

<http://miamondo.org>

La planète Archlinux



Table des matières

Chapitre 1: premiers pas avec Archlinux

- 1.1. Spécificité d'Archlinux
- 1.2. Installer l'image *.iso sur un support de stockage amovible
 - 1.2.1. Se procurer l'image *.iso
 - 1.2.2. Installer l'image iso sur une clé USB
- 1.3. Booter, c'est-à-dire démarrer sur le support d'installation
- 1.4. Bienvenue sur les terres de Messire Arcolinux de Cartemère
 - 1.4.1. Configurer le clavier avec la commande loadkeys fr
 - 1.4.2. Tester sa connexion au réseau et vérifier l'heure système
 - 1.4.3. timedatectl

Chapitre 2: partitionner son disque dur

- 2.1. Déterminer le nom de son disque grâce à fdisk
- 2.2. Choisir le type de la table de partitionnement
- 2.3. Partitionner son disque dur en BIOS
 - 2.3.1. Première partition
 - 2.3.2. Seconde partition
 - 2.3.3. Troisième partition
- 2.4. Écriture de la table des partitions
- 2.5. Partitionner son disque dur en UEFI

Chapitre 3: Le formatage des partitions

- 3.1. Les différents systèmes de fichiers
- 3.2. Les commandes de formatage mkfs et mkswap
- 3.3. variante UEFI
- 3.4. Le montage des partitions en mode BIOS
- 3.5. Le montage des partitions en mode UEFI

Chapitre 4: Installer la base du système

- 4.1. Démonter et remonter les partitions
- 4.2. Sélectionner le miroir
- 4.3. Installer la base
- 4.4. Configurer le nouveau système
 - 4.4.1. Générer le fichier /mnt/etc/fstab
 - 4.4.2. Chrooter dans le nouveau système
 - 4.4.3. Renseigner le nom de l'ordinateur
 - 4.4.4. Créer un lien symbolique pour le fuseau horaire
 - 4.4.5. Sélectionner la langue du système
 - 4.4.6. Configuration de la locale pour tout le système
 - 4.4.7. Disposition du clavier
 - 4.4.8. Générer l'image du noyau Linux avec mkinitcpio
 - 4.4.9. Définir un mot de passe pour l'utilisateur root
- 4.5. Installer le chargeur d'amorçage
 - 4.5.1. En mode BIOS
 - 4.5.2. En mode UEFI
- 4.6. Redémarrage

Chapitre 5: post-installation

- 5.1. Lancement du script de post-installation
- 5.2. Analyse des différents scripts de post-installation
 - 5.2.1. Le script principal : archlinux_post-install.sh
 - 5.2.2. La liste des applications à installer: pkglist.txt
 - 5.2.3. le script tint2rc
 - 5.2.4. menu.desktop
 - 5.2.5. menu.xml
 - 5.2.6. rc.xml
 - 5.2.7. Le fichier post-install yay.sh
- 5.3. Les gestionnaires de paquets pacman et yay
 - 5.3.1. pacman
 - 5.3.2. yay
- 5.4. Synchronisation et mise à jour automatisée

Chapitre 1: premiers pas avec Archlinux

Archlinux est une distribution qui adhère au principe KISS. Si tu es complètement débutant(e) dans le monde Linux, il est préférable que tu te tournes vers une distribution grand public telle qu'Ubuntu qui va prendre en charge toute l'installation pour te proposer un système clé en main et fonctionnel.

En revanche, si tu as déjà acquis une certaine expérience du monde Linux et que tu es à l'aise avec la console ainsi qu'avec les commandes basiques de Bash, je te conseille de te lancer dans l'installation d'Archlinux. Tu en apprendras beaucoup sur l'architecture des distributions libres et tu auras le bonheur de construire un système d'exploitation personnalisé à ta guise.

Contrairement à Ubuntu qui est une distribution à version renouvelée tous les six mois, Archlinux est une distribution à publication roulante (rolling release). Elle est mise à jour très régulièrement. Les documents de référence se trouvent à cette [adresse](#). Tout au long de ce cheminement, je t'inviterai à les consulter. Tu es prêt(e) pour une nouvelle aventure? Oui? Alors, c'est parti!

1.1. Spécificité d'Archlinux

Archlinux a ceci de différent qu'elle installe le minimum. Ensuite, il faut se prendre en main. Contrairement à Ubuntu, il ne suffit pas de booter sur le support d'installation et de répondre gentiment aux questions. Avec Archlinux, il faut :

- partitionner le disque dur qui va accueillir le nouveau système d'exploitation,
- monter les partitions,
- configurer le fuseau horaire, la langue (qu'on appelle **locale**), le clavier etc.
- chrooter,
- installer un environnement de bureau ou au moins, un gestionnaire de fenêtres tel qu'**Openbox**.
- installer tous les logiciels de votre choix.

Tout ceci, bien sûr en ligne de commande et sans l'aide d'une interface graphique!

N'hésite pas à consulter le site francophone consacré à [Archlinux](#). Tu y trouveras tout un tas d'informations qui te permettront d'atteindre ton but.

1.2. Installer l'image *.iso sur un support de stockage amovible

1.2.1. Se procurer l'image iso

Pour installer Archlinux, il faut d'abord se procurer l'image *.iso la plus récente. Rendez-vous au bas de cette page qui répertorie tous les miroirs classés par pays. Moi, par exemple, j'habite non loin de Hanovre. C'est donc tout logiquement que j'ai cliqué sur le lien du miroir de l'Université de Hanovre:

<http://ftp.uni-hannover.de/archlinux/iso/AAAA.MM.JJ>

Ensuite, j'ai sélectionné l'image *.iso et je l'ai téléchargée:

http://ftp.uni-hannover.de/archlinux/iso/AAAA.MM.JJ/archlinux-AAAA.MM.JJ-x86_64.iso

Bien évidemment, dans les deux liens que je viens de te donner, il faut remplacer les portions **AAAA/MM/JJ** par la version la plus récente proposée par le miroir.

Si tu tiens absolument à installer la version 32-bits d'Archlinux, c'est ici que ça se passe. En ce qui me concerne, je ferai uniquement référence au site basé sur la version 64-bits.

1.2.2. Installer l'image iso sur une clé USB

L'image pèse 615 Mo. C'est peu et cela permet de l'installer sur n'importe quelle clé USB. La commande ci-dessous va te permettre de déterminer quel disque correspond à la clé USB, en comparant la capacité de stockage de cette dernière avec les informations qui s'affichent dans la console.

```
user@hostname ~$ sudo fdisk -l
```

Chez moi, la clé USB est `/dev/sdb`. Ensuite, tu ouvres un terminal et tu entres cette commande (sur une seule ligne!) en utilisant le chemin absolu, c'est-à-dire celui qui part de la racine de ton système et qui remonte jusqu'à l'image `*.iso` à flasher sur la clé USB.

```
user@hostname ~$ sudo dd bs=4096 if=/chemin/vers/iso of=/dev/sdb  
status=progress&&sync
```

1.3. Booter, c'est-à-dire démarrer sur le support d'installation

Avant toute chose, assure-toi que ton ordinateur est branché sur le secteur et qu'il est connecté **en filaire** au réseau.

Te voici donc en possession d'un support d'installation qui est une clé USB contenant l'image *.iso d'Archlinux. Il va maintenant falloir modifier l'ordre d'amorçage pour que ton ordinateur boote non pas sur ton disque dur actuel mais sur ta clé USB Archlinux.

Pour ce faire, il suffit de redémarrer ton ordinateur et de presser la touche d'accès au boot menu. Ça varie d'un ordinateur à l'autre. Parfois, c'est la touche **F12**. Parfois, c'est **F9**. Cela peut être aussi **Echap**. Au final, il faut que ta clé USB soit tout en haut de la pile d'amorçage, de telle sorte que le **BIOS**, c'est-à dire le micrologiciel de démarrage, l'allume en premier. Si tout se déroule comme prévu, cette magnifique page d'accueil va s'afficher.



Sélectionne la première ligne et valide en cliquant sur **Entrée**. Tu vas basculer sur un écran noir, avec en haut à gauche une invite de commande:

```
archiso login: root (automatic login)
root@archiso ~ #
```

1.4. Bienvenue sur les Terres de Messire Arcolinus de Cartemère dit "Root le dingue"

Sans savoir si tu es un ange ou un démon, cette invite de commande te confère d'emblée le statut de **root**, c'est-à-dire qu'elle t'octroie les pleins pouvoirs. Rien que ça! Tu n'as même pas besoin d'entrer un mot de passe (automatic login). C'est open bar! En fait, tout comme Dieu a créé le monde, tu vas toi aussi créer le tien.

Lors de l'installation, ce statut de dictateur est nécessaire mais par la suite, lorsque le système sera installé, il te faudra créer un profil utilisateur. Si tu es l'administrateur du nouveau système d'exploitation, tu seras toujours habilité à utiliser les pouvoirs du **root** mais avec parcimonie et discernement... Et si possible à jeûn pour éviter d'effacer des fichiers de configuration nécessaires au bon fonctionnement de ta distribution.

1.4.1. Configuration du clavier avec la commande `loadkeys fr`

Je te laisse maintenant entrer la commande `loadkeys fr`. Comme ton clavier est configuré par défaut en QWERTY, il faut que tu tapes `loadkeys fr`.

```
Archiso login: root (automatic login)
```

```
root@archiso ~ #loadkeys fr
```

Ton clavier est désormais en français.

1.4.2. Tester la connexion au réseau et vérifier l'heure système

À présent, tu vas vérifier que tu es bien connecté(e) au réseau. Pour cela, entre la commande `ping` suivie d'une adresse quelconque. c'est comme un écho, en fait.

```
root@archiso ~ # ping cia.gov
```

Hein? L'adresse?... Non, ça veut dire *Confédération Internationale des Accordéonistes*... Si tu vois des lignes défiler, c'est que tu es connecté(e). Un petit **Ctrl+C** arrêtera le processus.

1.4.3. Timedatectl

Vérifions l'heure du système avec la commande **timedatectl**. C'est très important! Tu t'imagines bien qu'un ordinateur dont l'heure est configurée en 1974 ne risque pas d'installer des paquets de toute première fraîcheur!

```
root@archiso ~ # timedatectl
```

Si tu détectes une anomalie temporelle, je te suggère de mettre ton système à l'heure en utilisant la commande ci-dessous:

```
root@archiso ~ # timedatectl set-ntp true
```

Voilà pour ce premier chapitre. À présent, passons aux choses sérieuses...

Chapitre 2: Partitionner le disque dur

Ce deuxième épisode est consacré au partitionnement. Si je devais définir ce dernier, je dirais qu'il permet de diviser un disque dur en différents modules appelés **partitions**. Le but de cette opération est de classer proprement et de manière cohérente toutes les données que l'ordinateur va traiter ou qui sont nécessaires au bon fonctionnement du système d'exploitation. Chaque partition va accueillir un système de fichiers.

2.1. Déterminer le nom de son disque grâce à `fdisk`

Pour déterminer le nom du disque à partitionner, tu vas utiliser **`fdisk`**, un utilitaire présent dans l'image *.iso sur la clé USB. Pour le lancer, il suffit d'entrer cette commande:

```
root@archiso ~ # fdisk -l
```

`fdisk -l` retourne les informations en rouge ci-dessous. Là, je suis sur mon ordinateur portable. Il est équipé d'un unique disque dur d'une capacité de 1000 GB, c'est-à-dire 1 To. Il est baptisé **`/dev/sda`**. C'est donc celui-ci que je vais partitionner.

```

Disque /dev/sda : 931,5 GiB, 1000204886016 octets, 1953525168 secteurs
Unités : secteur de 1 x 512 = 512 octets
Taille de secteur (logique/physique) : 512 octets / 4096 octets
Taille d'E/S (minimale/optimale) : 4096 octets / 4096 octets
Type d'étiquette de disque : dos
Identifiant de disque : 0xd1f1a86e

Périphérique  Amorçage  Début  Fin          Secteurs  Taille  Id  Type
/dev/sda1     *          2048   1953723711  1953521664  931,5G  83  Linux

```

2.2. Choisir le type de la table de partitionnement

Maintenant, tu vas entrer la commande **cfdisk**. Elle va te permettre de créer la table de partitionnement et les partitions:

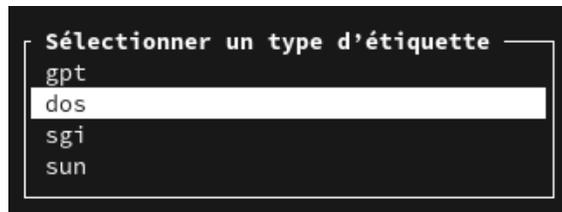
```
root@archiso ~ # cfdisk
```

La table de partitionnement correspond au sommaire des partitions. Pour un livre, c'est ce qu'on appelle la table des matières. Il existe deux types de tables qui sont **MS-DOS** et **GPT**. Choisis:

- **dos** si ton ordinateur est amorcé par un BIOS,
- **gpt** si ton ordinateur est récent et qu'il est amorcé par UEFI.

Personnellement, Je choisis **dos** parce que c'est un peu plus simple mais il faut savoir que sur mon ordinateur portable qui supporte

l'**UEFI**, j'ai choisi **gpt**. Je détaillerai donc toujours les deux cas de figure. Pour sélectionner, déplace-toi à l'aide des quatre flèches et valide en pressant la touche **Entrée**.



2.3. Partitionner son disque dur en BIOS

Les choses sérieuses commencent... Tu vas créer trois partitions:

- La première aura une taille de **35 GB**. Elle accueillera tous les fichiers de configuration nécessaire au bon fonctionnement du système d'exploitation. En outre, cette partition sera amorçable, c'est-à-dire que c'est elle qui lancera la procédure de démarrage. C'est une partition qui sera montée à la racine du système. Cette racine est symbolisé par une barre oblique /.
- La deuxième aura une taille de **8 GB**. Ce sera la partition dédiée à l'espace d'échange qu'on appelle aussi la swap. Donne-lui la taille qui correspond à celle de ta mémoire vive (**RAM**). Cette

petite partition a pour mission de décharger la mémoire vive lorsque celle-ci est presque pleine.

- La troisième occupera la taille restante et sera dédié au **/home**, c'est-à-dire au répertoire qui contient le répertoire de l'utilisateur (ou des utilisateurs, il peut y en avoir plusieurs). C'est ton répertoire personnel. Tu pourras y stocker tes documents, tes photos, tes vidéos, ta musique, tes téléchargements ou que sais-je encore... Tout au long de ce petit manuel, je nommerai le répertoire utilisateur **/home/\$user**. Mon propre répertoire se nomme **/home/miamondo**.

2.3.1. Première partition

Pour créer la première partition, il suffit de sélectionner **Nouvelle** à l'aide des flèches du clavier, et de valider avec la touche **Entrée**.

```

Disque : /dev/sda

Taille 931,5 GiB, 1000204886016 octets, 1953525168 secteurs

Étiquette : dos, identifiant : 0xd1f1a86e

Périphérique  Amorçage  Début          Fin  Secteurs  Taille  Id Type
--> Espace libre  2048  1953525167  1953523120  931,5G

UUID du système de fichiers : f128b9f8-b46a-4af6-ba2d-e2e8f127c7d4
Système de fichiers ext4
Point de montage : / (démonté)

[ Nouvelle ] [ Quitter ] [ Aide ] [ Écrire ] [ Sauvegarder ]

```

Puis, tu renseignes la taille de cette première partition, soit **35G**, avant de valider (toujours avec la touche **Entrée**).

```

          Étiquette : dos, identifiant : 0xd1f1a86e
Périphérique Amorçage Début          Fin  Secteurs  Taille  Id Type
>> Espace libre      2048 1953525167 1953523120 931,5G

UUID du système de fichiers : f128b9f8-b46a-4af6-ba2d-e2e8f127c7d4
Système de fichiers ext4
Taille de partition : 35G
Peut être suivi de M pour Mio, G pour Gio, T pour Tio ou s pour secteur

```

Ensuite, tu sélectionnes **primaire** et tu valides.

```

          Disque : /dev/sda
          Taille 931,5 GiB, 1000204886016 octets, 1953525168 secteurs
          Étiquette : dos, identifiant : 0xd1f1a86e
Périphérique Amorçage Début          Fin  Secteurs  Taille  Id Type
>> Espace libre      2048 1953525167 1953523120 931,5G

UUID du système de fichiers : f128b9f8-b46a-4af6-ba2d-e2e8f127c7d4
Système de fichiers ext4
Point de montage : / (démonté)

          [ primaire ] [ étendue ]
          0 primaire, 0 étendue, 4 libre

```

```

Disque : /dev/sda

Périphérique Amorçage Début Fin Secteurs Taille Id Type
/dev/sda1 2048 73402367 73400320 35G 83 Linux
>> Espace libre 73402368 1953525167 1880122800 896,5G

[ Nouvelle ] [ Quitter ] [ Aide ] [ Écrire ] [ Sauvegarder ]

Créer une nouvelle partition à partir de l'espace libre

```

La première partition **/dev/sda1** est configurée (Je l'ai surlignée en jaune). Une deuxième ligne apparaît (celle qui est en gris). Elle correspond à l'espace libre restant. Comme je l'ai précisé plus haut, la partition **/dev/sda1** est amorçable. Tu dois donc sélectionner cette fonctionnalité et valider ton choix, après t'être repositionné(e) sur la ligne **/dev/sda1**.

```

Périphérique Amorçage Début Fin Secteurs Taille Id Type
/dev/sda1 2048 73402367 73400320 35G 83 Linux
>> Espace libre 73402368 1953525167 1880122800 896,5G

Type de la partition : Linux (83)
UUID du système de fichiers : f128b9f8-b46a-4af6-ba2d-e2e8f127c7d4
Système de fichiers ext4
Point de montage : / (démonté)
[amorçable] [supprimer] [resize] [quitter] [type] [aide] [écrire] [sauvegarder]

Modifier l'indicateur d'amorçage de la partition actuelle

```

Un astérisque va apparaitre dans la colonne **amorçage**.

Périphérique	Amorçage	Début	Fin	Secteurs	Taille	Id	Type
/dev/sda1	*	2048	73402367	73400320	35G	83	Linux
>> Espace libre		73402368	1953525167	1880122800	896,5G		

Voilà, ta première partition baptisée **/dev/sda1** est totalement configurée mais elle n'est pas encore écrite sur le disque. Tu exécuteras cette opération plus tard. À présent, tu descends d'un cran et tu mets **Espace libre** en surbrillance.

Périphérique	Amorçage	Début	Fin	Secteurs	Taille	Id	Type
/dev/sda1	*	2048	73402367	73400320	35G	83	Linux
>> Espace libre		73402368	1953525167	1880122800	896,5G		

[Nouvelle] [Quitter] [Aide] [Écrire] [Sauvegarder]

Créer une nouvelle partition à partir de l'espace libre

2.3.2. Seconde partition

Tu vas maintenant créer la partition dédiée à l'espace d'échange (**swap**), d'une capacité équivalente à celle de ta mémoire vive (**RAM**).

```

Disque : /dev/sda

Taille 931,5 GiB, 1000204886016 octets, 1953525168 secteurs

Étiquette : dos, identifiant : 0xd1f1a86e

Périphérique Amorçage Début Fin Secteurs Taille Id Type
/dev/sda1 * 2048 73402367 73400320 35G 83 Linux
>> Espace libre 73402368 1953525167 1880122800 896,5G

Taille de partition : 8G

Peut être suivi de M pour Mio, G pour Gio, T pour Tio ou S pour secteur

```

Et tu vas modifier le type car c'est une partition un peu spéciale. Elle ne stocke pas de données ou du moins seulement pour soulager la RAM lorsque celle-ci est au bord de l'indigestion. Théoriquement, tu n'es pas obligé(e) de créer une partition SWAP, mais c'est vivement recommandée.

```

Périphérique Amorçage Début Fin Secteurs Taille Id Type
/dev/sda1 * 2048 73402367 73400320 35G 83 Linux
>> /dev/sda2 73402368 90179583 16777216 8G 83 Linux
Espace libre 90179584 1953525167 1863345584 888,5G

Type de la partition : Linux (83)
[amorçable] [supprimer] [resize] [quitter] [type] [aide] [écrire] [sauvegarder]

Modifier le type de partition

```

Je disais donc qu'il fallait modifier le type. Tu vas sélectionner **82** Partition d'échange.

```

Sélectionner un type de partition
...
70 DiskSecure Multi-Boot
75 PC/IX
80 Minix ancienne
81 Minix / Linux ancienne
82 Partition d'échange Linux / Solaris
83 Linux
84 OS/2 hidden or Intel hibernation
85 Linux étendue
...

```

Voilà, tu viens de créer la deuxième partition (en jaune), en l'occurrence la **swap**. Une troisième ligne apparaît. Elle correspond à l'espace libre restant.

```

Disque : /dev/sda

Périphérique Amorçage Début Fin Secteurs Taille Id Type
/dev/sda1 * 2048 73402367 73400320 35G 83 Linux
/dev/sda2 73402368 90179583 16777216 8G 82 swap
>> Espace libre 90179584 1953525167 1863345584 888,5G

```

2.3.3. Troisième partition

Pour créer la troisième et dernière partition, celle qui sera montée dans le **/home**, il faut s'y prendre exactement comme pour la première, sauf que tu lui attribues toute la taille restante, c'est-à-dire dans cet exemple, **888,5 GB**.

```

Disque : /dev/sda

Taille 931,5 GiB, 1000204886016 octets, 1953525168 secteurs
Étiquette : dos, identifiant : 0xd1f1a86e

Périphérique Amorçage Début Fin Secteurs Taille Id Type
/dev/sda1 * 2048 73402367 73400320 35G 83 Linux
/dev/sda2 73402368 90179583 16777216 8G 82 swap
>> Espace libre 90179584 1953525167 1863345584 888,5G

Taille de partition : 888,5G

Peut être suivi de M pour Mio, G pour Gio, T pour Tio ou S pour secteur

```

2.4. Écriture de la table des partitions

Toutes les partitions sont définies et configurées. Si tu regrettes ce que tu viens de faire parce que tu réalises un peu tard que tu as encore sur le disque dur, de vieilles photos de famille où l'on voit Tonton Alphonse, légèrement éméché, tenter de grimper sur la table avec son déambulateur pour chanter *Riiiiikita, jolie fleur de Java!*, rassure-toi, il n'est pas trop tard pour les récupérer. Tu as certes configuré les partitions mais tu ne les as pas encore écrites sur le disque dur. Par conséquent, tu peux toujours interrompre le processus. Mais si tu valides avec l'onglet **Écrire** et que tu confirmes ton choix, la nouvelle table de partitions effacera à jamais les lamentables pitreries de Tonton Alphonse.

```

Disque : /dev/sda

Périphérique Amorçage Début Fin Secteurs Taille Id Type
/dev/sda1 * 2048 73402367 73400320 35G 83 Linux
/dev/sda2 73402368 90179583 16777216 8G 82 swap
>> /dev/sda3 90179584 1953525167 1863345584 888,5G 83 Linux

Type de la partition : Linux (83)
[amorçable] [supprimer] [resize] [quitter] [type] [aide] [écrire] [sauvegarder]
Écrire la table de partitions sur le disque (des données peuvent être détruites)

```

Pour résumer, tu viens de diviser ton disque dur en trois partitions qui sont:

```
/dev/sda1 * 35G 83 Linux
/dev/sda2 8G 82 SWAP (partition d'échange)
/dev/sda3 888,5G 83 Linux
```

2.5. Partitionner son disque dur en UEFI

Si ton ordinateur est doté de l'**UEFI**, le partitionnement diffère quelque peu. Il faudra d'abord créer une partition dédiée au **/boot/efi** d'une capacité de **300 Mo**, par conséquent une toute petite partition. Mais elle est nécessaire.

- **/dev/sda1** pour le **/boot/efi**: **300M** type **EFI system**.
- **/dev/sda2** pour le **(racine)**: **35G** type **Linux filesystem**.
- **/dev/sda3** pour la **swap**: **8G** type **Linux swap**.
- **/dev/sda4** pour le **/home**: type **Linux filesystem**.

Conclusion

Voilà pour aujourd'hui. Que ce soit en **BIOS** ou en **UEFI**, ton disque dur est partitionné. À présent, tu peux sortir de l'utilitaire **cmdisk** (onglet **quitter**). La prochaine étape consistera à formater et à

monter les partitions pour pouvoir accéder aux données qu'elles contiennent.

Chapitre 3: le formatage des partitions

Le formatage est l'opération qui convertit une partition en système de fichiers et qui permet donc de classer les données qu'elle contient, dans un ordre logique au lieu de les stocker dans un tas informe.

3.1. Les différents systèmes de fichiers

Il existe une multitude de systèmes de fichiers différents, mais sache que le système de fichiers le plus utilisé sur une distribution Linux, est **ext4**.

3.2. Les commandes de formatage **mkfs** et **mkswap**

Pour formater les trois partitions que tu as créées au chapitre précédent, tu vas utiliser les commandes **mkfs** et **mkswap**.

mkfs s'utilise en lui adjoignant le système de fichier à l'aide d'un point, le tout suivi de la partition à formater. Attention, il y a un espace entre la commande de formatage et la partition.

mkswap formate l'espace d'échange, c'est-à dire la partition swap. Elle ne prend pas de système de fichiers en attribut. Là encore, il y a un espace entre la commande et la partition.

Donc, dans le code-ci-dessous :

- tu formates la partition **/dev/sda1** en un système de fichiers de type **ext4**.
- Tu formates la **swap**.
- Tu formates la partition **/dev/sda3** en un système de fichiers de type **ext4**.

```
root@archiso ~ # mkfs.ext4 /dev/sda1
```

```
root@archiso ~ # mkswap /dev/sda2
```

```
root@archiso ~ # mkfs.ext4 /dev/sda3
```

3.3. Variante UEFI

ATTENTION! ne formatez la partition **/dev/sda1** que s'il n'y a pas d'autres systèmes d'exploitation présents sur votre machine.

```
root@archiso ~ # mkfs.vfat -F32 /dev/sda1
```

```
root@archiso ~ # mkfs.ext4 /dev/sda2
```

```
root@archiso ~ # mkswap /dev/sda3
```

```
root@archiso ~ # mkfs.ext4 /dev/sda4
```

3.4. Le montage des partitions en mode BIOS

Les partitions sont formatées. Maintenant, il reste à définir les points de montage, c'est-à-dire l'endroit précis où le contenu de ces partitions sera accessible. Ce sont comme des portes en fait. Lorsqu'on monte une partition, on ouvre une porte à clé. Lorsqu'on démonte une partition, on sort et on referme la porte à clé.

Les trois partitions seront montées automatiquement au démarrage de l'ordinateur et démontées automatiquement à l'extinction de ce dernier, grâce à une commande que tu découvriras au prochain chapitre. Là encore, tu vas effectuer cette opération en console et tu vas te rendre compte que ce n'est pas bien compliqué.

Les commandes qui permettent de monter et démonter une partition sont **mount** et **umount**. Tu vas monter les partitions sur un point de montage de la clé d'installation. Ce point de montage s'appelle **/mnt**. C'est ici que tu vas commencer à bâtir Archlinux sur ton disque dur. Ce point de montage deviendra donc la racine de ton nouveau système. Lorsque l'installation sera terminée, il se réduira à une simple barre oblique: **/**

Pour que tu comprennes bien comment cela fonctionne, lorsque tu listes les répertoires qui sont à la racine de la clé USB d'installation avec la commande **ls**, voici ce que tu obtiens:

```
root@archiso ~ # ls /  
bin boot dev etc home lib lib64 mnt (...) tmp usr var  
root@archiso ~ # ls /mnt  
(vide)
```

Tout un tas de répertoires d'installation et de configuration! **/mnt** en fait partie et si tu le listes, tu constates que ce dernier est vide. C'est normal. Il ne sert que de point de montage pour une installation permanente. C'est le point de passage entre la clé USB d'installation et ton futur système d'exploitation Archlinux.

À présent, tu vas monter la première partition. la commande n'est pas bien compliquée. Je **mount /dev/sda1** sur **/mnt**.

```
root@archiso ~ # mount /dev/sda1 /mnt
```

Cette partition formatée en **ext4** est destinée à accueillir les fichiers de configuration du système, c'est-à-dire:

```
/bin /boot /dev /etc /home /lib (...) /var
```

La **swap**, c'est un peu spécial. Pour l'activer, il suffit d'entrer cette commande:

```
root@archiso ~ # swapon /dev/sda2
```

Maintenant, tu vas monter la troisième partition, celle destinée à accueillir le répertoire de l'utilisateur (ou des utilisateurs). Mon répertoire personnel s'appelle **miamondo** et il contient les sous-répertoires suivants:

- **Documents**
- **Images**
- **Videos**
- **etc...**

Le répertoire personnel sera monté non pas à la racine du système mais dans **/home**. Or, **/home** n'existe pas encore sur le système que tu es en train de bâtir. Il faut donc le créer avec la commande **mkdir(make directory = créer répertoire)**.

```
root@archiso ~ # mkdir /mnt/home && mount /dev/sda3 /mnt/home
```

Cette commande signifie donc:

- que tu crées un répertoire **/mnt/home**
- ET que tu montes la partition **/dev/sda3** sur **/mnt/home**

Pour résumer, voici ce que tu viens de faire:

```
root@archiso ~ # mount /dev/sda1 /mnt
root@archiso ~ # swapon /dev/sda2
root@archiso ~ # mkdir /mnt/home && mount /dev/sda3 /mnt/home
```

3.5. Le montage des partitions en mode UEFI

Dans la variante **UEFI**, l'option **-p** de la commande **mkdir** permet de créer des répertoires intermédiaires sans générer d'erreur si ces derniers existent déjà. Ici, c'est le cas de **/boot**.

```
root@archiso ~ # mount /dev/sda2 /mnt
root@archiso ~ # mkdir -p /mnt/boot/efi /mnt/home
root@archiso ~ # swapon /dev/sda3
root@archiso ~ # mount /dev/sda1 /mnt/boot/efi
root@archiso ~ # mount /dev/sda4 /mnt/home
```

Conclusion

Dans le prochain chapitre, tu vas installer le système de base et schtroumpfer à l'intérieur de celui-ci.

Chapitre 4: installer la base du système

4.1 Démonter et remonter les partitions

À ce stade de l'installation, tu peux très bien faire une pause-café et éteindre ton ordinateur. Mais il ne faut pas oublier de démonter les partitions au préalable. Pour cela, tu vas utiliser les commandes `umount` et `swapoff`.

```
root@archiso ~ # umount -R /mnt
```

```
root@archiso ~ # swapoff /dev/sda2 (/dev/sda3 pour la variante UEFI)
```

La première commande démonte de manière récursive (`-R`), toutes les partitions montées sur `/mnt` ou sur un de ses répertoires enfants. Dans la version **UEFI**, C'est le cas par exemple de `/dev/sda1` qui est monté sur `/mnt/boot/efi`.

La deuxième commande désactive la **SWAP**

4.2. Sélectionner le miroir

Lorsque la pause est terminée, il est temps de continuer l'installation. Tu rallumes l'ordinateur en le faisant rebooter sur la clé USB d'installation. **Il faut de nouveau passer le clavier en français et monter les partitions en mode BIOS ou en mode UEFI.**

Tu vas télécharger la base de ton nouveau système. Mais avant cela, pour sélectionner le miroir le plus proche de toi géographiquement, tu vas éditer ce fichier de configuration: **/etc/pacman.d/mirrorlist**. Pour ce faire, tu as à ta disposition un éditeur de texte qui s'utilise en console et qui s'appelle **nano**. Voici la commande:

```
root@archiso ~ # nano /etc/pacman.d/mirrorlist
```

Ce fichier se trouve donc sur la clé USB d'installation et pas sur le disque dur puisque son chemin absolu ne commence pas par **/mnt**. Te voici maintenant à l'intérieur d'un fichier qui contient des centaines d'adresses de serveurs disséminés dans le monde entier. Ils correspondent aux miroirs de téléchargement d'Archlinux. Si, comme moi, tu habites dans la région de Hanovre. Tu vas sélectionner le miroir de l'Université de Hanovre.

```
## Germany  
Server = http://ftp.uni-hannover.de/archlinux/$repo/os/$arch
```

Tu vas donc désactiver la connexion aux autres serveurs en remplaçant toutes les occurrences **server** par **#server**. En effet, le rajout d'un croisillon transforme la ligne en commentaire et la rend inexécutable. Le seul serveur que tu ne vas pas **commenter** (puisque c'est comme cela qu'on dit), sera celui de l'Université de Hanovre.

- C'est parti! Tu presses les touches **Alt+R**
- Tu entres le mot **Server** et tu presses la touche **Entrée**,
- Tu entres le mot **#Server** et tu presses la touche **Entrée**,
- Tu modifies toutes les occurrences en pressant la combinaison de touches **Maj+A**,
- Tu presses les touches **Ctrl+W** et tu entres le mot **hannover** avant de presser la touche **Entrée**. Cette opération va placer le curseur sur la ligne qui t'intéresse, à savoir le serveur de Hanovre.
- Tu déplaces le curseur en début de ligne. Tu supprimes le croisillon, ce qui a pour effet de **décommenter** la ligne et de la rendre exécutable.

- Tu presses les touches **Ctrl+X** pour enregistrer les modifications. On te demande de confirmer par un **Y(es)** ou un **J(a)** ou un **O(ui)**.
- Tu presses la touche **Entrée**. Le fichier est modifié.

4.3. Installer la base

À ce stade, pense à vérifier ta connexion au réseau:

```
root@archiso ~ # systemctl status dhcpcd
```

Si cette commande te renvoie des instructions contenant le mot **inactive**, fais ceci:

```
root@archiso ~ # systemctl start dhcpcd
```

À présent, tu peux passer à l'installation de base proprement dite, avec les métapaquets **base** et **base-devel**. C'est la commande **pacstrap** qui va s'en charger. J'en profite également pour installer **dialog**, une application qui affiche des boîtes de dialogue en console. C'est juste pour mettre un peu de fun dans la post-installation ;)

```
root@archiso ~ # pacstrap /mnt base base-devel pacman-contrib linux  
linux-firmware dialog wget zip unzip nano dhcpcd
```

Cette commande est entrée sur une seule ligne. Ne fais pas de retour à la ligne. L'installation peut s'étaler sur plusieurs minutes. Tout un tas de lignes de commande défile sur l'écran. Lorsque c'est terminé, tu peux lister `/mnt` et constater de visu, qu'une palanquée de fichiers de configuration a été installée avec succès sur ton tout nouveau système.

4.4. Configurer le nouveau système

4.4.1. Générer le fichier `/mnt/etc/fstab`

Maintenant, tu vas générer le fichier `/mnt/etc/fstab`. C'est lui qui définit comment les partitions doivent être montées. Il ne doit pas être incohérent avec ta table de partitions, sous peine de retarder considérablement le démarrage de l'ordinateur.

```
root@archiso ~ # genfstab -U -p /mnt >> /mnt/etc/fstab
```

4.4.2. Chrooter dans le nouveau système

Il est venu le temps de schtroumpfer dans ton système tout frais installé. Ce n'est pas sans une certaine émotion que grâce à la commande ci-dessous, tu franchis la porte `/mnt` pour te retrouver sur ton disque dur:

```
root@archiso ~ # arch-chroot /mnt
```

```
[root@archiso /]#
```

Tu noteras que l'invite de commande a changé, tout comme la racine du système qui n'est plus `/mnt` mais simplement `/`.

4.4.3. Renseigner le nom de l'ordinateur

Dans le fichier `/etc/hostname`:

```
[root@archiso /]# echo volgor > /etc/hostname
```

C'est quoi cette histoire d'écho? Que fait cette commande au juste? Eh bien **echo** signifie tout simplement **écrire**. Quant au simple chevron, il signifie «dans le fichier qui suit en écrasant tout le contenu qui s'y trouve». Donc, écris **volgor** dans le fichier `/etc/hostname`, après avoir écrasé son éventuel contenu. Si ce fichier

n'existe pas, prière de le créer. Bien entendu, si ton ordinateur s'appelle **Bourvil**, tu écris ce nom d'hôte à la place de **volgor**.

4.4.4. Créez un lien symbolique pour le fuseau horaire

Pour l'Allemagne:

```
[root@archiso /]# ln -sf /usr/share/zoneinfo/Europe/Berlin /etc/localtime
```

`ln -sf` crée le lien symbolique `/etc/localtime` qui pointe vers le fichiersuivant: `/usr/share/zoneinfo/Europe/Berlin`

4.4.5. Sélectionner la langue du système

C'est ce qu'on appelle la **locale**. Pour la sélectionner, tu vas éditer le fichier `/etc/locale.gen` et décommenter, c'est-à-dire supprimer le croisillon qui se trouve au début de ta locale. Ainsi, l'instruction deviendra exécutable.

```
[root@archiso /]# nano /etc/locale.gen
```

Si ta locale est notre belle langue française, supprime le croisillon devant `fr_FR.UTF-8 UTF-8`.

```
#fr_CA ISO-8859-1
#fr_CH ISO-8859-1
fr_FR.UTF-8 UTF-8
#fr_FR ISO-8859-1
```

Enregistre avec **Ctrl+X**. Puis, confirme avec **Y(es)** et exécute la commande suivante:

```
[root@archiso /]# locale-gen
```

4.4.6. Configuration de la locale pour tout le système

Tu rajoutes le nom de la locale au fichier `/etc/locale.conf`.

```
[root@archiso /]# echo LANG="de_DE.UTF-8" > /etc/locale.conf
```

Tu peux vérifier que le fichier a été correctement modifié grâce à la commande `cat`.

```
[root@archiso /]# cat /etc/locale.conf
```

4.4.7. Disposition du clavier

Tu dois bien te douter qu'il est plus facile de travailler avec un clavier dont la disposition des touches correspond à ce que tu tapes. Par conséquent, édite le fichier `/etc/vconsole.conf`. Si ton clavier est un QWERTZ allemand, voici la commande:

```
[root@archiso /]# echo KEYMAP=de > /etc/vconsole.conf
```

Si c'est un AZERTY français:

```
[root@archiso /]# echo KEYMAP=fr > /etc/vconsole.conf
```

4.4.8. Générer l'image du noyau Linux avec mkinitcpio

```
[root@archiso /]# mkinitcpio -p linux
```

Si la commande `mkinitcpio` te retourne `command not found`, installe-la de cette manière:

```
[root@archiso /]# pacman -S mkinitcpio
```

4.4.9. Définir un mot de passe pour l'utilisateur root

```
[root@archiso /]# passwd
```

Suivre les instructions

4.5. Installer le chargeur d'amorçage

Avant de rebooter et de profiter enfin de ton nouveau système d'exploitation, il faut que tu installes GRUB qui est un chargeur d'amorçage (**bootloader**, en anglais). C'est comparable à l'étincelle qui allume les brûleurs de ta gazinière en quelque sorte.

4.5.1. En mode BIOS

```
[root@archiso /]# pacman -S grub
```

```
[root@archiso /]# grub-install --no-floppy --recheck /dev/sda
```

```
[root@archiso /]# grub-mkconfig -o /boot/grub/grub.cfg
```

4.5.2 En mode UEFI

```
[root@archiso /]# pacman -S grub efibootmgr
```

```
[root@archiso /]# grub-install --target=x86_64-efi --efi-  
directory=/boot/efi --bootloader-id=arch_grub --recheck
```

```
[root@archiso /]# grub-mkconfig -o /boot/grub/grub.cfg
```

Attention! La deuxième commande s'écrit sur une seule ligne. Pas de retour à la ligne.

Pour la première fois, tu as utilisé une drôle de commande, en l'occurrence **pacman**. J'en parlerai davantage au prochain chapitre. C'est une commande que tu vas désormais utiliser très souvent, Il s'agit en fait du **gestionnaire de paquets**. Grâce à ce dernier, tu viens d'installer **grub** et **efibootmgr**.

Si vraiment cette opération échoue suite à une réinstallation, utilise la commande ci-dessous qui va "atomiser" les premiers secteurs du disque dur et te permettre de recommencer le processus d'installation de **GRUB**. Il faut bien avouer que ce dernier n'est pas toujours simple à installer.

```
[root@archiso /]# dd if=/dev/zero of=/dev/sda seek=1 count=2047
```

4.6. Redémarrage

À présent, tu peux quitter le **chroot**, c'est-à-dire sortir du système nouvellement installé pour retourner une dernière fois dans la clé USB d'installation.

```
[root@archiso /]# exit
```

Démonte les partitions et redémarre ton ordinateur.

```
root@archiso ~ # umount -R /mnt
```

```
root@archiso ~ # reboot
```

Tu n'as plus besoin de la clé USB d'installation. Retire-la. Au prochain démarrage, si tout s'est déroulé comme prévu, c'est la fenêtre d'ouverture du chargeur d'amorçage qui va t'accueillir. La prochaine étape va consister à installer tous les paquets nécessaires pour créer une distribution agréable et fonctionnelle.

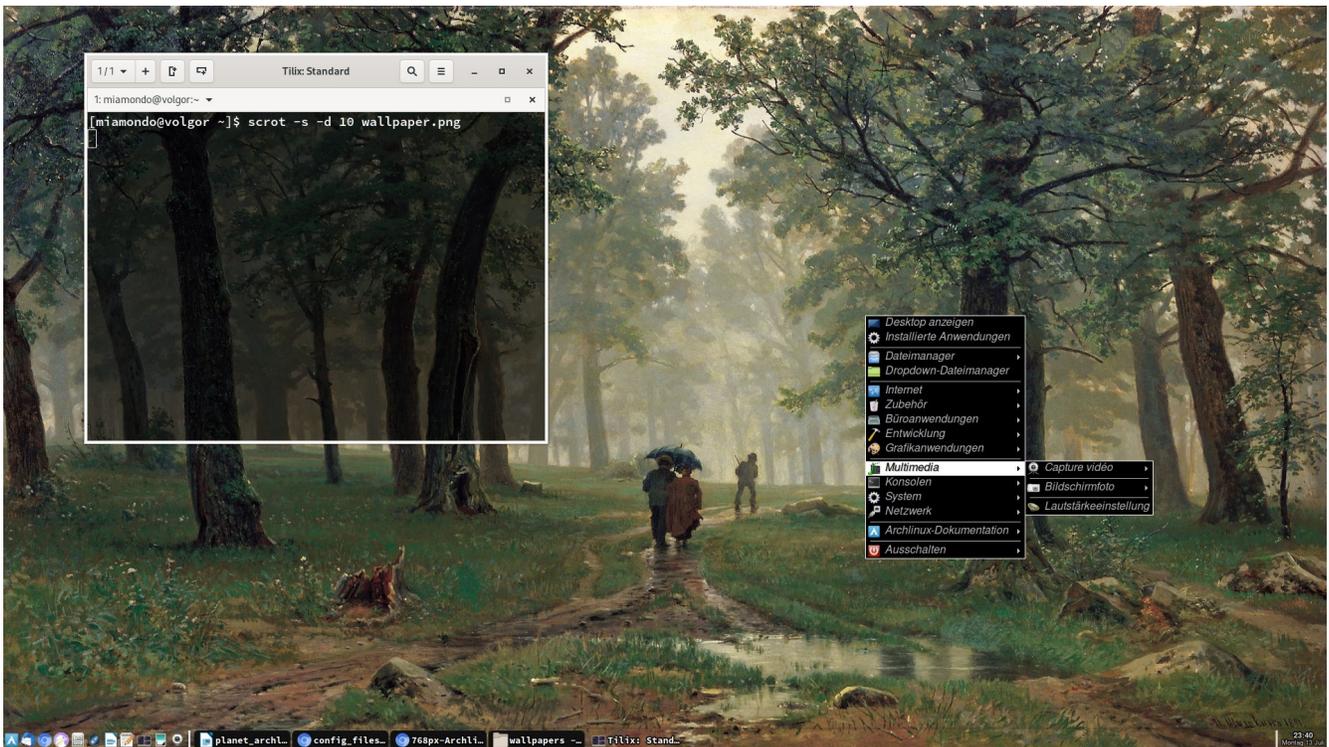
Chapitre 5: post-installation

Ton nouveau système Archlinux est opérationnel. Cela dit, ta mission n'est pas terminée. Tu dois encore installer un environnement graphique digne de ce nom. Pour l'instant, c'est une console noire qui t'accueille, et même si tu peux dès à présent accomplir beaucoup de choses en ligne de commande, une interface graphique est tout de même plus sympa.

Archlinux adhère au concept du KISS. Et ça tombe bien parce que moi aussi. C'est la raison pour laquelle, je te propose d'installer non pas un environnement de bureau mais:

- Le gestionnaire de fenêtres **openbox**, en apportant un soin tout particulier au menu,
- Le gestionnaire de fonds d'écran **nitrogen**,
- La barre de tâches **tint2**.

Tu vas donc bâtir un système d'exploitation léger, fonctionnel et facilement configurable. J'ai rédigé un petit script de post-installation accompagné de quelques fichiers de configuration. Si je n'ai pas fait de bêtises, voici le résultat que tu devrais obtenir au final.



Le fond d'écran est un tableau d'Ivan Chichkine intitulé *Pluie dans le bois de chênes*. J'aime beaucoup le réalisme de ce peintre russe. Le menu est celui d'**Openbox** en version **Onyx**, un thème sombre. Le thème clair **Natura** est bien aussi. Le menu peut s'ouvrir avec un clic droit n'importe où sur le bureau, ou bien en cliquant sur l'icône tout en bas à gauche (C'est une commande **xdotool**). J'ai également rajouté des icônes pour illustrer chaque onglet du menu. En fin d'article, je vais t'expliquer comment j'ai procédé. Ce n'est vraiment pas compliqué.

5.1. Lancement du script de post-installation

Dans ton nouveau système Archlinux, connecte-toi en **root**, en utilisant le mot de passe créé à l'étape précédente. Ensuite, vérifie la connexion au réseau.

```
root@archiso ~ # systemctl status dhcpcd
```

Si cette commande te renvoie des instructions contenant le mot **inactive**, fais ceci:

```
root@archiso ~ # systemctl start dhcpcd
```

```
root@archiso ~ # systemctl enable dhcpcd
```

La première démarre le service immédiatement, tandis que la seconde t'assure que le service sera démarré à chaque fois que tu allumeras ton ordinateur. Si tu veux vérifier que tout est en ordre, entre cette commande:

```
root@archiso ~ # systemctl status dhcpcd
```

Le mot **active** devrait apparaître.

Télécharge le lien ci-dessous en utilisant la commande `wget`. Il pointe vers mon dépôt `git`. Oui, je sais, il est long mais il fonctionne à tous les coups! **Il est sur une seule ligne. Donc, tu le rentres tel quel, sans te soucier du retour à la ligne. le terminal s'en charge tout seul.**

```
root@archiso ~ # wget --no-check-certificate https://gitlab.com/miamondo/post-
installation-of-archlinux/-/archive/master/post-installation-of-archlinux-
master.zip
```

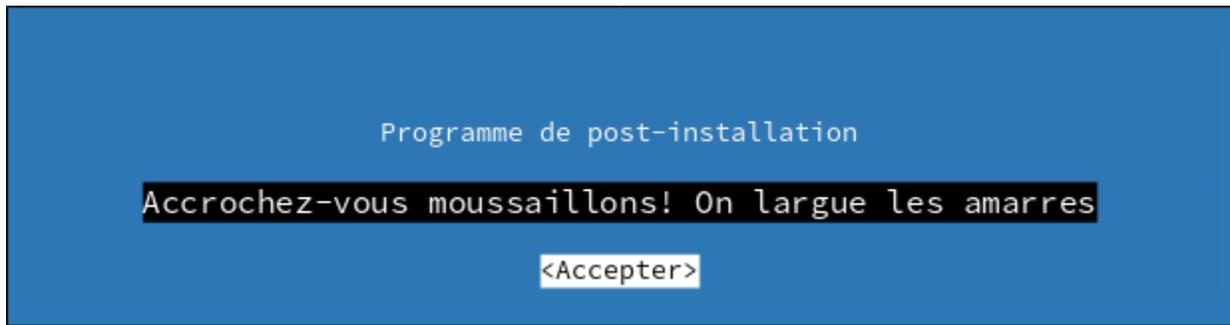
Puis, entre cette drôle de commande confectionnée par mes soins:

```
root@archiso ~ # unzip *.zip && mv *master/*.sh ~ && chmod +x *.sh && ./*.sh
```

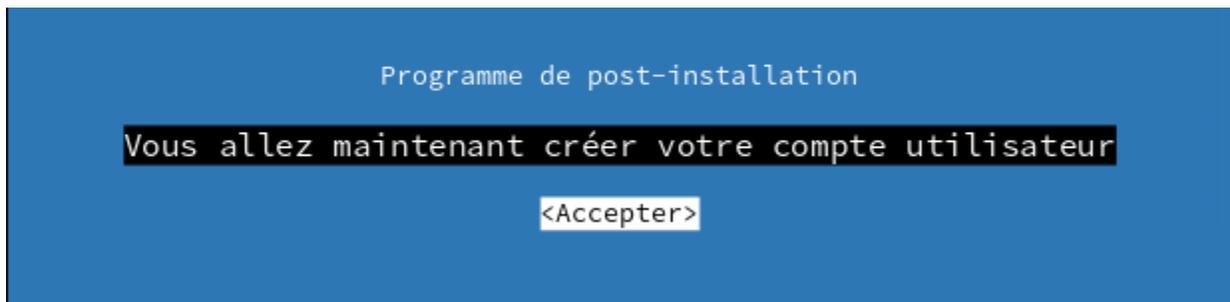
En fait, cette commande correspond exactement à ceci:

```
root@archiso ~ # unzip post-installation-of-archlinux-master.zip && mv post-
installation-of-archlinux-master/post_install-archlinux.sh ~ && chmod +x
post_install-archlinux.sh && post_install-archlinux.sh
```

Normalement, tu vas être accueilli par le message ci-dessous. Il t'annonce que la fête va bientôt commencer. Toutes les applications contenues dans le fichier `pkglist.txt` vont être installées. Si on te pose des questions, choisis toujours la réponse par défaut en pressant la touche **Entrée**.



Après cette opération, tu vas être invité à créer ton compte utilisateur. Tu seras d'emblée rajouté au groupe **sudo**. L'installation continue et le système redémarre.



Après le redémarrage, l'écran de login va apparaître. Tu vas te connecter avec ton nouveau nom d'utilisateur et valider avec ton mot de passe. Au bout du compte, comme je l'ai dit plus haut, tu vas te retrouver avec un bureau simple, convivial et fonctionnel.

5.2. Analyse des différents scripts de post-installation

Intéressons-nous à présent aux différents scripts de post-installation.

5.2.1. Le script principal : `archlinux_post-install.sh`

Tu le trouveras [ici](#), sur mon [dépôt](#) git. Je l'ai largement commenté.

5.2.2. La liste des applications à installer: `pkglist.txt`

Cette liste est bien pratique! Elle contient une application par ligne. Elles seront toutes récupérées et traitées par le gestionnaire de paquets `pacman`.

5.2.3. le script `tint2rc`

Il configure la barre de tâches. Par la suite, si tu veux modifier cette dernière, je t'invite à utiliser l'outil graphique qui est bien plus simple. Celui-ci s'appelle **Gestionnaire du panneau tint2** et tu le trouveras dans le menu, sous l'onglet **applications installées** ou bien directement sous `/usr/share/applications`

5.2.4. Menu.desktop

Ce fichier crée un lanceur que j'ai placé dans la barre de tâches **tint2** et qui déroule le menu. J'utilise pour cela, l'outil **xdotool** qui permet de simuler une commande clavier par l'intermédiaire d'un simple clic. Normalement, pour dérouler le menu à l'endroit où se trouve le pointeur de la souris, il suffit de cliquer droit ou de presser les touches **super** et **m**. (**Super** correspond à la touche **windows**.) Avec la commande **xdotool key super+m**, le clic gauche sur le bouton du lanceur remplace la commande clavier. Voici le contenu du fichier:

```
[Desktop Entry]

Type=Application
Exec=xdotool key super+m
Name=Menu principal
Categories=Utility;DesktopSettings;
Terminal=false
StartupNotify=true
Icon=/usr/share/icons/Faenza/places/48/distributor-logo-archlinux.png
Comment=Opens the applications menu
Comment[fr]=Ouvre le menu des applications
Keywords=menu;desktop;
```

5.2.5. menu.xml

C'est ce fichier qui va te permettre de rajouter ou supprimer des applications dans le menu. Une fois installé, son chemin est **/home/\$user/.config/openbox/menu.xml**. Tu peux le configurer à ta guise. Une fois qu'on a compris qu'il faut respecter les indentations

et que chaque balise ouverte doit être impérativement fermée, sa structure devient claire et son utilisation est intuitive. N'hésite pas à consulter la documentation officielle. Note que le répertoire des icônes est `/usr/share/icons`. J'ai préconfiguré ce menu, mais il se peut qu'il subsiste des incohérences.

5.2.6. rc.xml

Il s'agit du fichier de configuration d'**Openbox**. Tu peux être amené à le modifier si tu souhaites par exemple remplacer le thème **Onyx** par **Natura**. Tu trouveras les noms des différents thèmes dans le répertoire `/usr/share/themes`.

Cela dit, il existe une interface graphique qui fait très bien l'affaire, en l'occurrence **obconf**. Pour l'ouvrir, il te suffit d'entrer la commande **obconf** dans un terminal.

5.2.7. Le fichier post-install_yay.sh

Après le redémarrage de l'ordinateur, le gestionnaire de paquets **yay** va être installé. Il te permettra de compléter ta distribution avec des dépôts **AUR**, c'est-à-dire des dépôts proposés par des utilisateurs d'Archlinux.

5.3. Les gestionnaires de paquets pacman et yay

Archlinux met à disposition de ses utilisateurs deux gestionnaires de paquets qui sont **pacman** et **yay**.

5.3.1. pacman

C'est le gestionnaire officiel des paquets d'Archlinux. Il s'utilise en root et possède plusieurs options qui permettent de synchroniser, de mettre à jour, d'installer ou de supprimer des paquets. Je ne vais pas m'étendre davantage sur ce sujet. Il existe une page très bien faite qui explique tout de long en large et que je t'invite d'ores et déjà **à lire**. **pacman** est un outil que tu utiliseras quotidiennement.

5.3.2. yay

yay signifie **yet another yaourt**. C'est un gestionnaire qui permet d'installer des paquets issus du **dépôt des utilisateurs d'Archlinux (AUR)**. Il faut donc utiliser cet outil avec prudence car tous les paquets présents sur le dépôt AUR ne sont pas forcément vérifiés et maintenus. C'est sans doute la raison pour laquelle il ne m'a pas été possible de post-installer **yay** en root. Il a fallu que je rédige un **nouveau script** et que je modifie mon script principal de post-

installation pour lancer l'installation de **yay** au premier lancement de ma session utilisateur.

```
miamondo@volgor ~ $ git clone https://aur.archlinux.org/yay.git
```

```
miamondo@volgor ~ $ cd yay
```

```
miamondo@volgor ~ $ makepkg -si
```

Conclusion

Le voyage touche à sa fin. N'hésite pas à me faire part de tes critiques où des difficultés que tu as rencontrées lors de ce processus de post-installation. Je vais compléter et mettre à jour ce petit manuel régulièrement.

Bon courage