    /install/vmlinuz priority=low vga=normal
    initrd   /install/initrd.gz
}
```

Notez que ce fichier appelle le `df-preseed.cfg` décrit dans la section [“Pré-configuration de l'installateur”](#).

- `config/includes.binary/boot/grub/live-theme/theme.txt` : configuration graphique du menu GRUB

[theme.txt](#)

```
desktop-image: "/boot/grub/splash.png"
title-color: "#ffffff"
title-font: "DejaVu Sans Bold 16"
title-text: "Debian 11 par Debian-Facile : testez ou installez ..."
message-font: "Unifont Regular 16"
terminal-font: "Unifont Regular 16"

#help bar at the bottom
+ label {
    top = 100%-50
    left = 0
    width = 100%
    height = 20
    text = "@KEYMAP_SHORT@"
    align = "center"
```

```
        color = "#ffffff"
        font = "DejaVu Sans Bold 14"
    }

#boot menu
+ boot_menu {
    left = 10%
    width = 80%
    top = 52%
    height = 48%-80
    item_color = "#a8a8a8"
    item_font = "DejaVu Sans Bold 14"
    selected_item_color= "#ffffff"
    selected_item_font = "DejaVu Sans Bold 16"
    item_height = 18
    item_padding = 0
    item_spacing = 4
}

#progress bar
+ progress_bar {
    id = "__timeout__"
    left = 15%
    top = 100%-80
    height = 16
    width = 70%
    font = "DejaVu Sans Regular 14"
    text_color = "#000000"
    fg_color = "#ffffff"
    bg_color = "#a8a8a8"
    border_color = "#ffffff"
    text = "@TIMEOUT_NOTIFICATION_LONG@"
}
```

Pré-configuration de l'installeur

Le fichier `preseed.cfg` va vous permettre de supplanter les étapes de l'installeur Debian et ainsi accélérer le processus ou définir vos préférences. Ce fichier de configuration se situe par défaut dans `config/includes.binary/install/preseed.cfg`.

Pour l'exemple, voici celui de DFiso :

[preseed.cfg](#)

```
# uniquement les questions importantes
d-i debconf/priority string critical

# Nom de la machin par défaut
d-i netcfg/hostname string debian
```

```
# Ne pas créer de compte root (l'utilisateur ordinaire utilisera sudo).
d-i passwd/root-login boolean false

# Le compte sera ajouté à certains groupes.
d-i passwd/user-default-groups string audio cdrom video sudo netdev
plugdev fuse users lp lpadmin scanner floppy dip bluetooth

# fixer le 'hostname'
d-i netcfg/get_hostname string debian
d-i netcfg/get_hostname seen false

# tout dans une seule partition (atomic, home, multi)
d-i partman-auto/choose_recipe select atomic

# installation automatique de GRUB s'il n'ya qu'un seul OS installé sur
la machine.
d-i grub-installer/only_debian boolean true

# Réseau désactivé
d-i netcfg/enable boolean false
d-i apt-setup/use_mirror boolean false
```

Vous trouverez aisément des exemples de fichiers preseeds sur la toile.

Et voilà, on a fait le tour des modifications simples de votre Live. Il ne vous reste plus qu'à retourner dans votre dossier de build, revérifier vos dernières modifications puis lancer la formule magique :

```
lb clean && lb build
```

et hop... vous avez votre dérivée Debian personnalisée prête à être distribuée. Quelques mots à ce sujet dans le chapitre qui suit...

Distribuer votre Live : ligne de conduite pour les dérivées Debian

Vous venez de construire une dérivée Debian. Ce n'est pas n'importe quoi, vous utilisez le travail de toute une communauté, alors une petite ligne sérieuse :

Debian est une distribution GNU/Linux, mais pas que... Debian, c'est aussi une ligne de conduite, une charte, et certaines recommandations quant à la distribution de dérivée. Cette ligne de conduite est précisée dans les Debian Derivatives Guidelines (<https://wiki.debian.org/Derivatives/Guidelines>). je vous conseille vivement de consulter ce document afin de respecter un maximum la distribution grâce à laquelle vous avez pu construire la votre. Merci beaucoup.

Un mot sur les copyright

Si vous voulez diffuser votre live, ce que vous y ajoutez est de votre responsabilité, donc respectez les licences, créditez les auteurs, placez des liens dans un README... bref soyez responsables de ce

que vous diffusez : les débutants ou les curieux de Debian méritent de recevoir un truc propre ;))

Conclusion

Log log log ... c'est le maître-mot de ce processus de construction. Pourquoi ? Parce que vous aurez de la chance si tout se passe exactement comme vous le souhaitez du premier coup ! Lisez les logs `ma_debian_perso/build.log`.

Liens

- [le manuel officiel de la team debian-live](#)
- [un exemple pour construire un live USB persistant](#)

¹⁾

N'hésitez pas à y faire part de vos remarques, succès, améliorations ou échecs !

From:

<http://debian-facile.org/> - **Documentation - Wiki**

Permanent link:

<http://debian-facile.org/doc:install:live-build>

Last update: **20/09/2024 10:33**

