

Installer une basique Archlinux en 20 étapes (et qu'elle redémarre !)

Alternatives 87 - F. G. - Je ne sais pas quand

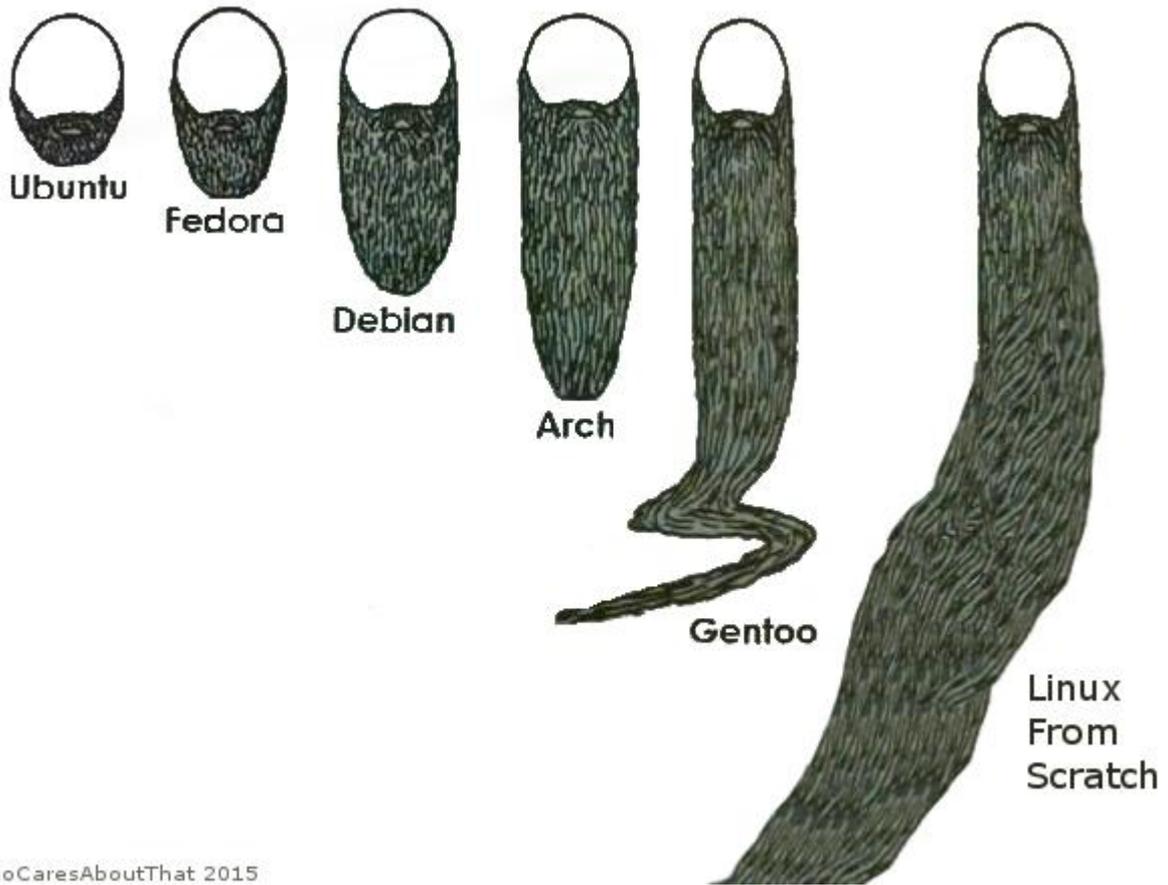
Motivations

Depuis le temps que j'en installe des distributions GNU/Linux on suit toujours (ou presque) le même ordre d'installation, les questions suivent toujours un même ordre au point que je sais quasiment d'avance la suite de l'installation.

Avec le temps vient l'envie d'en savoir plus, je ne suis pas scientifique de formation pour rien, on aime comprendre et modéliser, certain·e·s appellent cela le syndrome du rubik's cube, d'autres une sale manie de fouineur, mais toujours est-il qu'en savoir plus, s'autonomiser c'est progresser vers une plus grande liberté.

Du coup, avec ce même temps et l'envie d'en savoir plus je me tourne peu à peu vers des distributions linux où il m'est laissé plus de choses à faire par moi même, où, je peux par la même, installer plus finement un système nettoyé de tout le superflu où ne s'installe que ce dont j'ai envie ou besoin.

La démarche linuxienne peut se trouver dans bon nombre d'images sur internet, elle décrit l'évolution naturelle d'un utilisateur de GNU/Linux :



WhoCaresAboutThat 2015

on a aussi droit à d'autres représentations rigolotes :



The GNU/Linux User Life Cycle

Certes cette dernière image n'est pas très flatteuse pour les utilisateurs d'Arch (mais il y a pire, voire plus loin) mais elle donne une idée de l'évolution naturelle d'un utilisateur ou d'une utilisatrice de GNU/Linux : au début utilisant une distribution facile d'accès qui prend plein de décisions implicites laissant à la personne le plaisir d'utiliser le système sans trop d'efforts de compréhension des rouages internes, puis peu à peu c'est vers des versions plus difficiles qu'on arrive (en l'occurrence j'en suis aux stades 4 et 5 pour les choix de distributions).

Équipement préalable

L'équipement à avoir pour commencer est assez simple, cette installation va s'effectuer :

- sur une machine sans EFI actif donc en mode BIOS ou Legacy BIOS
- sans chiffrement d'aucune sorte
- 3 partitions basiques : /, /home et *swap*
- avec une clé USB comme medium de démarrage

La préparation commence par un petit tour sur le site <https://archlinux.org> et plus précisément la section *Download* une fois le choix du miroir effectué dans la liste proposée je finis par télécharger le fichier iso suivant : https://mirror.thekinrar.fr/archlinux/iso/2018.10.01/archlinux-2018.10.01-x86_64.iso avec par exemple la commande *curl* ou la commande *wget*.

```
curl -L -C- -O https://mirror.thekinrar.fr/archlinux/iso/2018.10.01/archlinux-2018.10.01-x86_64.iso
wget -c https://mirror.thekinrar.fr/archlinux/iso/2018.10.01/archlinux-2018.10.01-x86_64.iso
```

notez que :

- pour `curl` :
 - -L permet de suivre les redirections éventuelles si ce lien n'est pas direct
 - -O : indique d'utiliser le nom distant comme nom local (-o permet de donner un nom local différent du nom distant)
 - -C- : permet à curl de reprendre le téléchargement à l'octet près après une coupure de connexion
- pour `wget` l'option -c indique de reprendre le téléchargement à l'octet près en cas de coupure de connexion.

Une fois le fichier téléchargé on peut (par précaution ?) télécharger le fichier de signature ou le fichier md5sum ou sha1sum (je vais prendre le md5sum) ici <https://mirror.thekinrar.fr/archlinux/iso/2018.10.01/md5sums.txt> et vérifier que l'ISO est correcte :

```
[anonymous@Tungsten ISOs]$ md5sum -c md5sums.txt
archlinux-2018.10.01-x86_64.iso: Réussi
md5sum: archlinux-bootstrap-2018.10.01-x86_64.tar.gz: Aucun fichier ou dossier de ce type
archlinux-bootstrap-2018.10.01-x86_64.tar.gz: Échec d'ouverture ou de lecture
md5sum: Attention : 1 fichier de la liste n'a pas pu être lu
```

la 2e ligne du bloc de texte finissant par Réussi concernant le fichier "iso" téléchargé indique que ce fichier est exactement el même que celui téléchargé à l'octet près (non je ne parlerai pas de la collision md5...)

maintenant on passe à la préparation de la clé usb, je vais supposer (raisonnablement) que le disque interne de mon ordinateur est désigné par `/dev/sda` et qu'en branchant la clé USB un nouveau périphérique a été détecté en tant que `/dev/sdb`, je n'ai plus qu'à passer en administrateur (root) pour effectuer l'opération de gravure du support (je suppose bien sûr que je suis resté dans le dossier où a été téléchargée l'image `archlinux-2018.10.01-x86_64.iso`) :

```
su
Mot de passe :
dd status=progress bs=512 if=archlinux-2018.10.01-x86_64.iso of=/dev/sdb
```

A noter :

- `status=progress` permet d'afficher une progression au lieu de ... rien.
- `bs=512` dit à la commande `dd` de lire 512 octets et de les écrire en une fois (donc écriture pas blocs de 512 octets)

Avant de commencer on vérifie qu'on dispose d'une information cruciale pour bien démarrer : quelle est la touche du clavier de `[ESC]` ou `[Suppr]` ou de `[F1]` à `[F12]` qui permet de choisir le medium de démarrage par la suite. Ici c'est `[F12]` dans mon cas. On éteint l'ordinateur qui va migrer s'il ne l'était pas déjà.

On est prêts pour l'installation

Installation

Étape 1 Démarrage !

Branchement de la clé sur l'un des ports USB.

Démarrer et appuyer aussitôt sur la touche `[F12]`

Choisir d'installer (démarrer archlinux)

Étape 2 Passage en azerty du clavier

```
loadkeys fr
```

Cette commande permet de changer la disposition du clavier qui par défaut est "us" et va passer en "fr". Comme pour l'heure le clavier est us il faut taper l o q d k e y s (espace) f r

Étape 3 Régler la date et l'heure du système

bla bla bla

```
timedatctl set-ntp 1
timedatectl status
hwclock --systohc --utc
```

bla bla bla --utc -> -u sinon --localtime -> -l

Étape 4 Se connecter au grand vilain internet

Cette image est comparable à une image *netinstall* de chez Debian / Devuan : elle nécessite une connexion à internet pour télécharger les paquets etc..

Étape 5 Partitionnement du disque d'installation et formatage des partitions nouvellement formées

```
fdisk /dev/sda
```

1ère partition : bootable (a) de type (t) linux (83) primaire, au début de taille 40 G

2e partition : type (t) linux swap (82) primaire, taille 6 G

3e partition : type (t) linux (83) primaire taille : tout le reste

```
w
```

```
sync
```

```
mkfs.ext4 /dev/sda1
```

```
mkswap /dev/sda2
```

```
mkfs.ext4 /dev/sda3
```

Les partitions sont prêtes à être montées dans le dossier.

Étape 6 Mountage du futur système dans un dossier temporaire

```
swapon /dev/sda2
```

```
mount /dev/sda1 /mnt
```

```
mkdir /mnt/home
```

```
mount /dev/sda3 /mnt/home
```

Étape 7 Installation du système de base pour le futur ordinateur

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

Étape 7-bis (variante) Installation complète du futur système

Je vais supposer qu'une installation complète implique l'installation de X (partie graphique) et d'un environnement de bureau, pour cela archlinux offre par défaut :

- gnome
- kde (paquet : plasma-desktop)
- deepin
- mate
- xfce4
- lxde
- lxqt
- openbox
- ...

Mais avant de commencer on va taper une ligne pour faire connaissance avec les composants (la carte graphique principalement) de l'ordinateur et rechercher quelle carte graphique est à l'intérieur :

```
[anonymous@Tungsten ISOs]$ lspci | grep VGA
00:02.0 VGA compatible controller: Intel Corporation Mobile GM965/GL960 Integrated
Graphics Controller (primary) (rev 03)
```

explications :

- `lspci` pour liste les périphériques pci affiche une liste de nombreux périphériques internes de la machine (sauf ceux en USB)
- le pipe `|` permet d'envoyer la sortie de la commande à gauche de lui dans une commande à droite
- `grep` est une commande qui reçoit du texte et n'affiche que ce qu'on lui passe en option, ici je veux qu'elle affiche la ligne où figurera l'indication VGA

L'analyse rapide de cette sortie me dit que j'ai une carte graphique (VGA) de marque Intel, ensuite une recherche du paquet graphique pour cette carte intel va me donner le nom du paquet qu'il faudra installer :

```
[anonymous@Tungsten ISOs]$ pacman -Ss xf86 | grep video | grep intel
extra/xf86-video-intel 1:2.99.917+853+g0932a6b3-1 (xorg-drivers)
```

explications :

- dans archlinux tous les paquets de pilotes pour le serveur X commencent par xf86 (les anciens auront reconnu xfree86)
- `grep video` filtre les sorties pour ne garder que les pilotes vidéo
- `grep intel` car dans le nom de la sortie de la commande `lspci | grep VGA` le mot *Intel* est apparu. Bien sûr il faut adapter cela à sa propre configuration ! Maintenant que c'est ok je décide d'installer

```
pacstrap /mnt base base-devel wpa_supplicant dialog xorg xf86-video-intel grub lxde lxdm
```

Il n'y a plus qu'à aller boire un café, une tisane ou faire sa lessive le temps que ça s'installe.

Étape 8 Générer le fichier `fstab` de la future installation.

la future installation se trouve toute entière montée dans le dossier `/tmp` on va donc créer le fichier `/etc/fstab` au bon endroit :

```
genfstab -pU /mnt > /mnt/etc/fstab
```

explication :

- `-pU` équivaut à `-p -U` : `-p` veut dire qu'on retire du résultat tous les pseudo-systèmes de fichiers (tmpfs et loop) et le `-U` (en majuscule) indique d'utiliser les UUID au lieu des notations `/dev/sdxxx`
 - L'avantage des UUID c'est que si par mégarde un périphérique est reconnu une fois de temps en temps autrement que par l'ordre habituel, grub s'en fichera car il identifiera les périphériques par leur IDentifiant Unique Unifié (?)
- `/mnt` : c'est là qu'est la racine de notre futur système
- `>` signifie que la sortie de ce résultat sera directement écrite dans le fichier désigné après ce symbole, et si le fichier n'est pas vide il sera écrasé par ce qui va lui être envoyé.
- `/mnt/etc/fstab` car le fichier `/etc/fstab` du futur environnement est dans l'environnement accroché dans `/mnt`

Étape 9 Chroot vers le nouvel environnement.

L'opération du chroot (cruciale aussi lors des installations) est assez ... merdique. Il faut monter plusieurs points du systèmes, certains d'une certaine façon d'autre autrement, c'est la partie que je n'aime pas mais... sur arch ils ont créé un script ou un programme qui fait quasiment cela tout seul `arch-chroot` :

Variante 1 : en utilisant `arch-chroot`

Variante 2 : à l'ancienne via `chroot`

Étape 10 Définition de la nouvelle zone horaire du futur système et de l'heure

In `-sf /usr/share/zoneinfo/Europe/Paris /etc/localtime`

note : vous pouvez aussi le faire par `timedatectl set-timezone Europe/Paris`

note 2 : à l'étape 2 on peut aussi régler la date et l'heure du système avant de commencer les réjouissances par `timedatectl set-timezone Europe/Paris` puis `timedatectl set-ntp 1` puis `hwclock --systohc --utc`

Étape 11 Définition du clavier en mode textuel de la future installation

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

ce qui est suffisant mais on peut être plus précis par :

```
echo KEYMAP=fr-latin9 > /etc/vconsole.conf
```

le fichier `/etc/vconsole.conf` est typique d'une installation avec `systemd` ...

on peut ajouter la police du mode terminal par :

```
echo FONT=lat9w-16 >> /etc/vconsole.conf
```

(vous aurez noté le `>>` indiquant que cette ligne vient s'ajouter à la fin du fichier)

Étape 12 Définition du hostname et du fichier hosts

Ces deux fichiers sont utiles (surtout le premier) dans un réseau local ou pour faire du filtrage. Imaginons que la future machine va s'appeler `MonHote`, voici donc la commande :

```
echo MonHote > /etc/hostname
```

et ensuite on va ajouter ce nom aussi dans le fichier d'hôtes soit en éditant le fichier (j'aime bien l'éditeur `nano`) :

```
nano -cil /etc/hosts
GNU nano 3.1 /etc/hosts

1 # Static table lookup for hostnames.
2 # See hosts(5) for details.
3 127.0.0.1 MonHote
4 127.0.0.1 localhost.localdomain localhost

^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^J Justifier ^C Pos. cur.
^X Quitter   ^R Lire fich.^\ Remplacer ^U Coller    ^T Orthograp.^_ Aller lig.
```

comme cela se voit en bas, il faut taper **CTRL + O** suivi de **CTRL + X** pour enregistrer le nom puis quitter, mais on peut aussi envoyer directement le nom via echo comme précédemment :

```
echo 127.0.1.1      MonHote >> /etc/hosts
echo 127.0.0.1      localhost.localdomain      localhost >> /etc/hosts
```

vous aurez remarqué qu'aucune adresse IP n'a été saisie en IPV6 car pour l'heure je ne l'utilise pas ou le désactive.

Étape 13 Le futur réseau

Actuellement la machine pourra se connecter en réseau local (et internet) mais ne pourra pas faire grand chose de plus sauf si on lui ajoute entre 1 et 3 adresses IP (v4 ou v6) pour résoudre les noms de domaines dans le fichier `/etc/resolv.conf`. Pour se connecter automatiquement au réseau local de façon filaire dès le redémarrage :

```
systemctl enable dhcpcd
```

notez que ce n'est pas obligatoire et on peut très bien lancer manuellement le programme en cas de besoin et dans debian/devuan l'équivalent s'appelle `dhclient`. Pour une connexion wifi il y a le programme `wifi-menu` qui permet de configurer le wifi de `netctl` (le module de `systemd` qui gère les connexions wifi). `wifi-menu` permet de conserver les réglages wifi et ensuite il suffit d'activer ces services au démarrage pour que le réseau wifi soit exécuté dès le démarrage

```
wifi-menu
netctl enable nom-du-profil-wifi-choisi-dans-wifi-menu
systemctl enable netctl
```

Ces réglages ne sont pas nécessaire si on installe un gestionnaire de réseau graphique qui paramétera ce paramètre du style `wicd` ou `network-manager` (kde ou gnome). Imaginons que j'installe `wicd` (version gtk pour l'interface graphique)

```
pacman -S wicd-gtk dbus
systemctl enable dbus
systemctl enable wicd
```

si c'est le network-manager de gnome :

```
pacman -S network-manager network-manager-applet
systemctl enable NetworkManager
```

etc...

Étape 14 Les Locales linguistiques

Il y a 2 fichiers et 1 commande à exécuter. Les fichiers /etc/locale.conf et /etc/locale.gen. Pour le premier :

```
echo LANGUAGE=fr_FR > /etc/locale.conf
echo LANG=fr_FR.UTF-8 >> /etc/locale.conf
echo LC_ALL=C >> /etc/locale.conf
```

la 1ère et 3e ligne ne sont pas cruciales.

Ensuite soit on édite le fichier /etc/locale-gen et on décommente la ligne #fr_FR.UTF-8 UTF-8 soit on ajoute à la fin cette ligne-là par la commande echo, et ensuite on exécute le programme de génération des locales locale-gen :

```
echo fr_FR.UTF-8 UTF-8 >> /etc/locale.gen
locale-gen
```

Étape 15 Création des utilisateurs root et utilisateur

bla bla bla

```
passwd root
useradd -m -U -G wheel,users utilisateur
passwd utilisateur
```

bla bla bla

Étape 16

bla bla bla

```
commande, commande
bla bla bla
```

bla bla bla

Étape 17

bla bla bla

```
commande, commande  
bla bla bla
```

bla bla bla

Étape 18 Régénération de l'image au démarrage (initramfs)

Sur une installation simple non modifiée (pas de lvm, etc...) cette section est quasiment inutile car lors de la commande pacstrap précédente (étape 7) le metapaquet "base" installe mkinitcpio et il génère déjà la future image de linux, mais, si pour raisons x ou y on souhaite modifier le fichier de configuration du mkinitcpio (qui est `/etc/mkinitcpio.conf`) pour ajouter des "hooks" (crochet) c'est à dire des possibilités dès le démarrage, alors il vaut mieux exécuter cette commande.

```
mkinitcpio -p linux
```

Si aucun message d'erreur (ou éventuellement des alertes (warning) apparaissent pour un manque de pilotes) c'est bon, on peut passer à la suite.

Étape 19 Configuration de grub

C'est (historiquement) l'étape qui a failli me faire abandonner l'utilisation d'arch au début (il y a 2 ans), car en fait j'oubliais systématiquement la première des commandes, il faut donc dans un premier temps générer le fichier de configuration de grub (ici en mono-système car je suppose qu'il n'y a pas d'autre système d'exploitation sur le disque) et ensuite il faut installer la partie de grub nécessaire dans le MBR (car oui nous ne sommes pas dans de l'uEFI).

```
grub-mkconfig -o /boot/grub/grub.cfg  
grub-install /dev/sda
```

Cette étape est cruciale, si le message "No error reported" apparaît, c'est gagné !

Étape 20 Sortie et redémarrage

Maintenant que tout est prêt on va donc sortir de l'environnement chrooté, puis redémarrer le système ce qui aura aussi pour effet de démonter toutes les partitions.

```
exit  
reboot
```

La première commande `exit` permet de sortir de l'environnement chrooté quant à la seconde, le reboot, force toute les fermetures des différents services, des partitions montées etc.. et fait redémarrer la machine, il suffit alors au moment où la machine est éteinte (ou juste au début de son redémarrage alors que l'écran semble encore éteint) de retirer le medium d'installation.

L'arch démarre et est opérationnelle pour la suite !

Au final ...

La machine est prête et opérationnelle, elle est certes non protégée dans son contenu mais ce tutoriel permet au moins de le faire facilement.