

Installation Arch Linux

- [Installation OS](#)
- [Installation phpmyadmin](#)
- [Suite installation OS](#)
- [Macbook a1706 touchbar](#)

Installation OS

1. Commencé l'installation

<https://www.arcolinuxd.com/5-the-actual-installation-of-arch-linux-phase-1-uefi/>

1. les commande de base a effectuer

#passé le clavé en azerty

```
loadkeys fr-latin1
```

2. ce connecté a la wifi

#on lance l'interface de wifi

```
iwctl
```

#on sort la list des systeme wifi

```
station list
```

#on scan les wifi

```
station wlan0 scan
```

#on se connect

```
station wlan0 connect SFR-0a00
```

#on rentre le mdp

#on regarde les info de connection

```
station wlan0 show
```

3. **si on veut se connecter en ssh**

#sur la session root@archiso on mes un mdp avec la command suivant

passwd

#après s'être connecté a internet la command pour voir son ip

ip addr

#on peut lancer la command ssh avec l'ip trouver avec station wlan0 show depuis un autre ordi

ssh root@mon_ip

#en cas d'erreur effectuer ces command sur votre pc

ssh-keygen -R adresse-ip-du-serveur

4. **On partitionne les disque**

en 3 partir principale

-/dev/sda1 1G Linux - EFI system partition

-/dev/sda2 8G Linux Swap

-/dev/sda3 le reste Linux

5. **début de l'installation**

#on active la synchronisation de l'horloge

timedatectl set-ntp true

#après avoir partitionné le disque avec cfdisk

#on formate les partition

mkfs.fat -F32 /dev/sda1

mkswap /dev/sda2

swapon /dev/sda2

mkfs.ext4 /dev/sda3

ou

mkfs.fat -F32 /dev/nvme1n1p1

mkswap /dev/nvme1n1p2

```
swapon /dev/nvme1n1p2  
mkfs.ext4 /dev/nvme1n1p3
```

#on monte le disque

```
mount /dev/sda3 /mnt  
mkdir -p /mnt/boot/efi  
mount /dev/sda1 /mnt/boot/efi
```

ou

```
mount /dev/nvme1n1p3 /mnt  
mkdir -p /mnt/boot/efi  
mount /dev/nvme1n1p1 /mnt/boot/efi
```

#mise a jour des repository

```
reflector --country France --age 12 --protocol https --sort rate --save /etc/pacman.d/mirrorlist
```

#vérifier les repository

```
nano /etc/pacman.d/mirrorlist
```

#on installe la base du linux dans /mnt

```
pacstrap -K /mnt base base-devel linux linux-firmware nano
```

#encas d'erreur utilisé la commande

```
pacman -Sy archlinux-keyring
```

#configuration du système

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

#vérifier que tout est bien installé

```
nano /mnt/etc/fstab
```

6. **on rentre dans notre nouvelle machine**

#on passe sur le nouveau systeme root

```
arch-chroot /mnt
```

n défini le fuseau horraire a France

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

#pour adapter l'horologe matérielle

```
hwclock --systohc
```

#on choisi le langage du systeme en décommentant la ligne voulu ici ces fr_FR.UTF-8.

```
nano /etc/locale.gen
```

#on mes a jour le changement avec

```
locale-gen
```

#on crée le fichier pour garder la langue a jour

```
nano /etc/locale.conf
```

#on rentre dedans --> LANG=fr_FR.UTF-8

#on crée le fichier vconsole pour garder en mémoire le type de clavier utilisé

```
nano /etc/vconsole.conf
```

#on rentre dedans --> KEYMAP=fr-latin1

#on nomme notre machine en modifiant ce qui a dans le dossier

```
nano /etc/hostname
```

#on peut modifier les host localhost ect avec

```
nano /etc/hosts
```

#rentré dedans

```
127.0.0.1 localhost
```

```
::1 localhost
```

```
127.0.0.1 Archlinux.localdomain Archlinux
```

#on install le networkmanager

```
pacman -S tree networkmanager
```

#et on peut l'activer avec

```
systemctl enable NetworkManager
```

#on modifier le mot de passe root

passwd

#on telecharge les paquet grub et ces efiboot

pacman -S grub efibootmgr

#pour commencer l'installation

grub-install --target=x86_64-efi --efi-directory=/boot/efi

#on le configure avec

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

#création du noyaux linux

mkinitcpio -p linux

7. on quitte l'installation

#enfin on peut quitter avec les command

exit

#suivi

umount -R /mnt

#suivi

reboot

8. script archiso

crée le fichier script avec

cd /home && nano script_installation1 && chmod +x script_installation1

copier les ligne suivante dans le fichier et enregistré avec ctrl + s et ctrl +x pour quitté

#!/bin/bash

echo "début"

#on active la synchronisation de l'horloge

timedatectl set-ntp true

```
#après avoir partitionné le disque avec cfdisk
#on formate les partition
mkfs.fat -F32 /dev/sda1
mkswap /dev/sda2
swapon /dev/sda2
mkfs.ext4 /dev/sda3
#on monte le disque
mount /dev/sda3 /mnt
mkdir /mnt/boot
mkdir /mnt/boot/efi
mount /dev/sda1 /mnt/boot/efi
#mise a jour des repository
reflector --country France --age 12 --protocol https --sort rate --save /etc/pacman.d/mirrorlist
#encas d'erreur utilisé la commande
pacman -Sy archlinux-keyring
#on installe la base du linux dans /mnt
pacstrap -K /mnt base linux linux-firmware nano
```

lancé le script avec

```
./script_installation1
```

2. **Boot sur la nouvelle machine**

1. **utilisé l'outils networkmanager pour ce connecter a internet**

entré la commande "**nmtui**" pour lancé l'interface de connection a internet

2. **ajouter les droit sudo a un utilisateur**

```
#créer un nouvel utilisateur
```

```
useradd USER_NOM
```

```
#changer son mdp
```

```
passwd USER_NOM
```

```
#Créer le dossier utilisateur 'home'
```

```
mkdir /home/USER_NOM
```

```
#Appliquer les autorisations pour l'utilisateur dans ce dossier
```

```
chown -R USER_NOM:users /home/USER_NOM
```

```
#vérifier si l'utilisateur a le droit d'utiliser la commande sudo *
```

```
sudo -IU USER_NAME
```

```
#l'ajouter à la liste des sudoers
```

```
usermod -aG wheel USER_Name
```

```
#ensuite éditer le fichier suivant
```

```
nano /etc/sudoers
```

```
#aller à la ligne user privilège specification contenant la ligne suivante
```

```
sudo ALL=(ALL) ALL
```

```
#ajouté
```

```
USER_NAME ALL=(ALL) ALL
```

3. installé et configuré des paquet utile

```
#créer un nouvel utilisateur sinon on pourra pas se connecter à distance sur la machine
```

```
#installé ssh
```

```
pacman -S openssh
```

```
#entrer la commande suivante pour activer le ssh au démarrage
```

```
systemctl enable sshd
```

```
#installé des paquet utile
```

```
sudo pacman -S sudo unzip git maven gimp thunderbird firefox ufw htop
```

4. installé la command yay

#ce placé dans /opt est installer le github yay

```
cd /opt && sudo git clone https://aur.archlinux.org/yay-git.git
```

#donné les droit au utilisateur

```
sudo chown -R USER_NAME:users ./yay-git
```

#entré dans son dossier et installé le logiciel

```
cd yay-git && makepkg -si
```

#le mettre a jour

```
sudo yay -Syu
```

5. installer et configurer une interface graphique

#installé les paquet nécessaire a crée une interface graphique

```
sudo pacman -S lxde lxdm xorg-xinit
```

```
sudo pacman -Syu xorg xf86-video-intel nvidia-utils
```

```
sudo pacman -Syu xorg xorg-server fluxbox xf86-video-intel nvidia nvidia-utils
```

Pour cinnamon

```
pacman -S cinnamon lightdm lightdm-gtk-greeter xorg
```

#minimaliste

```
sudo pacman -S xorg-server lightdm xfce4
```

#lancé au démarrage

```
systemctl enable lightdm
```

#lancé au démarrage lxde avec la commande

sudo systemctl enable lxdm

#copier le fichier xinitrc pour notre utilisateur

```
sudo cp /etc/X11/xinit/xinitrc ~/.xinitrc
```

#activé l'exécution du fichier .xinitrc de votre nom_user il permettra de lancé interface graphique de votre choix

sudo chmod +x ~/.xinitrc

#on rajoute dans ce même fichier la commande "exec startlxde"

```
sudo nano ~/.xinitrc
```

3. **Erreur rencontré**

1. **on boot "Fail to load Kernel modules "**

#Démarrage lent avec une erreur "fail to load kernel module"

#j'ai désinstaller linux et nvidia qui est dépendant de linux

```
sudo pacman -R nvidia linux
```

#j'ai boot sur arch-chroot avec une cle usb d'installation pour réinstaller linux

Installation phpmyadmin

<https://www.youtube.com/watch?v=ZpazlwFMqY8>

```
sudo pacman -S mariadb apache php php-apache  
sudo systemctl enable httpd -now  
sudo nano /etc/httpd/conf/httpd.conf
```

#les deux ligne suivante doit etre commenté et décommenté pour donné ceci :

```
“  
#LoadModule mpm_event_module modules/mod_mpm_event.so  
LoadModule mpm_prefork_module modules/mod_mpm_prefork.so  
“
```

a la fin du fichier rajouter

```
“  
LoadModule php_module modules/libphp.so  
AddHandler php-script .php  
# Configuration de phpMyAdmin  
Include conf/extra/phpmyadmin.conf  
“
```

```
sudo systemctl restart httpd  
sudo systemctl enable mariadb -now  
sudo mariadb-install-db --user=mysql --basedir=/usr --datadir=/var/lib/mysql
```

```
sudo mysql -u root
```

```
>use mysql  
>flush privileges;  
>ALTER USER 'root'@'localhost' IDENTIFIED BY 'new_password';
```

ou

```
sudo mysql -u root -p
```

```
>CREATE USER 'flo'@'localhost' IDENTIFIED BY 'flo';  
>GRANT ALL PRIVILEGES ON mydb.* TO 'flo'@'localhost';  
>FLUSH PRIVILEGES;
```

>quit

mysql_secure_installation

sudo pacman -S phpmyadmin

#configuré apache

sudo nano /etc/httpd/conf/extra/phpmyadmin.conf

#entrer dedans

Alias /phpmyadmin "/usr/share/webapps/phpMyAdmin"

<Directory "/usr/share/webapps/phpMyAdmin">

DirectoryIndex index.php

AllowOverride All

Options FollowSymLinks

Require all granted

</Directory>

sudo nano /etc/php/php.ini

#enlever le ";" de la ligne extension=mysql

sudo systemctl restart httpd

Suite installation OS

Suite de l'installation OS Archlinux

Configuration Computer

processeur : i5

RAM : x4 4Go --> 16Go DDR3

GPU : 4070 RTX

user : bat

Installation YAY

```
pacman -S archlinux-keyring
```

#ce placé dans /opt est installer le github yay

```
cd /opt && sudo git clone https://aur.archlinux.org/yay-git.git
```

#donné les droit au utilisateur

```
sudo chown -R USER_name:users ./yay-git
```

#entré dans son dossier et installé le logiciel

```
cd yay-git && makepkg -si
```

#le mettre à jour

```
yay -Syu
```

Configuration Ssh

```
cd
mkdir .ssh
cd .ssh
mkdir myKeys
touch config
touch authorized_keys
```

configuré interface a distance

```
sudo nano /etc/ssh/sshd_config
```

Assurez-vous que les lignes suivantes sont présentes et décommentées

```
X11Forwarding yes
X11DisplayOffset 10
X11UseLocalhost yes
```

```
sudo systemctl restart sshd
```

Installation ZSH

#téléchargement zsh

```
sudo pacman -S zsh
```

changement de bash par default

```
chsh -s /bin/zsh
```

installation zsh

```
sh -c "$(curl -fsSL
https://raw.githubusercontent.com/ohmyzsh/ohmyzsh/master/tools/install.sh)"
```

recharger zsh

```
source ~/.zshrc
```

Thème zsh --> <https://github.com/ChesterYue/ohmyzsh-theme-passion>

1. clone repo: `git clone https://github.com/ChesterYue/ohmyzsh-theme-passion`;
2. copy theme: `cp ../ohmyzsh-theme-passion/passion.zsh-theme ~/.oh-my-zsh/themes/passion.zsh-theme`;
3. modify rc: open `~/.zshrc` find `ZSH_THEME` edit to `ZSH_THEME="passion"`;
4. execute rc: `source ~/.zshrc`

```
sudo pacman -S bc
```

Installation Nvidia

```
sudo pacman -S nvidia nvidia-utils nvidia-settings  
sudo reboot
```

```
#sudo pacman -S dkms nvidia-dkms
```

Installation Environnement graphique

```
#installer les paquets nécessaire à la création d'une interface graphique
```

```
sudo pacman -S lxde lxdm xorg-xinit  
sudo pacman -S xorg xorg-server xf86-video-intel
```

```
#lancer au démarrage lxde avec la commande
```

```
sudo systemctl enable lxdm
```

#copier le fichier xinitrc pour notre utilisateur

```
sudo cp /etc/X11/xinit/xinitrc ~/.xinitrc
```

#Activé l'exécution du fichier .xinitrc de votre nom_user il permettra de lancé interface graphique de votre choix

```
sudo chmod +x ~/.xinitrc
```

#On rajoute dans ce même fichier la commande "exec startlxde"

```
sudo nano ~/.xinitrc
```

Installation Cinnamon

```
sudo pacman -S cinnamon
```

Choisir Cinnamon dans l'interface de login

thème Cinnamon

```
yay -S orchis-theme  
yay -S whitesur-gtk-theme  
yay -S cinnamon-big-sur-theme
```

thème icone

```
yay -S papirus-icon-theme
```

Installation Python + Stable Diffusion

voir page : <https://wiki.mira-ceti.ovh/books/bot/page/installation-python-controler-generator-prompt>

Installation Autre packet nécessaire

```
sudo pacman -S discord scrcpy wget android-udev
```

```
yay -S android-studio-canary slack-desktop adb
```

```
#packet de surveillance consommation
```

```
sudo pacman -S glances htop nvtop sysstat bmon gkrellm atop conky
```

```
sudo pacman -S rsync
```

```
sudo pacman -S firefox chromium
```

```
sudo pacman -S geany code
```

```
#packet autre
```

```
sudo pacman -S gimp vlc gnome-terminal audacious xfburn firefox unzip unrar chromium gedit  
pidgin vim thunderbird libreoffice pipewire pipewire-alsa deluge -y
```

Macbook a1706 touchbar

<https://bbs.archlinux.org/viewtopic.php?id=309376>

changer kernel linux version to 6.12.51

```
uname -r
```

```
yay -S downgrade
```

utiliser les commande suivante pour prendre un kernel compatible inférieur a 6.12.51 les linux doit etre a la meme version

```
sudo downgrade linux-lts  
sudo downgrade linux-lts-headers
```

```
sudo mkinitcpio -P
```

```
sudo reboot
```

```
git clone
```

```
cd
```

```
make
```

```
sudo mkdir -p /usr/lib/modules/$(uname -r)/updates/dkms/
```

```
sudo cp *.ko /usr/lib/modules/$(uname -r)/updates/dkms/
```

```
sudo depmod -a
```

```
sudo nano /etc/mkinitcpio.conf
```

```
replace  
MODULES=()
```

```
with  
MODULES=(applespi intel_lpss_pci spi_pxa2xx_platform apple_ib_tb)
```

```
sudo modprobe applespi  
sudo modprobe apple_ib_tb
```

```
sudo mkinitcpio -P
```

```
sudo mkdir -p /usr/src/applespi-0.1  
sudo cp * /usr/src/applespi-0.1/
```

```
sudo dkms add -m applespi -v 0.1  
sudo dkms build -m applespi -v 0.1  
sudo dkms install -m applespi -v 0.1
```

```
ls /lib/modules/$(uname -r)/updates/dkms/
```

```
sudo modprobe applespi  
sudo modprobe apple_ib_tb  
lsmod | grep apple
```

```
pilote touch bar
```

```
yay -S macbook12-spi-driver-dkms
```

autre

pour yay -S mac-touchbar-plus

sudo pacman -S rustup

rustup install stable

rustup default stable

rustup toolchain install nightly

RUSTUP_TOOLCHAIN=nightly yay -S mac-touchbar-plus