

NOTICE

Installation du dual boot

Sommaire :

Partie 1 : Partitionnement du disque	p.2
Partie 2 : Installation de Windows 7	p.6
Partie 3 : Installation de Archlinux	p.13
Partie 4 : Installation du dual boot	p.16

- Le lecteur CD ide0 contient l'image iso d'installation de la distribution Archlinux.
- Le lecteur CD ide1 contient les pilotes pour le support virtio.
- Le lecteur CD ide2 contient l'image iso d'installation de Windows7.

Partitionnement du disque

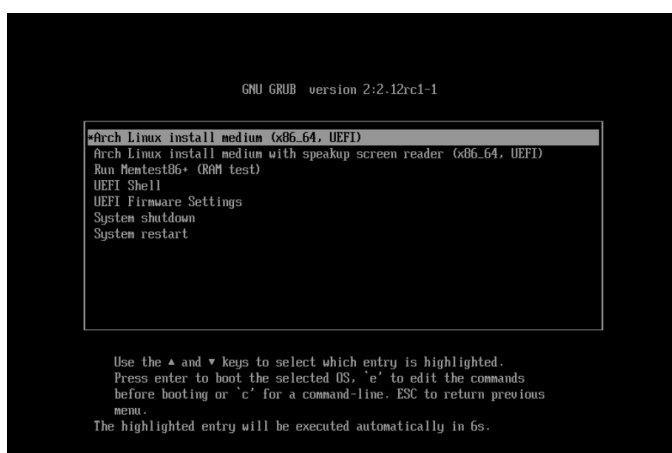
Pour commencer, il faut accéder au BIOS. Pour ce faire, il faut appuyer sur **Echap**. Pour se déplacer dans le BIOS, il faut utiliser les **flèches** du clavier. Pour confirmer un choix, il faut appuyer sur **Entrée**.



Sélectionner le disque contenant l'image iso d'installation de Archlinux. Ici, c'est **UEFI QEMU DVD-ROM QM00001**



Une fois arrivé sur la fenêtre GNU GRUB, on appuie sur **Entrée** pour accéder à Linux.



Il faut patienter un peu le temps que Linux charge. Une fois terminé, on atterrit sur un terminal de commande.

```
Arch Linux 6.4.12-arch1-1 (tty1)
archiso login: root (automatic login)

To install Arch Linux follow the installation guide:
https://wiki.archlinux.org/title/Installation_guide

For Wi-Fi, authenticate to the wireless network using the iwctl utility.
For mobile broadband (WWAN) modems, connect with the mmcli utility.
Ethernet, WLAN and WWAN interfaces using DHCP should work automatically.

After connecting to the internet, the installation guide can be accessed
via the convenience script installation_guide.

root@archiso ~ #
```

Pour être à l'aise quand on va taper les commandes, il faut mettre le clavier en azerty. Pour ce faire, il faut taper la commande **loadkeys fr**.

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

On entre la commande **fdisk -l** pour voir tous les disques sur le pc ainsi que leurs noms (ce qui va nous être utile pour la suite). Ici, c'est **/dev/sda** qui nous intéresse.

```
127 root@archiso ~ # fdisk -l
Disk /dev/sda: 40 GiB, 42949672960 bytes, 83886080 sectors
Disk model: QEMU HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: 739A1226-FBC4-4592-AA5E-9C1A29616B78

Disk /dev/loop0: 682.6 MiB, 715755520 bytes, 1397960 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
root@archiso ~ #
```

On tape la commande **cgdisk /dev/sda** pour voir les partitions présentes sur le disque sda ainsi que les gérer.

```
root@archiso ~ # cgdisk /dev/sda
```

Une fois la commande tapée, on tombe sur le menu ci- dessous. Comme avec le BIOS, il faut se déplacer avec les **touches** du clavier. Pour créer une partition, il faut cliquer sur **New**.

```
cgdisk 1.0.9.1
Disk Drive: /dev/sda
Size: 83886080, 40.0 GiB

Part. #   Size   Partition Type   Partition Name
-----
          40.0 GiB   free space

[ Align ] [ Backup ] [ Help ] [ Load ] [ New ] [ Quit ] [ Verify ] [ Write ]

Create new partition from free space
```

Pour cette étape, on laissera la valeur par défaut (pour les 3 partitions).

```
First sector (2048-83886046, default = 2048): _
```

Comme on veut une partitions EFI, il faut donc une partition de 512Mo. Il faut donc écrire 512M.

```
Size in sectors or {KMGTP} (default = 83881984): 512M_
```

Comme on veut une partition EFI, on entre le code 0xEF00 pour indiquer qu'il s'agit d'une partition EFI (il faut appuyer sur Entrée pour savoir quel code choisir en fonction du type de partition).

```
Hex code or GUID (L to show codes, Enter = 8300): 0xEF00_
```

Pour cette étape, libre à nous de choisir un nom.

```
Enter new partition name, or <Enter> to use the current name:
EFI_
```

Une fois terminé, on voit notre partition créée. On recommencera la même étape pour la partition Linux et Linux Home en se plaçant sur le Freespace libre de 40Go.

```
cgdisk 1.0.9.1
Disk Drive: /dev/sda
Size: 83886080, 40.0 GiB

Part. #   Size   Partition Type   Partition Name
-----
2         1007.0 KiB   Linux filesystem
1         512.0 MiB    EFI system partition   EFI
          39.5 GiB   free space
```

- Hex code or GUILD : 83
- Partition name : vide

Linux Home :

- First sector : vide
- Size in sector : 5G
- Hex code or GUILD : 8302
- Partition name : vide

On peut voir que toutes nos partitions sont créées. Pour les valider et les enregistrer, il faut les écrire en se plaçant sur **write** comme dans l'image ci- dessous.

```

cgdisk 1.0.9.1
Disk Drive: /dev/sda
Size: 83886080, 40.0 GiB

Part. #   Size   Partition Type   Partition Name
-----
1         1007.0 MiB   free space
3         512.0 MiB     EFI system partition   EFI
2         10.0 GiB     Linux filesystem        Linux
2         5.0 GiB       Linux /home              Linux Home
2         24.5 GiB      free space

[ Align ] [ Backup ] [ Delete ] [ Help ] [ Info ] [ Load ] [ Name ] [ Quit ] [ Type ] [ Verify ]
[ Write ]

Write partition table to disk (this might destroy data)

```

Il faut valider pour chaque partitions, le fait que l'on veut les écrire. Pour ce faire, il faut taper **yes**.

```

Are you sure you want to write the partition table to disk? (yes or no): yes_

Warning!! This may destroy data on your disk!

```

Maintenant, on a plus qu'à quitter **cgdisk** en se plaçant sur **Quit** et vérifier qu'avec la commande **lsblk** que nos partitions sont bien créées.

```

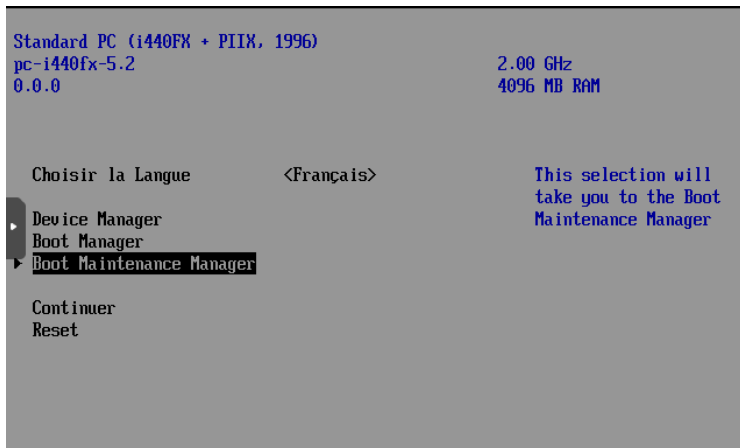
root@archiso ~ # lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0       7:0    0 682.6M 1 loop /run/archiso/airootfs
sda         8:0    0   40G  0 disk
├─sda1      8:1    0   512M  0 part
├─sda2      8:2    0    5G  0 part
└─sda3      8:3    0   10G  0 part
sr0         11:0    1 804.3M  0 rom  /run/archiso/bootmnt
sr1         11:1    1 375.7M  0 rom
sr2         11:2    1    2.9G  0 rom
root@archiso ~ #

```

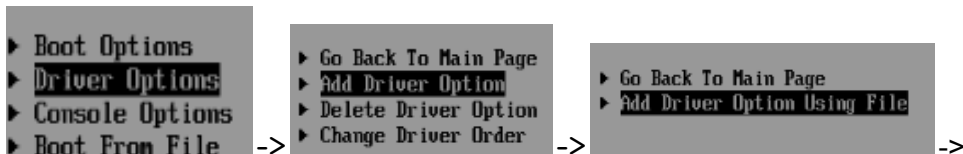
Installation de Windows 7

Dans cette partie, on va s'occuper d'installer Windows 7. Pour ce faire, il faut d'abord installer un support virtuel pour pouvoir lancer Windows.

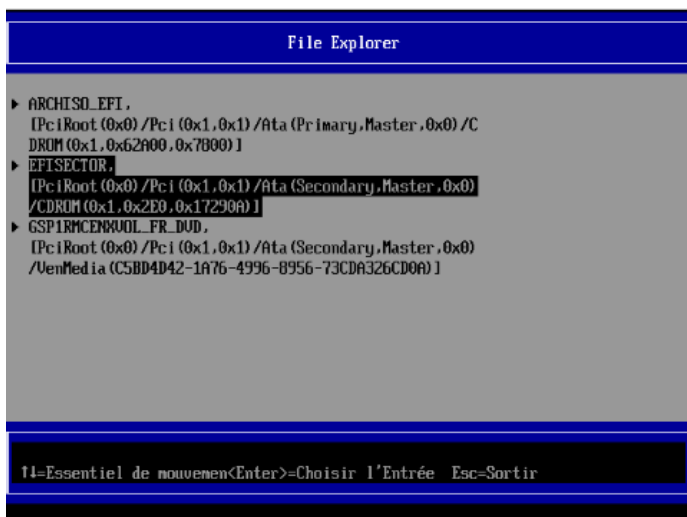
Il faut aller dans le BIOS. Il faut lancer la machine et appuyer sur Echap. Une fois dans le BIOS, il faut aller dans **Boot Maintenance Manager**.



Ensuite, il faut aller dans :



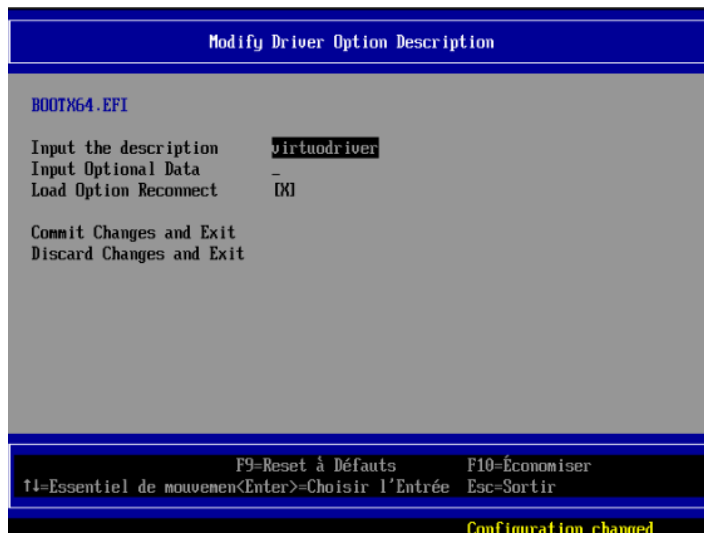
Ensuite, on choisit le cd qui contient le support virtuel qui est **EFI SECTOR** (voir photo ci-dessous).



Puis, on va dans :

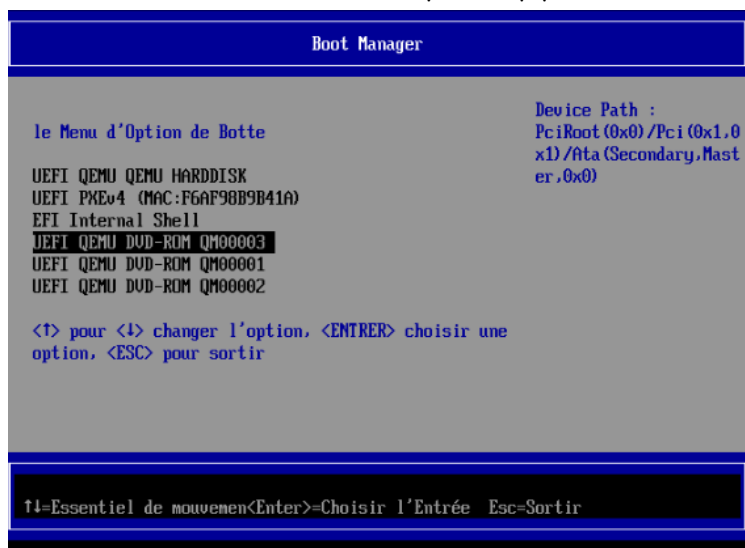


Une fois dans **Input The Description**, on choisit le nom qu'on veut et on confirme avec **Entrée**.



On sauvegarde les changements en allant dans **Commit Changes and Exit**.

On va dans **Boot Manager** et on lance le cd qui contient l'image iso d'installation Windows 7 qui s'appelle **UEFI QEMU DVD6ROM QM00003**

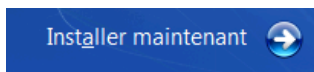


Normalement, il faudra appuyer sur **Entrée** pour pouvoir aller sur Windows. On attend quelques minutes le temps que ça charge.

Pour lancer Windows, laisser les paramètres par défaut (voir image ci-dessous). Cliquez sur **suivant**.

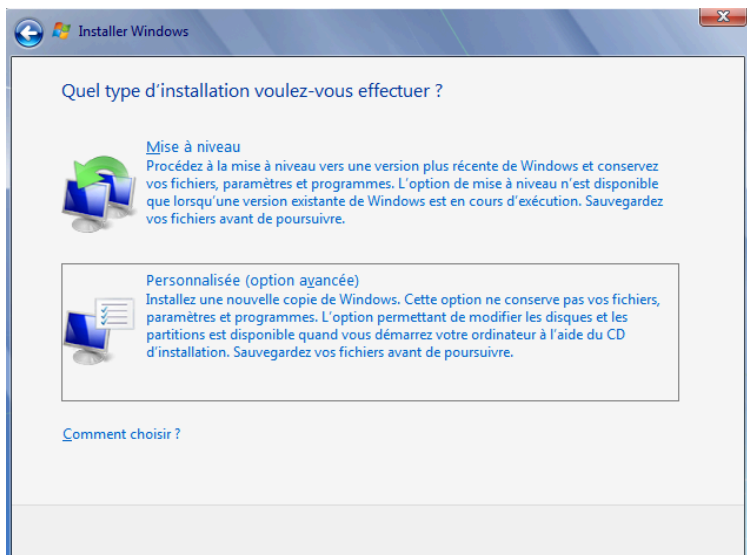


Ensuite, cliquez sur Installer maintenant.

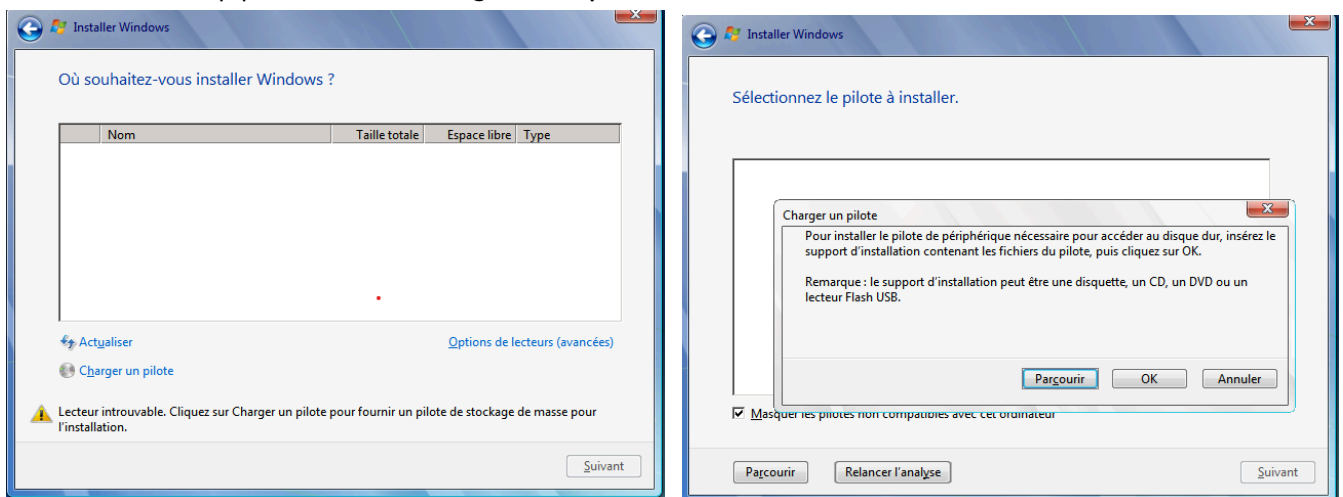


Lire et accepter le contrat de licence et appuyer sur **suivant**.

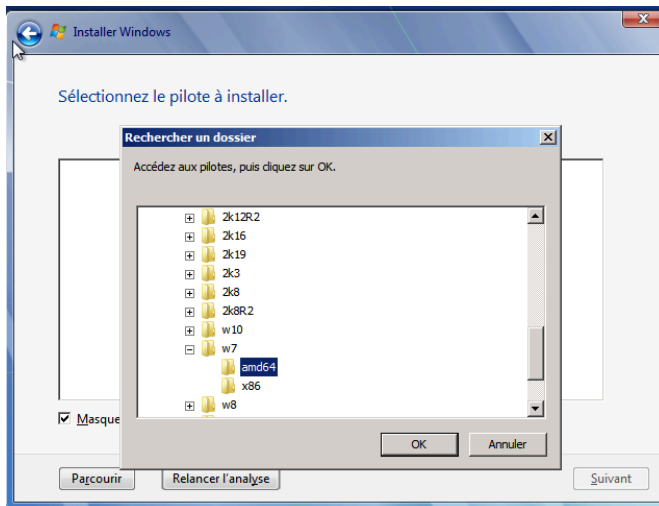
Il faut ensuite choisir une installation personnalisée.



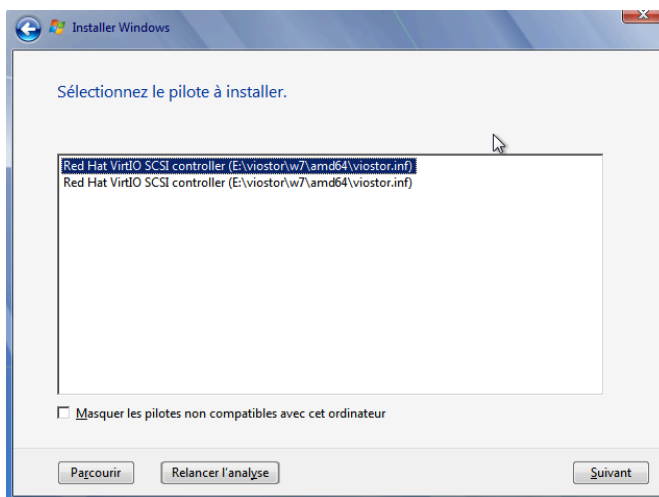
Ensuite, on appuie sur Charger un pilote -> Parcourir



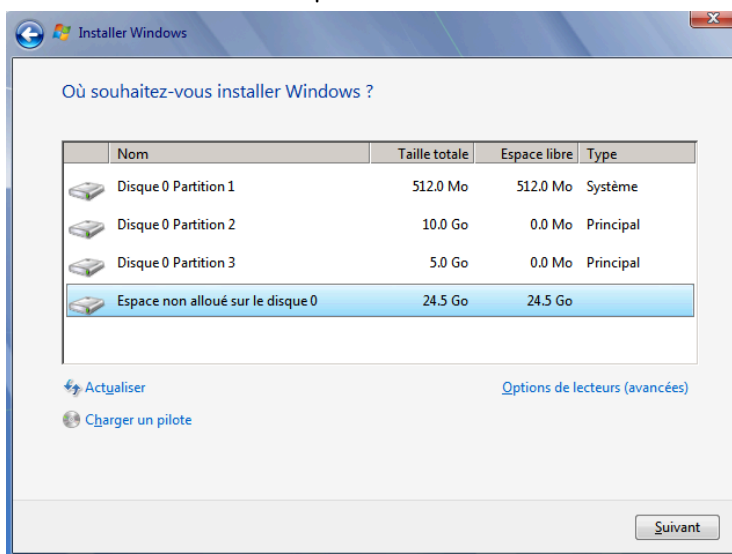
On choisit le cd(E) le cd qui contient le support virtuel. On sélectionne le fichier **viostor** -> **amd64**. On appuie sur OK une fois le chemin défini.



On décoche la case masquer les pilotes non compatibles et on sélectionne le cd contenant virtuo.

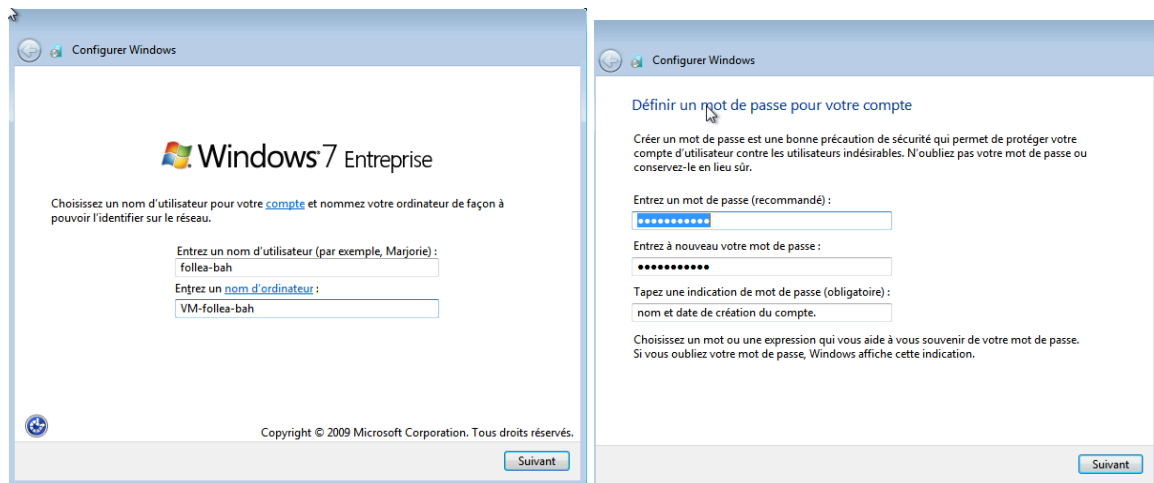


On sélectionne l'espace non alloué.

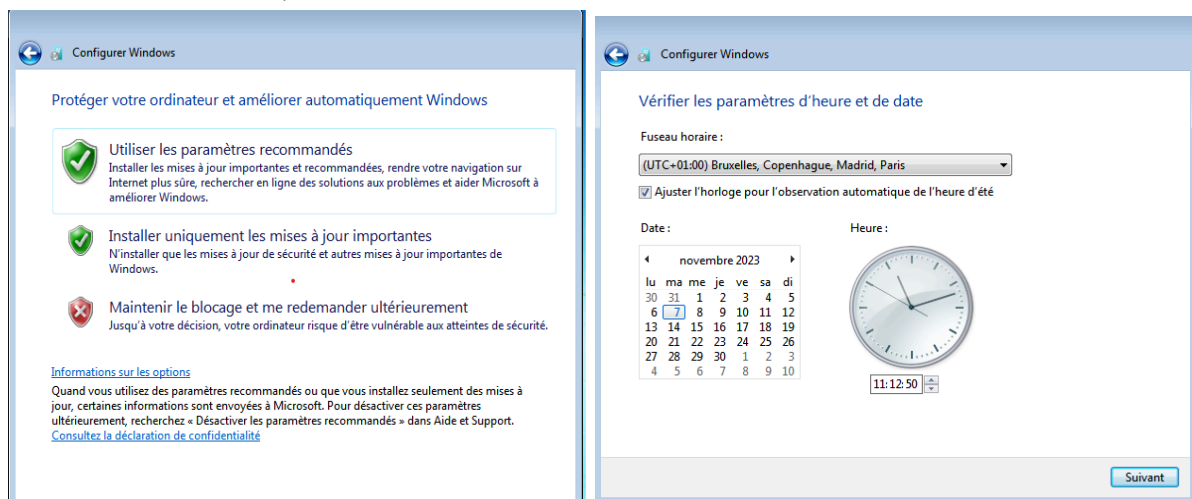


On patiente pendant l'installation et on clique sur **redémarrer maintenant**.

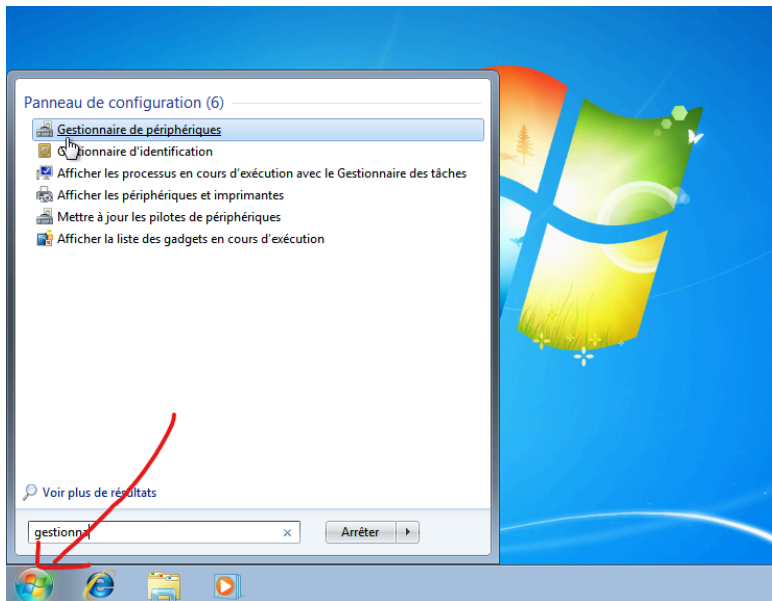
On va passer à la configuration de Windows. Il faut définir un nom en donnant le nom VM-login1-login2. Dans notre cas, VM-follea-bah. Il faut aussi définir un mot de passe ainsi qu'une indication au cas où on perd le mot de passe. Pour notre cas, le mot de passe est : folleabah07



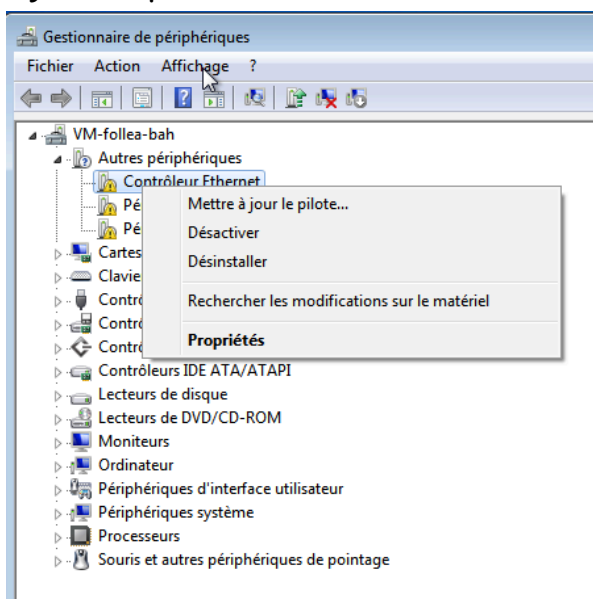
Pour la suite de la configuration de Windows, on utilisera les **paramètres recommandés**. Après cela, on choisit l'heure.



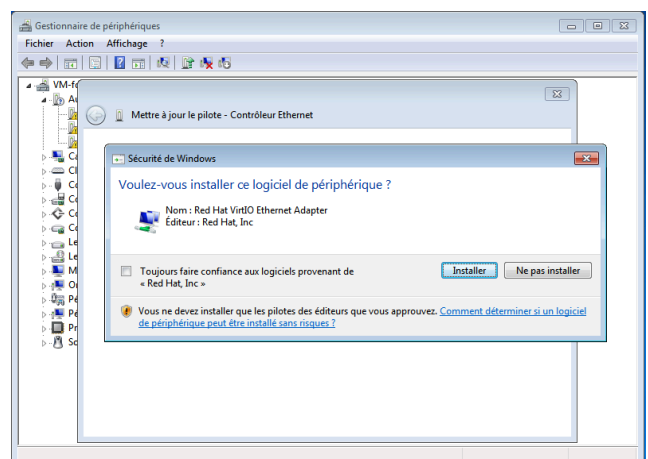
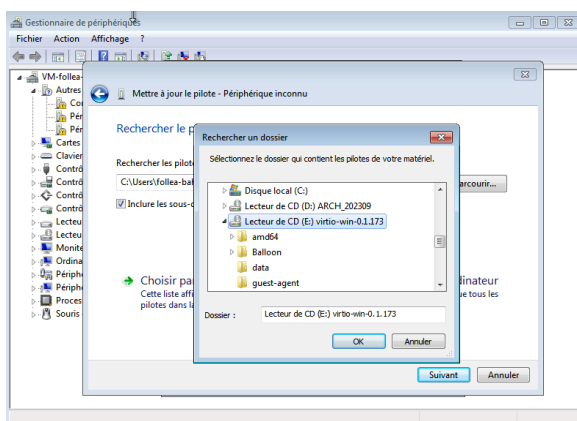
Il faut accéder au **gestionnaire de périphérique**. Pour ce faire, il faut appuyer sur la touche **Windows** ou cliquer sur le logo Windows en bas à gauche. Une fois cela fait, on tape dans la **barre de recherche** **gestionnaires de périphérique** et on l'ouvre.



Une fois dans le gestionnaire de périphérique, on sélectionne le **contrôleur Ethernet**. On fait un clic droit dessus et on va dans **Mettre à jour le pilote**.

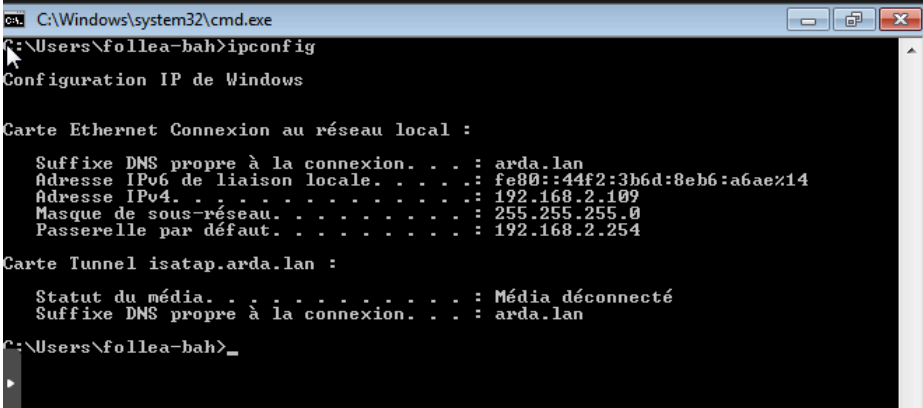


On va dans **Rechercher un pilote sur mon ordinateur**. Ensuite, on sélectionne le cd virtuo et on appuie sur **OK** pour mettre à jour. On appuie sur **installer**.



Maintenant, on va vérifier si on a bien une adresse ip. Pour ce faire, on ouvre le cmd en tapant dans la barre de recherche Windows "cmd" (même étape que pour ouvrir le gestionnaires de périphérique)

Dans le cmd, on tape la commande `ipconfig` pour voir les informations de votre IP, votre configuration IP. On peut donc bien voir qu'on a une adresse IP.



```
C:\Windows\system32\cmd.exe
C:\Users\follea-bah>ipconfig

Configuration IP de Windows

Carte Ethernet Connexion au réseau local :
    Suffixe DNS propre à la connexion. . . : arda.lan
    Adresse IPv6 de liaison locale. . . . . : fe80::44f2:3b6d:8eb6:a6ae%14
    Adresse IPv4. . . . . : 192.168.2.109
    Masque de sous-réseau. . . . . : 255.255.255.0
    Passerelle par défaut. . . . . : 192.168.2.254

Carte Tunnel isatap.arda.lan :
    Statut du média. . . . . : Média déconnecté
    Suffixe DNS propre à la connexion. . . : arda.lan
C:\Users\follea-bah>
```

Installation de Archlinux

Avant de commencer et pour être plus à l'aise, on peut rentrer la commande `loadkeys fr` pour passer son clavier en azerty.

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

Pour commencer, il faut formater les partitions Linux et Linux Home avec la commande `mkfs` (on peut faire la commande `lsblk` pour voir les partitions ainsi que le type de partitions grâce à la taille de la partitions).

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

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

On monte ensuite la partition Linux sur `/mnt` grâce à la commande `mount` (l'option `-v` est facultative, elle permet d'avoir un détail d'exécution de la commande).

```
root@archiso ~ # mount -v /dev/sda2 /mnt/  
mount: /dev/sda2 mounted on /mnt.
```

Il faut maintenant créer les fichiers `home` et `boot` dans `/mnt`.

```
root@archiso ~ # mkdir /mnt/boot
```

```
root@archiso ~ # mkdir /mnt/home
```

On va vérifier si on a bien du réseau. On peut utiliser la commande `ping` pour vérifier avec un site si on a bien une connexion.

Ici, on voit qu'on a une réponse du site donc on a de la connexion.

```
1 root@archiso ~ # ping google.com  
PING google.com (172.217.18.206) 56(84) bytes of data:  
64 bytes from ham02s14-in-f206.1e100.net (172.217.18.206): icmp_seq=1 ttl=115 time=3.12 ms  
64 bytes from ham02s14-in-f206.1e100.net (172.217.18.206): icmp_seq=2 ttl=115 time=3.55 ms  
64 bytes from par10s38-in-f14.1e100.net (172.217.18.206): icmp_seq=3 ttl=115 time=3.21 ms  
64 bytes from ham02s14-in-f206.1e100.net (172.217.18.206): icmp_seq=4 ttl=115 time=3.37 ms  
64 bytes from par10s38-in-f14.1e100.net (172.217.18.206): icmp_seq=5 ttl=115 time=3.46 ms  
64 bytes from ham02s14-in-f206.1e100.net (172.217.18.206): icmp_seq=6 ttl=115 time=3.50 ms  
^C  
--- google.com ping statistics ---  
6 packets transmitted, 6 received, 0% packet loss, time 5008ms  
rtt min/avg/max/mdev = 3.123/3.370/3.551/0.155 ms
```

On ouvre le fichier `mirrorlist` qui se trouve dans `/etc/`.

```
root@archiso ~ # vim /etc/mirrorlist
```

Une fois dans le vim, on va sur le site de arch linux pour regarder parmi la liste ceux qui ont le statut de **Successfully Syncing Mirrors**. Le premier étant bon pour notre machine (peut être pas la vôtre), il faut cliquer sur la touche `i` pour pouvoir le modifier et garder les un ou deux bons serveurs. Une fois les modifications effectuées, il faut faire `echap` et taper `:wq` pour quitter vim. Ces étapes sont à faire à chaque fois qu'on veut modifier et quitter vim.

```

##### Arch Linux mirrorlist generated by Reflector #####
# With:      reflector @/etc/xdg/reflector/reflector.conf
# When:      2023-11-15 20:45:50 UTC
# From:      https://archlinux.org/mirrors/status/json/
# Retrieved: 2023-11-15 21:45:13 UTC
# Last Check: 2023-11-15 20:06:16 UTC

#Server = https://mirror.cyberbits.eu/archlinux/$repo/os/$arch
#Server = https://mirror.osbeck.com/archlinux/$repo/os/$arch
#Server = https://mirror.ubrco.de/archlinux/$repo/os/$arch
#Server = https://mirror.moson.org/arch/$repo/os/$arch
#Server = https://mirror.sunred.org/archlinux/$repo/os/$arch
#Server = https://mirror.f4st.host/archlinux/$repo/os/$arch
#Server = https://de.arch.mirror.kescher.at/$repo/os/$arch
#Server = https://archlinux.thaller.ws/$repo/os/$arch
#Server = https://ftp.halifax.rwth-aachen.de/archlinux/$repo/os/$arch
#Server = https://at.arch.mirror.kescher.at/$repo/os/$arch
#Server = https://geo.mirror.pkgbuild.com/$repo/os/$arch
#Server = https://archmirror.it/repos/$repo/os/$arch
#Server = https://md.mirrors.hacktegit.com/archlinux/$repo/os/$arch
#Server = https://america.mirror.pkgbuild.com/$repo/os/$arch
#Server = https://mirrors.kernel.org/archlinux/$repo/os/$arch
#Server = https://asia.mirror.pkgbuild.com/$repo/os/$arch
#Server = https://seoul.mirror.pkgbuild.com/$repo/os/$arch
#Server = https://sydney.mirror.pkgbuild.com/$repo/os/$arch
#Server = https://mirrors.neusoft.edu.cn/archlinux/$repo/os/$arch
#Server = https://europe.mirror.pkgbuild.com/$repo/os/$arch

```

Il faut maintenant installer les packages de base avec la commande `pacstrap`.

```
root@archiso ~ # pacstrap -K /mnt base linux linux-firmware
```

On monte la partition de Linux home sur `/mnt/home/` et on va ensuite générer le fichier `fstab` grâce à la commande `genfstab`.

```
root@archiso ~ # mount -v /dev/sda3 /mnt/home/
mount: /dev/sda3 mounted on /mnt/home.
```

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

On va maintenant se bloquer dans `/mnt/` avec `arch-chroot`.

```
1 root@archiso ~ # arch-chroot /mnt
[root@archiso /]#
```

Pour cette étape, on va installer toutes les commandes nécessaires pour éviter de le faire par la suite. Il faudra installer grâce à la commande `pacman` les commandes suivantes : `vim` , `netctl`, `dhcpcd` , `grub` , `efibootmgr` , `os-prober` , `NetworkManager`

```
root@archiso ~ # pacman -S vim netctl dhcpcd grub efibootmgr os-prober NetworkManager
```

Avec la commande `vim`, on va ouvrir le fichier `locale.gen` et dé-commentez `fr_FR.UTF-8 UTF-8`.

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

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

Une fois cela fait, on va générer les locales.

```
[root@archiso /]# locale-gen
Generating locales...
```

On va ensuite générer le fichier **locale.conf**. On va utiliser la commande **vim**. Une fois dans le fichier, on va définir la variable **LANG**.

```
root@archiso ~ # vim /etc/locale.conf
```

```
LANG=fr_FR.UTF-8
```

On va ensuite ouvrir le fichier **vconsole.conf** pour garder le pc en azerty tout le temps quand on aura tout installer. On utilisera **KEYMAP=fr**.

```
root@archiso ~ # vim /etc/vconsole.conf
```

```
KEYMAP=fr
```

Nous allons ensuite configurer le nom de la machine sous forme **VM-AB-XY** ou **AB** sont les deux premières lettres de **login1** et **XY** les deux premières lettres du **login2**. On va donc créer un fichier **hostname** dans **/etc** ainsi que l'ouvrir avec **vim** et l'éditer.

```
root@archiso ~ # vim /etc/hostname
```

```
VM-FO-BA
```

Installation du Dual boot

Pour commencer, on va créer un point de montage (un dossier ou nous allons faire notre dual boot) qui s'appellera `/esp`. On utilisera la commande `mkdir`.

```
[root@archiso ~]# mkdir /esp
```

On va ensuite monter la partition EFI dans notre point de montage `/esp`.

```
[root@archiso ~]# mount /dev/sda1 /esp
```

Avant de procéder à la suite, on va vérifier si notre système est bien en 64 bits. On s'aidera de la commande `uname -m`. Ici, c'est bien le cas.

```
[root@archiso ~]# uname -m
x86_64
```

On peut donc procéder à l'installation du grub grâce à la commande `grub-install`. On indique obligatoirement `arch_grub` comme nom de `bootloader-id` ainsi que que le `boot-directory` est `/esp`.

```
[root@archiso esp]# grub-install --target=x86_64-efi --efi-directory=/esp --boot-directory=/esp --bootloader-id=arch_grub
Installing for x86_64-efi platform.
Installation finished. No error reported.
[root@archiso esp]#
```

Pour vérifier si tout va bien, on vérifie la présence du fichier `/esp/EFI/arch_grub/grubx64` en se déplaçant avec `cd` dans le répertoire et vérifie son contenu avec `ls`.

```
[root@archiso ~]# cd /esp/EFI/arch_grub
[root@archiso arch_grub]# ls
grubx64.efi
```

On va générer le fichier de configuration de grub avec la commande `mkconfig`.

```
[root@archiso ~]# grub-mkconfig -o /esp/grub/grub.cfg
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-linux
Found initrd image: /boot/initramfs-linux.img
Found fallback initrd image(s) in /boot:  initramfs-linux-fallback.img
Warning: os-prober will not be executed to detect other bootable partitions.
Systems on them will not be added to the GRUB boot configuration.
Check GRUB_DISABLE_OS_PROBER documentation entry.
Adding boot menu entry for UEFI Firmware Settings ...
done
```

On peut voir que tout se passe bien sauf qu'il ne détecte pas windows. Pour ce faire, on va ouvrir le fichier `/etc/default/grub` et dé-commenter la dernière ligne de commande ci-dessous en enlevant le `#`.


```
root@archiso ~ # vim /etc/default/grub
```

```
# Uncomment to use basic console
GRUB_TERMINAL_INPUT=console

# Uncomment to disable graphical terminal
#GRUB_TERMINAL_OUTPUT=console

# The resolution used on graphical terminal
# note that you can use only modes which your graphic card supports via VBE
# you can see them in real GRUB with the command `videoinfo'
GRUB_GFXMODE=auto

# Uncomment to allow the kernel use the same resolution used by grub
GRUB_GFXPAYLOAD_LINUX=keep

# Uncomment if you want GRUB to pass to the Linux kernel the old parameter
# format "root=/dev/xxx" instead of "root=/dev/disk/by-uuid/xxx"
#GRUB_DISABLE_LINUX_UUID=true

# Uncomment to disable generation of recovery mode menu entries
GRUB_DISABLE_RECOVERY=true

# Uncomment and set to the desired menu colors. Used by normal and wallpaper
# modes only. Entries specified as foreground/background.
#GRUB_COLOR_NORMAL="light-blue/black"
#GRUB_COLOR_HIGHLIGHT="light-cyan/blue"

# Uncomment one of them for the gfx desired, a image background or a gfxtheme
#GRUB_BACKGROUND="/path/to/wallpaper"
#GRUB_THEME="/path/to/gfxtheme"

# Uncomment to get a beep at GRUB start
#GRUB_INIT_TUNE="480 440 1"

# Uncomment to make GRUB remember the last selection. This requires
# setting 'GRUB_DEFAULT=saved' above.
#GRUB_SAVEDefault=true

# Uncomment to disable submenus in boot menu
#GRUB_DISABLE_SUBMENU=y

# Probing for other operating systems is disabled for security reasons. Read
# documentation on GRUB_DISABLE_OS_PROBER, if still want to enable this
# functionality install os-prober and uncomment to detect and include other
# operating systems.
GRUB_DISABLE_OS_PROBER=false
```

On va donc re générer le fichier grub.cfg pour voir si Windows est détecté. Pour notre cas, le fichier détecte bien Windows.

```
GRUB_DISABLE_OS_PROBER=false
"/etc/default/grub" 63L, 2072B écrit(s)
[root@VM-F0-BA ~]# grub-mkconfig -o /esp/grub/grub.cfg
Création du fichier de configuration GRUB■
Image Linux trouvée : /boot/vmlinuz-linux
Image mémoire initiale trouvée : /boot/initramfs-linux.img
Found fallback initrd image(s) in /boot: initramfs-linux-fallback.img
Attention : le sondeur de systèmes d'exploitation sera exécuté pour détecter d'autres partitions amorçables.
Sa sortie sera utilisée pour détecter les binaires amorçables qu'elles contiennent et créer de nouvelles entrées d'amorçage.
Windows Boot Manager trouvé sur /dev/sda1@EFI/Microsoft/Boot/bootmgfw.efi
Ajout de l'entrée du menu d'amorçage pour les paramètres du firmware UEFI ■
fait
[root@VM-F0-BA ~]#
```

On doit faire en sorte que son contenu du fichier /Boot/bootx64.efi soit remplacé par celui de grubx64.efi On pensera à laisser grubx64.efi à son emplacement et à garder une copie du fichier original bootx64.efi sous le nom bootx64.efi.initial dans le même répertoire.

On commence par créer une copie du fichier bootx64.efi

```
[root@archiso ~]# cp /esp/EFI/Boot/bootx64.efi /esp/EFI/Boot/bootx64.efi.initial
```

On copie le fichier bootx64.efi dans un fichier temporaire (temp).

```
root@archiso ~]# cp /esp/EFI/Boot/bootx64.efi temp
```

On copie le fichier grubx64.efi dans bootx64.efi

```
root@archiso ~]# cp /esp/EFI/arch_grub/grubx64.efi /esp/EFI/Boot/bootx64.efi
```

On copie le fichier temp dans grubx64.efi

```
root@archiso ~]# cp temp /esp/EFI/arch_grub/grubx64.efi
```

On supprime le fichier temp

```
[root@archiso ~]# rm temp
```

Tout ça permet donc d'inverser le contenu de grubx64.efi et bootx64.efi en crée une copie de bootx64.efi.

On va maintenant mettre un mot de passe pour root grâce à la commande passwd root.

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

On entre le mot de passe que l'on veut. Ici, ça sera : folleabah07

Ne pas oublier d'être en azerty (commande : "loadkeys fr" pour se mettre en azerty). Le caractère ne s'affiche pas quand on tape le mot de passe par mesure de sécurité.

Il faut le réécrire une deuxième fois pour le confirmer.

```
New password:
Retype new password:
passwd: password updated successfully
```

On va maintenant commencer la configuration de la connexion.

Il faut d'abord savoir le nom de son interface réseau. On utilise la commande ip a et regarder sur la photo ci dessous ou se trouve le nom.

```
root@archiso ~ # ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: ens18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether f6:af:98:b9:b4:1a brd ff:ff:ff:ff:ff:ff
    altname enp0s18
    inet 192.168.2.173/24 metric 100 brd 192.168.2.255 scope global dynamic ens18
        valid_lft 572sec preferred_lft 572sec
    inet6 fe80::f4af:98ff:feb9:b41a/64 scope link proto kernel_ll
        valid_lft forever preferred_lft forever
```

Ici, le nom de l'interface réseau est ens18.

On va créer un fichier de configuration dhcp `/etc/netctl/ens18-dhcp` (choisissez n'importe quel nom de fichier, mais il faut s'en souvenir).

```
root@archiso ~ # vim /etc/netctl/ens18-dhcp
```

Il recopier exactement comme dans la photo ci dessous

```
Description='Comexion internet en DHCP'
```

```
Interface=ens18
Connection=ethernet
IP=dhcp
```

On va ensuite activer et démarrer le profil netctl avec `enable` et `start`.

```
[root@VM-F0-B8 ~]# netctl enable ens18-dhcp
'/etc/systemd/system/multi-user.target.wants/netctl@ens18\x2ddhcp.service' -> '/usr/lib/systemd/system/netctl@.service'
generated '/etc/systemd/system/netctl@ens18\x2ddhcp.service.d/profile.conf'
```

```
root@archiso ~ # netctl start ens18-dhcp
```

On va continuer la suite sur le Archlinux qu'on a installé avec le grub.
On va aussi vérifier si l'installation du grub à bien fonctionné.

Il faut rentrer la commande `exit` pour sortir de l'emprisonnement de `/mnt`.

Une fois cela fait, on peut redémarrer la machine pour aller sur le dual boot et choisir Archlinux pour terminer la configuration du réseau. Il faut entrer la commande `reboot`.

Si tout se passe bien, vous devrez avoir Windows ainsi que Archlinux dans le dual boot. On sélectionne Archlinux.



Une fois sur Arch linux, il faut se connecter. Le login est root et le mot de passe est celui que vous avez choisi tout à l'heure avec passwd.

Pour finir, on va activer et démarrer le profil qu'on a créé avec systemctl en choisissant dhcpcd et NetworkManager.

```
[root@VM-F0-BA ~]# systemctl enable NetworkManager
Created symlink /etc/systemd/system/multi-user.target.wants/NetworkManager.service → /usr/lib/systemd/system/NetworkManager.service.
Created symlink /etc/systemd/system/dbus-org.freedesktop.nm-dispatcher.service → /usr/lib/systemd/system/NetworkManager-dispatcher.service.
Created symlink /etc/systemd/system/network-online.target.wants/NetworkManager-wait-online.service → /usr/lib/systemd/system/NetworkManager-wait-online.service.

[root@VM-F0-BA ~]# systemctl enable dhcpcd
Created symlink /etc/systemd/system/multi-user.target.wants/dhcpcd.service → /usr/lib/systemd/system/dhcpcd.service.

[root@VM-F0-BA ~]# systemctl start dhcpcd

[root@VM-F0-BA ~]# systemctl start NetworkManager
```

Pour vérifier si on a du réseau, on va ping le site de google.com et voir si on a une réponse.

```
[root@VM-F0-BA ~]# ping google.com
PING google.com (172.217.18.206) 56(84) octets de données:
64 octets de ham02s14-in-f206.1e100.net (172.217.18.206) : icmp_seq=1 ttl=115 temps=3.44 ms
64 octets de ham02s14-in-f206.1e100.net (172.217.18.206) : icmp_seq=2 ttl=115 temps=3.80 ms
64 octets de ham02s14-in-f206.1e100.net (172.217.18.206) : icmp_seq=3 ttl=115 temps=3.41 ms
64 octets de ham02s14-in-f206.1e100.net (172.217.18.206) : icmp_seq=4 ttl=115 temps=3.68 ms
```

On a bien du réseau.

La configuration du dual boot est terminée.

Pour une meilleur utilisation de la notice, vous pouvez l'ouvrir avec google docs :
https://docs.google.com/document/d/192AFYucWMTsMFmNVld46Swe7ovhvHclZRI4xhN_oMHI/edit?usp=sharing