

Séminaire d'Installation d'Arch Linux

Département d'informatique

Eric Hervet et Andy Couturier

Faculté des sciences
Université de Moncton

20 mars 2025



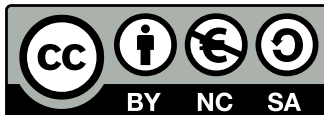
Table des matières

1. Pourquoi ?
2. Support d'installation
3. Partitionnement, formatage et montage
4. Installation
5. Chroot
6. Bootloader
7. Initramfs
8. Compression de la mémoire
9. Configurations importantes
10. Comptes utilisateurs
11. Arch User Repository (AUR)
12. Installation d'un bureau graphique
13. Redémarrage
14. Extras

Eric Hervet et Andy Couturier

Séminaire d'Installation d'Arch Linux • © ⓘ ⓘ ⓘ

Table des matières



Ce document est mis à disposition selon les termes de la licence **Creative Commons “Attribution – Pas d'utilisation commerciale – Partage dans les mêmes conditions 4.0 International”**.

1. Pourquoi ?

1 Pourquoi ?

- La plupart des distributions Linux ont un installateur graphique facile d'utilisation. Il existe même des scripts d'installation semi-automatiques pour Arch Linux.
- Le séminaire proposé consiste à installer Arch Linux manuelle avec des commandes dans un terminal.
- Mais pourquoi s'infliger des souffrances en installant manuellement Arch Linux ?
 - ⇒ **Pour apprendre.**
- Une personne informaticienne explore en profondeur pour comprendre le fonctionnement des systèmes, tandis qu'une personne technicienne se concentre sur leur mise en œuvre et leur maintenance. L'approche scientifique pousse toujours à approfondir la connaissance.
- En réalisant cette installation, vous allez :
 - Mieux comprendre la structure d'un système d'exploitation.
 - Comprendre le système de partitionnement et sa relation aux systèmes de fichiers.
 - Comprendre le processus de démarrage d'un système d'exploitation.
 - Apprendre à personnaliser et maîtriser votre environnement Linux.

2. Support d'installation

2 Support d'installation

- Pour créer une clé USB qui contient l'image d'installation d'Arch Linux :
 1. Rendez vous à l'adresse <https://archlinux.org/download/>.
 2. Choisissez un serveur Canadien dans la liste et cliquez sur son lien.
 3. Via la page d'index du serveur, téléchargez le fichier ISO qui aura un nom comme : archlinux-20XX.XX.01-x86_64.iso.
- Si vous êtes déjà sur Linux :
 1. Ouvrez un terminal.
 2. Branchez la clé USB qui servira de support d'installation.
 3. `$sudo dmesg` pour identifier le code sdX de la clé USB.
 4. Placez vous dans le répertoire où se trouve l'ISO en utilisant cd.
 5. `$sudo dd bs=32M if=archlinux-20XX.XX.01-x86_64.iso of=/dev/sdX conv=fsync oflag=direct status=progress`
- Si vous êtes sur Windows (nos sincères sympathies) : Vous pouvez utiliser Rufus.

3. Partitionnement, formatage et montage

3.1 Table de Partitionnement

- **Définition** : Une table de partitionnement est une structure située sur un disque décrivant comment l'espace de stockage est divisé en zones distinctes appelées partitions.
- **Rôle** : Elle détermine où chaque partition commence et se termine sur le disque et le type de système de fichiers de chaque partition.
- **Fonctionnement** :
 - Elle est placée généralement à un emplacement fixe sur le disque et est utilisée par l'ordinateur au démarrage pour détecter et gérer les partitions présentes.
 - Elle indique quelles partitions sont amorçables (« *bootable* »), aidant ainsi le processus de démarrage du système d'exploitation.
- **Systèmes de fichiers** :
 - Chaque partition peut être formatée avec un système de fichiers spécifique, permettant de stocker et d'organiser les fichiers.
 - Différents systèmes de fichiers peuvent coexister sur le même disque dans différentes partitions.

3.2 MBR vs GPT

- Il y a deux formats de table de partitionnement possible.
- **MBR (Master Boot Record)**
 - Taille maximale de disque : 2 TiB.
 - Limité à 4 partitions primaires.
 - Plus ancien, compatible avec BIOS uniquement.
 - Pas de redondance, si le MBR est corrompu, les données peuvent être inaccessibles.
- **GPT (GUID Partition Table)**
 - Taille maximale de disque : 9.4 ZiB.
 - Supporte jusqu'à 128 partitions primaires.
 - Utilise des identifiants globaux uniques (GUID) qui permettent l'identification précise des partitions avec un identifiant PARTUUID.
 - Plus moderne, conçu pour UEFI.
 - Redondance avec deux copies de la table, une au début et une à la fin du disque.
 - ▶ Inclut un hash CRC32 pour détecter la corruption de la table.

3.3 Table de partitionnement

#	Device	Mountpoint	Type	Taille
1	/dev/sdX1	/boot (/mnt/boot)	ef00	1 GiB
2	/dev/sdX2	/ (/mnt)	8300	Reste

- Les mountpoints entre parenthèses seront les mountpoints temporaires des partitions pendant l'installation. Ils se trouveront sur le système de fichier existant du support d'installation.
- Le type de partition `ef00` indique qu'il s'agit d'une partition EFI. Cela signifie que le système à un firmware UEFI pour le démarrage.
- La taille de 1 GiB est généralement suffisante pour stocker le noyau Linux et les fichiers de démarrage nécessaires.
- Le type de partition `8300` est l'identifiant pour une partition Linux standard sous le schéma de partitionnement GPT.
- La partition racine utilisera le reste de l'espace libre du disque.

3.4 Partitionnement

- Branchez la clé sur laquelle vous allez installer Arch Linux et ensuite faites immédiatement `$dmesg` pour identifier le code `sdX` de la clé USB.
- `$gdisk /dev/sdX` pour commencer à partitionner le disque interactivement.
 - `o` puis `y` pour effacer toutes les partitions.
 - `n` pour créer une nouvelle partition.
 - `Enter` pour choisir la partition par défaut (1).
 - `Enter` pour choisir le secteur de départ par défaut.
 - `+1G` pour choisir le secteur de fin 1 GiB après le secteur de départ.
 - `ef00` pour le type de partition.
 - `n` pour créer une nouvelle partition.
 - `Enter` pour choisir la partition par défaut (2).
 - `Enter` pour choisir le secteur de départ par défaut.
 - `Enter` pour choisir le secteur de fin par défaut (fin du disque).
 - `Enter` pour choisir type de partition par défaut (8300).
 - `p` pour vérifier le partitionnement.
 - `w` puis `y` pour écrire la table de partitions au disque.

3.5 Formatage de la partition de démarrage

- `$mkfs.fat -F32 /dev/sdX1`
- **Conformité UEFI (Unified Extensible Firmware Interface)** : Le standard UEFI requiert une partition système EFI (ESP) au format FAT32.
- **Pilotes UEFI intégrés** : Les firmwares UEFI incluent des pilotes FAT. Sans ce format, l'UEFI ne peut ni localiser ni charger le bootloader, rendant le démarrage impossible.
- **Compatibilité universelle** : FAT32 est pris en charge nativement par la plupart des systèmes d'exploitation, simplifiant la mise à jour de la partition (fichiers EFI, bootloaders, etc.) depuis n'importe quelle plateforme.

3.6 Formatage de la partition racine

- `$mkfs.f2fs /dev/sdX2`
- **F2FS (Flash-Friendly File System)** a été conçu spécifiquement pour les supports de stockage à base de mémoire flash (SSD, eMMC, clés USB, cartes SD).
- **Log-structured file system** : Les données modifiées sont toujours inscrites dans de nouveaux blocs, au lieu de les réécrire directement au même endroit. Cela permet au système de grouper plusieurs écritures aléatoires en flux d'écritures séquentielles. Cela réduit la fragmentation, améliore considérablement les performances d'écriture et permet une répartition plus uniforme des écritures ce qui prolonge la durée de vie du support (wear-leveling).
- **Garbage collection** : L'espace est divisé en segments, et ceux contenant des données obsolètes sont régulièrement regroupés en blocs contigus. Ce processus limite l'amplification d'écriture, maintient un stockage homogène et préserve durablement les performances.
- **TRIM** : Le TRIM indique au contrôleur flash quels blocs sont libres (fichiers supprimés), lui permettant de les utiliser pour répartir uniformément les écritures, ce qui prolonge la durée de vie du support (wear-leveling).

3.7 Montage des partitions

- Le montage des partitions sous Linux est le processus qui rend un système de fichiers prêt à être utilisé par le système d'exploitation. Cela implique d'associer un système de fichiers situé sur une partition à un répertoire spécifique du système de fichiers global appelé point de montage (mountpoint).
- `$mount /dev/sdX2 /mnt` : On monte la partition racine dans le système de fichiers temporaire fourni par l'environnement d'installation.
- `$mkdir /mnt/boot` : Création du répertoire /mnt/boot pour accueillir la partition de démarrage.
- `$mount /dev/sdX1 /mnt/boot` : On monte la partition de démarrage dans le répertoire boot du système de fichiers racine.
- Ces opérations intègrent ainsi l'arborescence de fichier de /dev/sdX1 et de /dev/sdX2 dans celle du support d'installation, ce qui permet d'accéder à la structure complète du futur système.

3.8 Fichier fstab

- `$mkdir /mnt/etc` : Création du répertoire des configurations du système.
- `$genfstab -t PARTUUID /mnt > /mnt/etc/fstab` : Scanne les partitions montées et crée un fichier /mnt/etc/fstab en utilisant les **PARTUUID** pour identifier de manière unique chaque partition.
- Le fichier /mnt/etc/fstab spécifie les partitions à monter, leurs points de montage, les types de systèmes de fichiers, les options de montage, et les paramètres de dump et de vérification.
- Le fichier /mnt/etc/fstab est utilisé pour indiquer au système quelles partitions monter et comment procéder.
- **Pourquoi PARTUUID ?**
 - Les noms de périphériques comme /dev/sda, /dev/sdb peuvent changer d'un démarrage à l'autre, surtout si des supports amovibles sont connectés.
 - Les **PARTUUID** sont des identifiants uniques et persistants assignés à chaque partition, garantissant un montage fiable quel que soit l'ordre de détection des périphériques.

4. Installation

4.1 Paquets

- L'une des plus grandes forces de Linux réside dans son système de paquets. Chaque logiciel est distribué sous forme de paquet avec une signature cryptographique qui garantit son authenticité et son intégrité.
- Grâce à la vérification cryptographique des paquets, il n'est pas nécessaire de faire confiance au serveur qui distribue les logiciels (miroir) lui-même, puisqu'un paquet non-signé ou altéré sera rejeté automatiquement par le gestionnaire de paquets du système (pacman).
- N'importe qui peut héberger un miroir des dépôts Arch Linux, c'est-à-dire un serveur qui contient une copie de tous les paquets officiels créés par les développeurs d'Arch Linux.

4.2 Optimisation des miroirs

- L'écosystème des miroirs d'Arch Linux comprend plus de 800 serveurs répartis à travers le monde. Parmi ceux-ci, certains serveurs sont géographiquement plus proches, offrant ainsi une latence réduite ainsi qu'une bande passante souvent supérieure.
- Avant d'installer le système, il est donc préférable d'optimiser la liste des miroirs pour télécharger les paquets plus rapidement.
- `$pacman --noconfirm -Syu reflector` : Installe et met à jour la base de données des paquets, puis installe `reflector` un outil d'optimisation des miroirs.
- `$reflector --verbose --sort rate --country Canada --protocol http,https --save /etc/pacman.d/mirrorlist` : Met à jour la liste des miroirs en utilisant ceux au Canada et trié du plus rapide au plus lent.

4.3 Optimisation de pacman

- Pour améliorer les performances de pacman, éditez le fichier de configuration `/etc/pacman.conf`. Décommentez la ligne `ParallelDownloads` puis définissez sa valeur à **32** (ou un autre nombre selon vos préférences).

```
#ParallelDownloads = 5
ParallelDownloads = 32
```

- Cette modification permet d'effectuer plusieurs téléchargements en parallèle, réduisant significativement le temps total de mise à jour ou d'installation des paquets.

4.4 Installation du système

- `$pacstrap /mnt base base-devel linux linux-headers linux-firmware xorg-drivers nvidia-open intel-ucode amd-ucode zram-generator git` ← Ajouter un éditeur de texte (ex. nano, vim).
 - **base** : Paquets essentiels pour un système Arch minimal.
 - **base-devel** : Ensemble d'outils de développement pour la compilation de logiciels.
 - **linux et linux-headers** : Noyau Linux et ses en-têtes nécessaires pour le développement de modules.
 - **linux-firmware** : Collection de firmwares pour divers matériels pour assurer un bon support matériel.
 - **xorg-drivers et nvidia-open** : Pilotes graphiques (Intel, AMD, Nvidia) pour assurer un bon support matériel.
 - **intel-ucode et amd-ucode** : Microcodes pour les processeurs Intel et AMD, permettant l'installation des mises à jour critiques du microprogramme.
 - **zram-generator** : Utilitaire permettant de configurer zram pour la compression de la mémoire vive.
 - **git** : Système de gestion de versions distribué qui sera utile pour installer des paquets plus tard.

4.5 Transfère des optimisations

- `$cp /etc/pacman.conf /mnt/etc/pacman.conf` : Copie la configuration de pacman dans le nouveau système, préservant les optimisations précédemment définies.
- `$cp /etc/pacman.d/mirrorlist /mnt/etc/pacman.d/mirrorlist` : Transfère la liste des miroirs optimisée dans le nouveau système, afin de conserver la liste de miroirs optimisés.

5. Chroot

5.1 Chroot

- `$arch-chroot /mnt`
- La commande `arch-chroot` permet de « rentrer » dans l'environnement du nouveau système installé sur le point de montage `/mnt`.
- Elle modifie la racine du système de fichiers, faisant de `/mnt` le nouveau `/`.
- Une fois à l'intérieur, vous pouvez gérer votre système comme s'il était démarré, par exemple installer des paquets, configurer des services ou éditer des fichiers de configuration, sans avoir à démarrer immédiatement le nouveau système.

5.2 Étapes du processus de démarrage

1. **Firmware** : Le firmware (UEFI) initialise le matériel minimal, localise la partition système EFI (ESP) et charge le bootloader se trouvant dessus.
2. **Bootloader** : Le bootloader (systemd-boot) sélectionne le noyau et charge également l'initramfs en mémoire, puis passe la main au noyau en lui fournissant les paramètres nécessaires.
3. **Initramfs** : Chargé avant l'exécution complète du noyau, l'initramfs sert de système de fichiers temporaire. Il contient les pilotes et outils indispensables pour trouver et monter la véritable racine du système.
4. **Noyau Linux** : Utilisant l'initramfs, le noyau s'initialise, configure les périphériques, monte le système de fichiers racine, et lance le processus init.
5. **Processus init** : Le premier processus est systemd (PID 1). Il lance les services, monte les partitions complémentaires, configure le réseau, met en place et lance l'environnement utilisateur final.

6. Bootloader

6.1 Bootloader

- `$bootctl install` : Installe le chargeur de démarrage (*bootloader*) systemd-boot sur la partition EFI.
- Créer le fichier `/boot/loader/loader.conf` avec le contenu suivant :

```
default arch.conf
timeout 0
```

- Cette configuration définit le fichier `arch.conf` comme l'entrée par défaut et désactive le délai d'attente (`timeout 0`), ce qui lance immédiatement Arch Linux au démarrage.

6.2 Bootloader

- `$blkid -s PARTUUID -o value /dev/sdX2 > /boot/loader/entries/arch.conf` : Insère la PARTUUID de la partition racine dans `arch.conf` pour éviter de devoir la retaper manuellement.
- Éditer le fichier `/boot/loader/entries/arch.conf` avec le contenu suivant :

```
title Arch Linux
linux /vmlinuz-linux
initrd /intel-ucode.img
initrd /amd-ucode.img
initrd /initramfs-linux-fallback.img
options root=PARTUUID=<PARTUUID> rw
```

6.3 Bootloader

- **title Arch Linux** : Définit le nom qui apparaîtra dans le menu de démarrage.
- **linux /vmlinuz-linux** : Indique le chemin vers le noyau Linux.
- **initrd /intel-ucode.img** : Charge le microcode des processeurs Intel.
- **initrd /amd-ucode.img** : Charge le microcode des processeurs AMD. Ces fichiers ne sont appliqués que si le matériel correspondant est utilisé.
- **initrd /initramfs-linux-fallback.img** : Charge une image de système de fichier minimal en mémoire (initramfs) qui fournit les pilotes essentiels au noyau. La version `fallback` contient tous les pilotes disponibles, contrairement à la version standard qui est générée selon la configuration matérielle de l'ordinateur où a été réalisée l'installation.
- **options root=PARTUUID=<PARTUUID> rw** : Spécifie les options de démarrage du noyau, en particulier la partition racine (`root`) et l'option `rw` pour un accès en lecture-écriture.

7. Initramfs

7 Initramfs

- Pour permettre à l'initramfs de monter une partition racine se trouvant sur une clé USB, éditez le fichier de configuration `/etc/mkinitcpio.conf` comme suit :

```
MODULES=(usbhid xhci_hcd)
```

- Ce sont les modules nécessaires pour s'assurer que les pilotes USB requis sont chargés pendant le processus d'initialisation.
- `$mkinitcpio -P` : Reconstitue les images initramfs en appliquant les modifications.

8. Compression de la mémoire

8 Compression de la mémoire

- Pour activer la compression de la mémoire avec Zram, éditez le fichier de configuration `/etc/systemd/zram-generator.conf` comme suit :

```
[zram0]
zram-size = ram / 2
compression-algorithm = zstd
```

- Ces paramètres réservent la moitié de la RAM physique pour les pages compressées en mémoire, augmentant la capacité de mémoire disponible sans recourir à la pagination sur le disque (*swap*). L'algorithme `zstd` offre un bon équilibre entre vitesse et taux de compression.

9. Configurations importantes

9.1 Horloge et fuseau horaire

- `$ln -sf /usr/share/zoneinfo/Canada/Atlantic /etc/localtime` : Configure le fuseau horaire du Canada atlantique.
- `$systemctl enable systemd-timesyncd` : Active le service de synchronisation périodique de l'horloge fourni par `systemd` pour qu'il démarre automatiquement au démarrage.

9.2 Langue du système

- Pour configurer la langue du système à l'anglais canadien, modifiez le fichier de configuration `/etc/locale.gen` comme suit :

```
#en_CA.UTF-8 UTF-8 (Décommentez cette ligne)
```

- `$locale-gen` : Génère les locales sélectionnées pour le système.
- Ensuite, configurez la locale système par défaut dans le fichier `/etc/locale.conf` :

```
LANG=en_CA.UTF-8
```

- Ces paramètres assurent que le système utilise l'anglais canadien comme langue par défaut. Vous pouvez configurer la locale de votre choix (par ex. `fr_CA.UTF-8`).

9.3 Fichier hosts et hostname

- Pour configurer le nom d'hôte (nom de la machine), éditez le fichier `/etc/hostname` et insérez le nom d'hôte de votre choix :

```
<nom_hote_choisi>
```

- Ensuite, mettez à jour `/etc/hosts`. Remplacez `<nom_hote_choisi>` par votre nom d'hôte choisi :

```
127.0.0.1    localhost
127.0.1.1    <nom_hote_choisi>
::1         <nom_hote_choisi>.localdomain <nom_hote_choisi>
```

- Ces entrées permettent de mapper le nom d'hôte à l'adresse locale, facilitant ainsi la résolution locale des noms de domaines.

10. Comptes utilisateurs

10 Comptes utilisateurs

- `$useradd -m -g users <nom_utilisateur>` : Crée un nouveau compte utilisateur. Remplacez `<nom_utilisateur>` par le nom d'utilisateur de votre choix.
- `$usermod -G wheel <nom_utilisateur>` : Ajoute le nouveau compte utilisateur au groupe `wheel` qui permettra à l'utilisateur d'utiliser la commande `sudo`.
- Pour permettre aux membres du groupe `wheel` d'utiliser `sudo`, éditez `/etc/sudoers` comme suit :

```
# %wheel ALL=(ALL) ALL (Décommentez cette ligne)
```

- `$passwd` : Crée le mot de passe du compte `root`.
- `$passwd <nom_utilisateur>` : Crée le mot de passe de l'utilisateur.

11. Arch User Repository (AUR)

11 Arch User Repository (AUR)

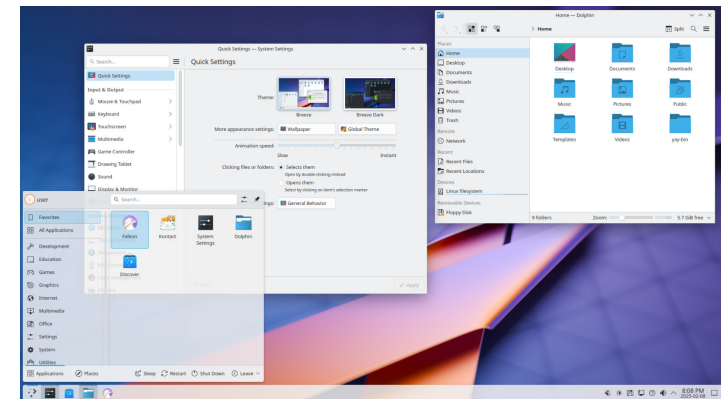
- L'Arch User Repository (AUR) est un dépôt communautaire pour Arch Linux qui permet le partage et l'accès à des paquets créés par les utilisateurs, étendant l'offre logicielle au-delà des dépôts officiels.
- Cela fait d'Arch Linux la distribution en existence avec le plus grand nombre de paquets disponibles, et ce, de manière très accessible par rapport à d'autres distributions (par ex. PPA sur Ubuntu).
- `$su <nom_utilisateur>` : Entrer dans le compte utilisateur non privilégié.
- `$cd` : Naviguer vers le répertoire home de l'utilisateur pour avoir l'accès en écriture.
- `$git clone https://aur.archlinux.org/yay-bin.git` : Récupérer les fichiers nécessaires pour construire Yay, un gestionnaire de paquets pour l'AUR.
- `$cd yay-bin` : Naviguer dans le répertoire contenant les fichiers.
- `$makepkg -sri` : Construire et installer le paquet de Yay.
- `Ctrl + D` ou `$exit` : Sortir du compte utilisateur non privilégié.

12. Installation d'un bureau graphique

12.1 Terminologie

- **Display Manager (Gestionnaire d'affichage)** : C'est le logiciel qui gère l'écran de connexion et l'authentification des utilisateurs, lançant ensuite l'environnement de bureau choisi. Exemples : GDM, LightDM.
- **Window Manager (Gestionnaire de fenêtres)** : Il gère le positionnement et l'apparence des fenêtres sur l'écran, permettant des actions comme minimiser ou redimensionner. Exemples : KWin, Mutter.
- **Desktop Environment (Environnement de bureau)** : C'est un ensemble intégré d'applications et d'outils qui fournit une interface utilisateur graphique complète, comprenant un gestionnaire de fenêtres et des applications de base. Exemples : KDE Plasma, GNOME.

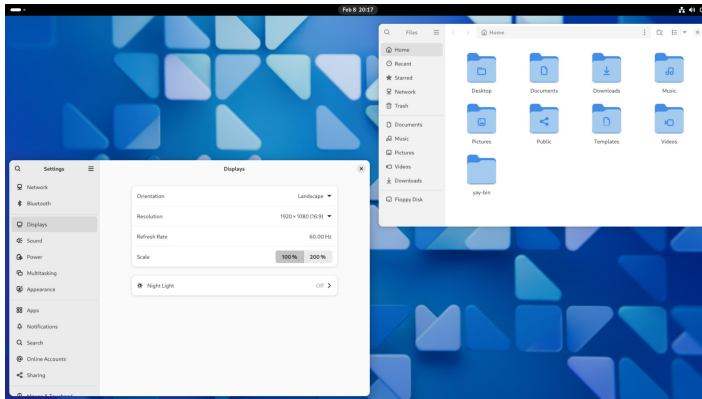
12.2 KDE Plasma



- `$pacman -S networkmanager plasma kde-applications sddm`
- `$systemctl enable NetworkManager sddm`

12.3

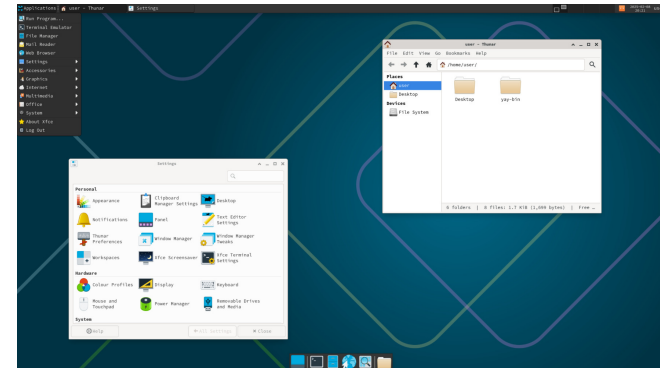
Gnome



- `$pacman -S networkmanager gnome gnome-extra`
- `$systemctl enable NetworkManager gdm`

12.4

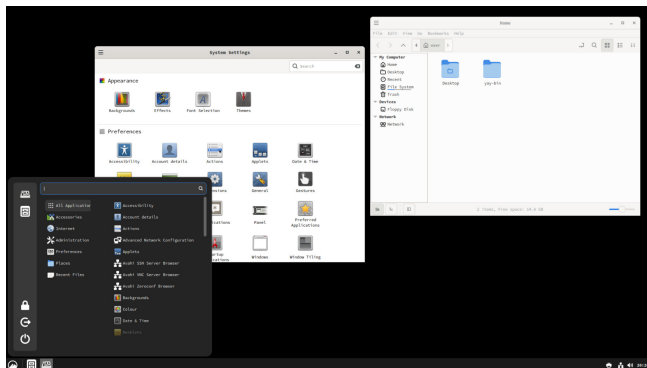
Xfce



- `$pacman -S networkmanager network-manager-applet xfce4 xfce4-goodies lightdm lightdm-gtk-greeter`
- `$systemctl enable NetworkManager lightdm`

12.5

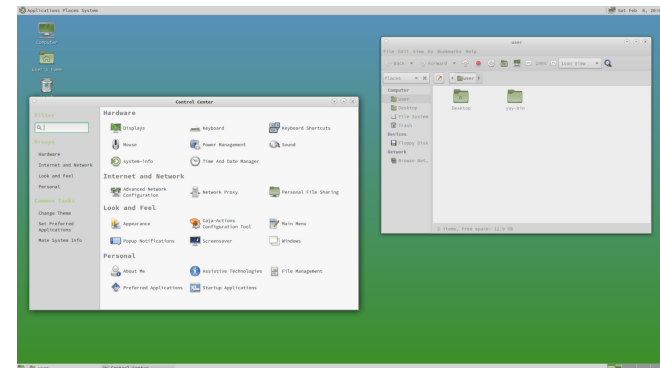
Cinnamon



- `$pacman -S networkmanager cinnamon lightdm lightdm-gtk-greeter`
- `$systemctl enable NetworkManager lightdm`

12.6

MATE



- `$pacman -S networkmanager network-manager-applet mate mate-extra lightdm lightdm-gtk-greeter`
- `$systemctl enable NetworkManager lightdm`

13. Redémarrage

13 Redémarrage

- `Ctrl + D` ou `$exit` : Sortir du chroot.
- `$umount -R /mnt` : Démonter les partitions `/mnt/boot` et `/mnt`.
- `$reboot` : Redémarrer le système.

14. Extras

14.1 Mise à jour du système

- Pour synchroniser la base de données des paquets et mettre à jour l'ensemble du système à partir des dépôts officiels, exécutez :
 - `$sudo pacman -Syu`
- Si vous avez installé des paquets du AUR, vous pouvez mettre à jour tous les paquets (officiels et AUR) avec :
 - `$sudo yay -Syu`
- Si le paquet `linux` a été mis à jour, il est recommandé de redémarrer le système, mais cela est optionnel.
- Ces commandes garantissent que vous disposez des dernières versions, bénéficiant ainsi des correctifs de sécurité et des améliorations récentes.
- Il est recommandé de réaliser cette opération régulièrement afin de maintenir un système stable, sécurisé et performant.

14.2 Configuration d'un serveur SSH

- Installez le paquet OpenSSH afin d'installer le service SSH :
 - `$sudo pacman -S openssh`
- Activez et démarrez le service SSH pour qu'il se lance automatiquement au démarrage :
 - `$systemctl enable sshd`
 - `$systemctl start sshd`
- Vérifiez que le service fonctionne correctement avec :
 - `$systemctl status sshd`
- Par la suite, pour se connecter en SSH **depuis une autre machine**, utilisez :
 - `$ssh nom_utilisateur@adresse_ip`

Eric Hivet et Andy Couturier

Université de Moncton, 20 mars 2025

