Démystifier Archlinux ou Archlinux en moins de 20 étapes

Ou comment vous faire pousser un peu plus de barbe sur le visage

Federico GONZÁLEZ

Alternatives87 (http://alternatives87.org)

2019-03-06

Démystifier Archlinux, ou comment installer Archlinux en moins de 20 étapes.

... et vous faire pousser un peu plus de barbe.

Qu'est-ce qu'« Archlinux » ?

Archlinux est une distribution [1] différente de la distribution *debian GNU/Linux* que nous installons par défaut dans notre association **Alternatives87**.

- *Archlinux* est une distribution de réputation difficile, plutôt associée à des nerds ;
- C'est une distribution plutôt orientée poste de travail ;
- L'installation se fait à partir d'une iso live d'installation sans interface graphique ni même menus en curses ;
- Il faut suivre une procédure décrite dans les différents sites et forums consacrés à cette distribution ;

1. Une distribution est une version de l'environnement GNU/Linux avec une philosophie, des paquets, des environnements, etc... Il en existe des centaines différentes dont l'actualité peut être suivie ici : `https://distrowatch.com`

```
[?] Debian installer main menu

Choose the next step in the install process:

  Choose language
  Configure the keyboard
  Detect and mount CD-ROM
  Load installer components from CD
  Detect network hardware
  Configure the network
  Set up users and passwords
  Configure the clock
  Detect disks
  Partition disks
  Install the system
  Configure the package manager
  Install the GRUB boot loader on a hard disk
  Install the LILO boot loader on a hard disk
  Continue without boot loader
  Finish the installation
  Change debconf priority
  Check the CD-ROM(s) integrity
  Save debug logs
  Execute a shell
  Eject a CD from the drive
  Abort the installation
```

The GNU/Linux User Life Cycle

Quelles sont les différences avec GNU/Linux debian ?

- Installation très différente, plus manuelle, pas guidée ;
- Gestionnaire de paquets différents *Pacman* textuel ;
- Moins de paquets maintenus dans *Archlinux* pure que dans *debian* même sans contrib et non-free ;
- Pas d'interfaces graphique pour installer des paquets dans une *Archlinux* pure ;
- Pour installer des paquets il faut aller installer une interface graphique qui n'est pas disponible dans les paquets par défaut (donc la compiler) sauf à installer une dérivée d'Archlinux (A.U.R.) donc certains peuvent poser des soucis de sécurité (Dépôts orphelins) ;
- Certains paquets disponibles un jour sur les dépôts classiques peut se trouver du jour au lendemain viré vers les dépôts AUR (cas de « srm »).

Gestionnaire de paquets

Les différentes étapes de l'installation

- préparation : exécution de l'environnement d'installation (configurations ....)
- formatage des partitions cible ;
- installation d'une base ou plus dans la cible ;
- chroot ;
- configurations au sein du chroot ;
- installation du grub ;
- sorties et redémarrage.

Où obtenir l'ISO d'installation ?

L'ISO[2] d'Archlinux est disponible sur le site d'archlinux et dans plusieurs miroirs :

- ici : `https://archlinux.org/download`
  ou encore :
- ici (pour la dernière) :
  `https://mirror.thekinrar.fr/archlinux/iso/2019.02.01/`

Une fois l'image téléchargée, la gravure se fait assez facilement par la commande « dd » en mode superutilisateur sur un support amovible (CD, DVD, clé USB ...).

2. Une ISO est un fichier contenant une « image » du medium d'installation, pas une copie fichier par fichier mais une copie piste par piste. Pour les personnes ayant connu M$-DOS cela correspondait au résultat de la commande diskcopy. Sous GNU/Linux on utilise la commande « dd » en lignes de commandes pour un résultat similaire.

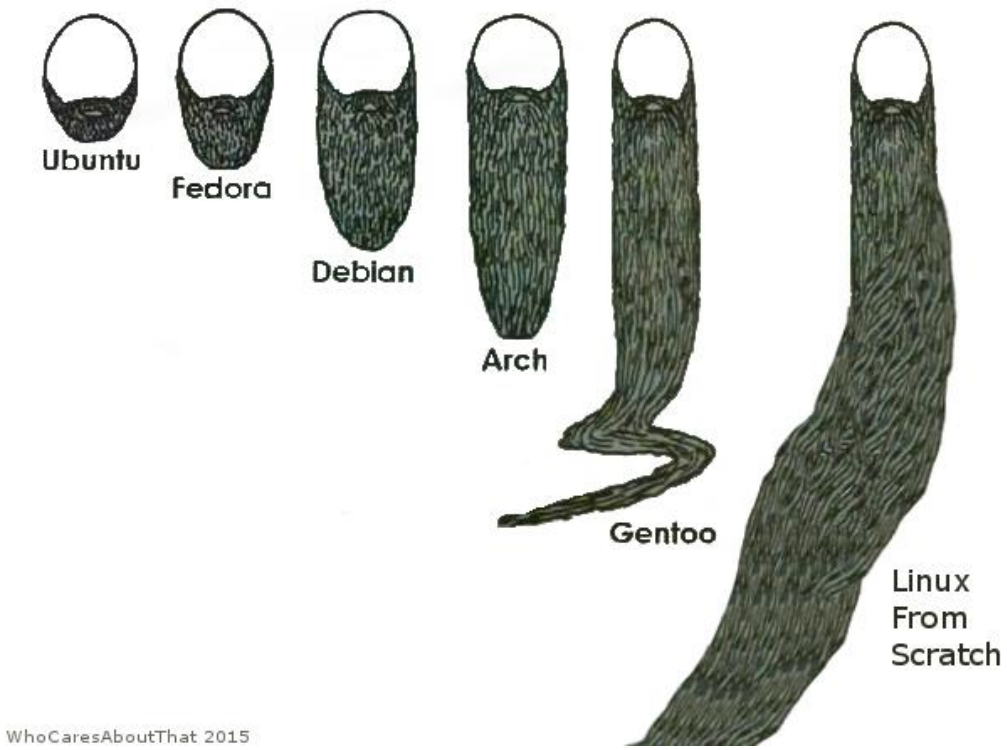
```
$~ sudo dd status=progress if=fichier_iso_d_installation.iso of=/dev/sXYZ
```

ou bien :

```
$~ su
#~ dd status=progress if=fichier_iso_d_installation.iso of=/dev/sXYZ
```

Une image de l'Archlinuxien

En parlant d'image, la suivante va montrer justement l'image d'un·e archlinuxien·ne à barbe :



Elle montre bien évidemment tout le côté barbesque des utilisateurs de distributions réputées très difficiles.

Étape 1 : La phase de préparation

Dans la phase de préparation il faut bien évidemment prendre le temps de réfléchir à comment sera configuré le futur système :

- Quelle façon de découper le disque ?
- Quelles informations concernant la machine ?
- Quels utilisateurs ?

Prêt·e ? Alors go, on démarre sur le medium d'installation !

Étape 2 : Configurations du système hôte

Le système hôte est le système sur lequel on vient de démarrer, i.e. la clé USB. Il nécessite qu'on y paramètre l'agencement du clavier, la connexion à internet et la date et l'heure. Toutes les opérations sont faites en administrateur ! Attention !

Il faut donc commencer par passer en français avec la commande `loadkeys fr` (attention : A $\rightarrow$ B).

Ensuite il faut se connecter au réseau, soit en wifi `wifi-menu` soit en câble éthernet `dhcpcd`.

Si câble éthernet branché au démarrage une ip est sans doute déjà attribuée, pour le vérifier, utilisez `ip addr`

```
ip addr

dhcpcd ou wifi-menu ou rien du tout si ok. // si pas d'adresse attribuée

ip addr // si pas d'adresse attribuée initialement
```

Étape 3 : Réglage de l'heure et de la date du

L'étape suivante consiste à synchroniser heure et date depuis le site web.

Cette étape est importante pour pouvoir ainsi ne pas avoir de problèmes avec les signatures de paquets.

```
timedatctl set-ntp 1

timedatectl status

hwclock --systohc --utc
```

Étape 4 : Préparation des partitions

Bien que l'usage soit souvent d'utiliser `fdisk` dans de nombreux tutoriels en ligne, je préconise tout de même l'utilisation du bon vieux `cfdisk` pour aider lorsqu'on est pas à l'aise avec l'installation en mode textuel pur.

Pour la suite de l'installation, le disque cible (installation) sera /dev/sda.

Le disque sera coupé en 3 partitions : `/dev/sda1` pour la racine `/`, ensuite `/dev/sda2` pour le *swap* et enfin `/dev/sda3` pour la partition `/home`.

```
cfdisk /dev/sda ...

ou bien :

fdisk /dev/sda
```

Étape 5 : Installation de la base du système

Une fois le système configuré vient l'installation du système de base, cela se fait simplement via une commande. Mais avant cela il va falloir accrocher le disque actuel d'installation `/dev/sda` et ses différentes partitions `/dev/sda1`, `/dev/sda2` et `/dev/sda3` dans un dossier du système hôte.

Les documentations habituelles parlent du dossier `/mnt` comme point d'ancrage habituel. La diapo suivante montre l'ancrage des différentes partitions, puis l'installation du système de base pour finir :

```
mount /dev/sda1 /mnt
mkdir /mnt/home
mount /dev/sda3 /mnt/home
swapon /dev/sda2

// et on vérifie par un lsblk :
NAME   MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda      8:0    0 931,5G  0 disk
\-sda1   8:1    0    29G  0 part /
\-sda2   8:2    0     5G  0 part [SWAP]
\-sda3   8:3    0 897,5G  0 part /home
sdb      8:16   1   3,8G  0 disk
\-sdb1   8:17   1   602M  0 part  ?????(à chercher)
\-sdb2   8:18   1    64M  0 part

pacstrap /mnt base
```

Une variante consiste à utiliser cette commande `pacstrap` pour installer d'autres paquets essentiels par la suite à savoir `dialog` pour les menus textuels et `wpa_supplicant` pour la gestion du wpa du wifi et tant qu'à faire aussi `grub` pour qu'il soit déjà présent sur le support cible.

```
pacstrap /mnt base dialog wpa_supplicant grub
```

Une fois cette base installée, vient la préparation d'un fichier important pour la suite : le fichier « `fstab` » de la future installation.

Le paramètre `-p` indique que sont exclus les pseudo systèmes et le paramètre `-U` indique que les UUID sont à utiliser plutôt que les `/dev/...`

```
genfstab -p -U /mnt > /mnt/etc/fstab
```

Étape 6 : chroot !

Bien bien bien !

La base est posée, les paquets essentiels pour une suite sont déposés aussi, le fichier de configurationd de la future table de partition est écrit au bon endroit, il est temps de quitter l'environnement hôte et de passer dans la cible.

Cela se fait par la commande `arch-chroot`

Elle a l'avantage d'effectuer tous les montages de partitions dans /dev et /proc facilement. Le paramètre « bash » final est purement optionnel.

```
arch-chroot /mnt bash
```

Étape 7 et suite des opérations, la configuration du système cible.

La configuration du système cible consiste à configurer une partie des paramètres nécessaires, vous remarquerez que ce sont les mêmes opérations que lorsqu'on installe un système debian.

- le keymap : au redémarrage on souhaite avoir le clavier en français (ou en Bépo !)
- les locales : autant que les commandes soient en français par la suite aussi
- le hostname : il faut que l'ordinateur soit nommé et différencié sur le réseau local.
- le fichier hosts : nécessaire pour gnome : associer le hostname et 127.0.1.1
- le locale.gen
- le localtime
- locale-gen

La première étape étant de configurer la langue pour le futur clavier et ensuite les locales.

```
echo KEYMAP=fr > /etc/vconsole.conf

echo LANG=fr_FR.UTF-8 > /etc/locale.conf
echo LC_ALL=C >> /etc/locale.conf //cette ligne est optionnelle
echo ... >> /etc/locale.conf
echo fr_FR.UTF-8 UTF-8 >> /etc/locale.gen
locale-gen
```

Etape 8 : Réglage du nom d'hôte

```
echo NouveauNomHôte > /etc/hostname
echo 127.0.1.1      NouveauNomHôte.localdomain    NouveauNomHôte >> /etc/hosts
```

Étape 9 : Réglage du décalage temporel / fuseau horaire

Pour chaque ordinateur fonctionnant avec UNIX ou un clone (et Archlinux en est) il est conseillé de laisser l'horloge matérielle (hw) sur l'heure GMT et d'indiquer dans quel fuseau horaire se trouve physiquement la machine. C'est le rôle du fichier `/etc/localtime` qui est un raccourcis du fichier `/usr/share/zoneinfo/[Continent]/[Ville]`

```
rm /etc/localtime
ln -s /usr/share/zoneinfo/Europe/Paris > /etc/localtime
```

Étape 10 : Les utilisateurs

Il y a d'office un utilisateur créé, c'est `root`. Pour lui ne manque que le mot de passe à générer.

Ensuite si le futur système a besoin d'un utilisateur au quotidien, il va falloir le créer, puis, lui attribuer un mot de passe différent, et l'ajouter à certains groupes.

Bien que la commande `useradd` puisse ajouter directement les groupes additionnels à l'utilisateur via l'option `-G` j'ai choisi de décomposer cela en 2 commandes pour faire découvrir `usermod` la commande de modification.

```
\item passwd root

\item useradd -m utilisateur

\item usermod -aG audio,video,optical,wheel utilisateur

\item passwd utilisateur
```

Étape 11 : Recompilation du noyau avec les bons crochets.

Le noyau a été compilé par défaut, cette 11e étape peut s'avérer souvent superflue, mais, on peut cependant l'exécuter afin de bien recompiler le noyau pour la machine où est placée cette installation.

Les « crochets » — hooks en anglais — sont des modules installés dans l'image qui se lance au démarrage.

```
mkinitcpio -g linux -c /etc/mkinitcpio.conf

// notez qu'un simple :

mkinitcpio -g linux

// suffit amplement pour une installation simple.
```

Un petit rappel simplifié de comment démarre une installation avec (GNU/)Linux ? Allez !

Donc pour mémoire au démarrage grub va chercher un fichier spécifique qui va appeler un fichier contenant une « image » d'un linux exécutable, très minimaliste, mais qui va jouer le rôle de démarrage du système et de vérification des partitions en place, etc...

Une fois que tout semble ok et les services vérifiés, alors cette image cède la place à l'installation proprement dite.

Étape 12 : Installation de GRUB

GRUB — ou GRUB 2) est le chargeur de démarrage qui a en charge de gérer le démarrage du système, il propose par défaut et sauf à modifier le fichier `/boot/grub/grub.cfg` de démarrer par défaut et une ligne vers des paramètres avancés.

Dans les paramètres avancés, on trouve le démarrage normal et le démarrage de secours (très utile surtout si on a une installation sur un disque externe qui se balade d'une machine à une autre) car ce mode recheck tout le matériel présent au démarrage.

La dernière phase consiste à installer les fichiers nécessaires sur le disque interne de la future installation, i.e. `/dev/sda`.

```
mkdir /boot/grub // il est possible que ce dossier n'existe pas.

grub-mkconfig > /boot/grub/grub.cfg

// ou la variante :

grub-mkconfig -o /boot/grub/grub.cfg

grub-install /dev/sda
```

Étape 13 : Fin d'installation

La sortie de l'environnement chrooté se fait d'abord par la commande `exit` puis une fois sorti(e) il suffit de faire redémarrer la machine par `reboot`.

Certains tutoriels indiquent de démonter les partitions du disque cible avant de redémarrer, ce qui peut être utile mais une installation simple comme celle-ci ne le nécessite pas.

Une fois le début du redémarrage il faut retirer le medium d'installation.

Étape 14 : Configurations post-installation d'une interface graphique

Première étape : se reconnecter au réseau wifi via `wifi-menu` ou ethernet via `dhcpcd`[3]

L'installation actuelle n'est que textuelle, cela est peu utile pour un utilisateur lambda, aussi on va y installer un minimum de paquets pour avoir au moins une interface graphique, je choisis *LXDE* pour sa simplicité d'utilisation. Pour gérer les connexion internet j'utiliserai `wicd` et sa version gtk : `wicd-gtk`.

Bien sûr pour que *LXDE* fonctionne il faut que le serveur *X* soit opérationnel et (optionnel mais pratique) un *display manager* simpliste tel que *lightdm*.

Mais avant il faudra commencer par voir la carte graphique utilisée et rechercher le pilote nécessaire à son fonctionnement car le paquet *xorg* n'installe pas tous les drivers, juste 2 ou 3 basiques. Ceci se fera par la commande `lspci` et on cherchera la ligne contenant les lettre `VGA`.

3. Notez que sous *debian* la commande est `dhclient`.

```
wifi-menu // ou // dhcpcd // ou rien du tout !

lspci | grep VGA

pacman -S xorg lxde lightdm lightdm-gtk-greeter xf86-video-[marque
```

Étape 15 : configuration du clavier graphique et test !

Le clavier graphique n'est pas encore configuré, cependant la commande `localectl` de *systemd* va permettre de configurer cela, ensuite ...

... on teste ! on va demander à `systemctl` de démarrer lightdm (ou plutôt lightdm.service), si tout se passe bien on pourra ainsi lui demander de le démarrer systématiquement.

Ensuite on va aussi démarrer `wicd` comme gestionnaire de connexions réseau.

```
localectl status

localectl set-x11-keymap fr pc105 latin9

systemctl start lightdm.service

// s'il y a un bug : le clavier est en anglais !
```

Étape 16 : Démarrages automatiques

Cette fois-ci c'est le moment d'ajouter ce qu'il faut au démarrage automatique du système. Cela sera fait par la commande

```
systemctl enable nom.du.service
```

Et c'est F I N I !

```
systemctl enable ligthdm.serivce

systemctl enable wicd
```

Licence

...osez l'alternative !

Cette présentation est placée sous licence libre GNU FDL

Elle a été réalisée en utilisant uniquement des logiciels libres.

Permission vous est donnée de copier, distribuer et/ou modifier ce document selon les termes de la Licence GNU Free Documentation License, Version 1.3 ou ultérieure publiée par la Free Software Foundation ; sans section inaltérable, sans texte de première page de couverture et sans texte de dernière page de couverture. Une copie de cette Licence est incluse dans la section appelée « GNU Free Documentation License » du code source de ce document.

À propos de l'association

Alternatives87

est une association à but non lucratif (loi 1901).

Notre objectif est la promotion des logiciels libres sur Limoges et sa région.

N'hésitez pas à nous rejoindre !

`http://alternatives87.org`