

SAE : Dualboot - Rapport

RENAUD—GRANGE Jules

BUT info 1 TD A alpha

Table des matières

1 Introduction.....	2
2 Installation d'Ubuntu.....	2
2.1 L'image disque.....	2
2.2 La situation actuelle.....	3
2.2.1 Les composants.....	3
2.2.2 L'OS et les performances.....	3
2.3 L'installation.....	4
2.4 le MBR et la gestion des OS.....	5
2.4.1 Le partitionnement.....	5
2.4.2 Qu'est-ce que le SWAP.....	5
3 L'accès à distance.....	6
3.1 SSH et Putty.....	6
3.2 Transfert de fichiers.....	7
4 Les avantages et inconvénients du Multiboot.....	9
4.1 Des avantages.....	9
4.2 Pas que des avantages.....	9
4.3 Une alternative.....	9
5 Le serveur Minecraft.....	10
5.1 L'installation et l'initialisation du serveur.....	10
5.2 Problème de routage.....	11

1 Introduction

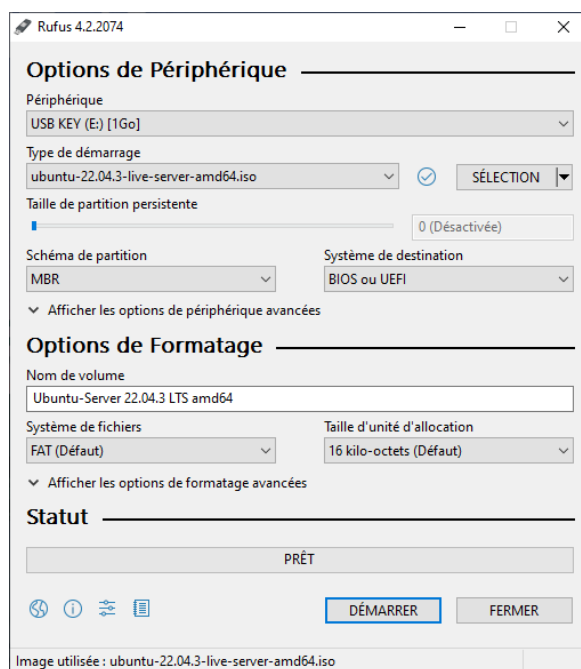
Pour mon Dualboot j'ai décidé d'utiliser un ancien ordinateur de bureau appartenant anciennement à ma mère, un HP Pavillon sous Windows 10 (originellement sous Windows 7) datant de 2015~2016, dans l'objectif d'en faire un serveur accessible en tout temps, pour par exemple stocker des données ou héberger un serveur Minecraft.

J'ai donc opté pour la version serveur d'Ubuntu : **Ubuntu Server 22.04 LTS**

2 Installation d'Ubuntu

2.1 L'image disque

J'ai donc commencé par télécharger l'image ISO (l'image disque) de la version souhaitée sur le site d'Ubuntu, afin de le mettre sur une clé USB, j'utilise Rufus 4.2, et en quelques minutes ma clé est prête.



Il est important de noter que j'ai au préalable, enregistré les données de la clé que je souhaitait garder, l'installation d'Ubuntu demandant un formatage de la clé, supprimant alors toute données.

J'ai dans un premier temps eu un léger problème à cette étape car j'avais d'abord opter pour la version la plus récente, Ubuntu Server 23.10, mais celle-ci refusait de s'installer, j'ai donc opté pour la dernière version la plus stable, et aussi la plus recommandée.

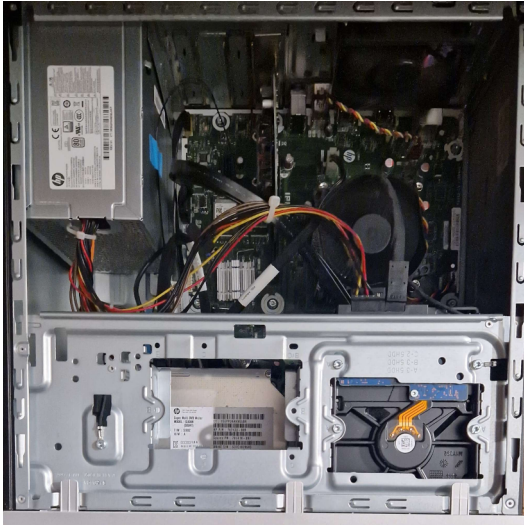
La clé maintenant prête, je peux passer sur ma machine, pour cela je la branche, dans mon salon, à ma télévision, je branche aussi un clavier et une souris, et avant toute chose je le lance normalement avec Windows, car il est important de parler des performances actuels du PC.

2.2 La situation actuelle

2.2.1 Les composants

Premièrement, les composants, la carte mère est une HP 2B47, équipée d'un processeur Intel i5 6400, un processeur équipé de 4 cœurs et à une fréquence de 2.70GHz, sur la carte mère est branchée une barrette de RAM DDR3 de 4Go et 1600Hz et est équipée d'un disque dur d'1To.

La carte graphique ne nous intéresse pas plus que ça, Ubuntu Server n'ayant pas d'interface graphique à proprement parler.



2.2.2 L'OS et les performances

L'ordinateur était originellement sous Windows 7 et a été mis à jour vers Windows 10, et ce n'est pas une exagération de dire qu'il a du mal à faire tourner Windows, le démarrage est lent, l'ouverture de l'explorateur de fichier prends du temps, je n'ose même pas lancer un jeu dessus... bref, il galère.

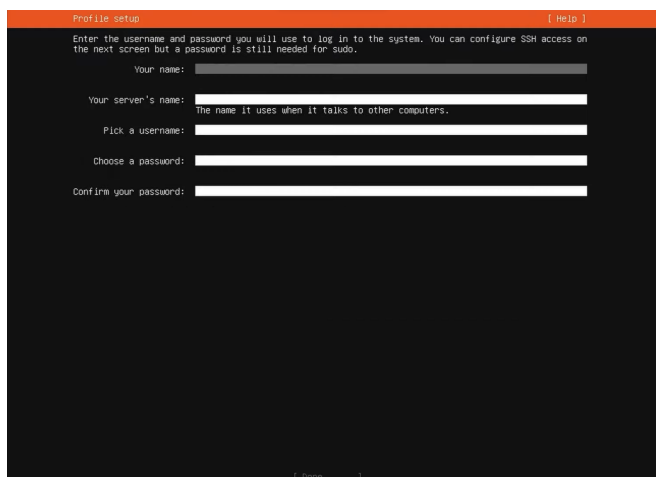
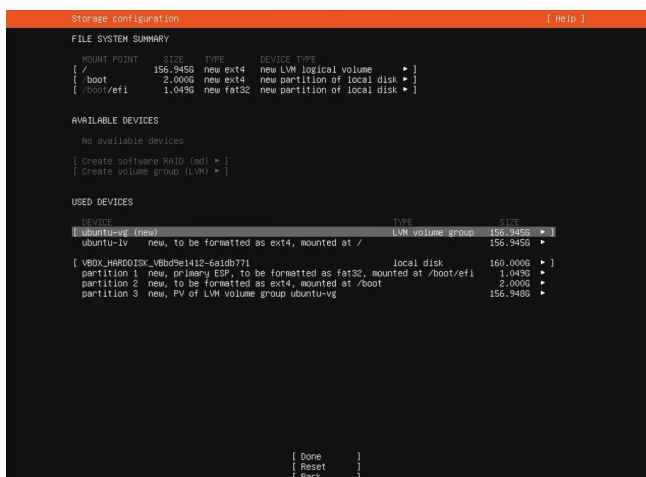
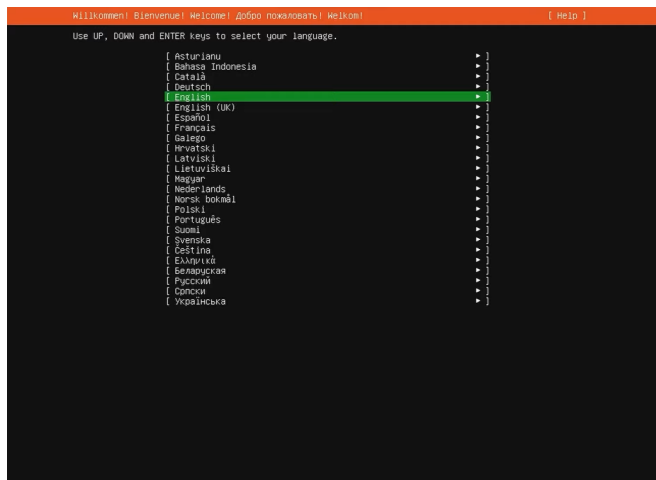
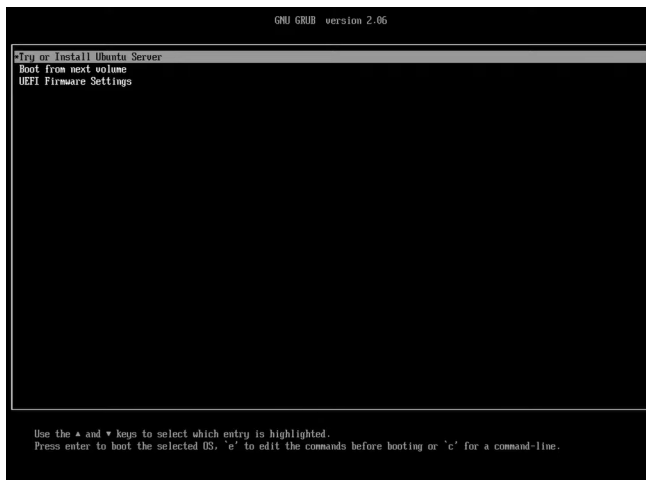
Mais ce n'est pas un problème, je compte l'utiliser en tant que serveur, et pour cela seul le processeur, la mémoire vive et le disque dur m'intéressent.

Il y a pleins de fichier appartenant à ma mère sur ce PC, et je compte bien tous les supprimer, mais certains peuvent être important, je prends donc le temps de tout mettre sur une clé avant de réinitialiser le PC, après cela, j'ai pu voir une légère améliorations des performances, sûrement du au nombre d'applications lancées en arrière-plan qui prenaient inutilement des performances sur le processeur/la mémoire.

2.3 L'installation

Passons maintenant au Dualboot, je branche ma clé avec Ubuntu, et je relance le PC, je reste appuyé sur F10 durant le démarrage afin d'accéder au BIOS et je change l'ordre de Boot, en mettant ma clé en première position.

Une fois fait, je me retrouve sur le menu d'installation d'Ubuntu, je choisis ma langue, la configuration de mon clavier (azerty/qwerty), le type d'installation (Ubuntu Server/minimized), la connexion au réseau, la partition dédiée à Ubuntu et finalement, mon profil, c'est à dire mon identifiant et mot de passe que j'utiliserais pour accéder au serveur, ainsi que le nom du serveur.



Images provenant d'une vidéo, je ne sais pas comment prendre des captures d'écrans lors de l'installation, ou dans le BIOS

Une fois ces configurations terminées, me voila sur le terminal de commande de mon serveur, mais je ne vais pas continuer à monopoliser la télé du salon, alors mon objectif est de pouvoir accéder à mon serveur depuis mon ordinateur personnel sur le même réseau.

2.4 le MBR et la gestion des OS

Avant de passer à la suite, je me dois de faire une parenthèse pour parler du MBR et du BIOS :

Lorsque l'utilisateur allume l'ordinateur, celui-ci lance la séquence de démarrage contenue dans le BIOS, celle-ci vient exécuter le logiciel de démarrage écrit dans le MBR du disque dur, ce logiciel va ensuite retrouver la partition contenant le/les systèmes d'exploitations installés et va le lancer.

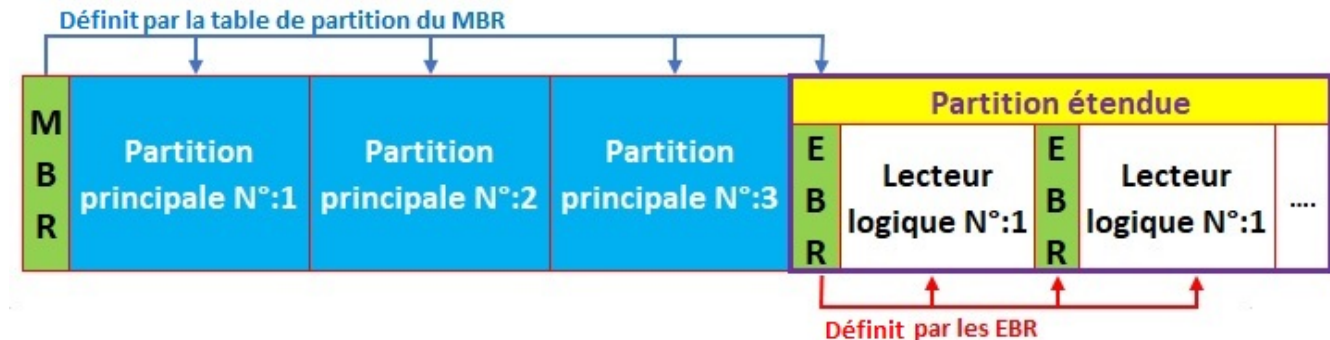
Le MBR (Master Boot Record) est donc le premier secteur adressable d'un disque dur, et contient la table des partitions, ainsi qu'une routine d'amorçage, un code permettant de charger l'OS, ou, dans le cas d'un multiboot, contient un chargeur d'amorçage, tel que Gnu Grub, qui permet, au démarrage de la machine de choisir quel OS lancer.

Voulant simplement un serveur sous Ubuntu, je ne compte pas utiliser Windows à nouveau sur mon serveur, donc je n'ai pas installé de programme d'amorçage, mais il est très pratique pour pouvoir tout simplement choisir l'OS voulu sans avoir à repasser par le BIOS.

2.4.1 Le partitionnement

Le partitionnement est la séparation de différentes zones dans l'espace du disque dur, ce qui permet d'installer différents OS n'utilisant pas le même système de fichier.

Une partition étendue est une partition qui contient des sous-partitions.



2.4.2 Qu'est-ce que le SWAP

Le SWAP, l'espace d'échange, est une zone du disque dur faisant partie de la mémoire virtuelle (composée de la RAM et du SWAP), il est utilisé pour décharger la RAM lorsqu'elle est saturée.

L'espace d'échange est utile, par exemple, lorsque l'on met l'ordinateur en mode veille.

Cet espace peut être sous forme de partition du disque dur ou sous la forme d'un fichier (sous Ubuntu l'espace d'échange est sous forme de fichier depuis la version 17.04).

3 L'accès à distance

3.1 SSH et Putty

Après quelques recherches, je découvre l'application Putty, qui me permet de faire exactement cela, mais il faut d'abord, du côté du serveur, faire quelques manipulations :

```
$ sudo apt install openssh-server
```

SSH (Secure Socket Shell) est un protocole réseau permettant d'accéder à distance à un ordinateur, de manière sécurisée.

Pour me connecter au serveur via Putty, il me faut maintenant 2 informations, l'adresse IP de ma machine sur le réseau, et le port par lequel SSH me permet l'accès.

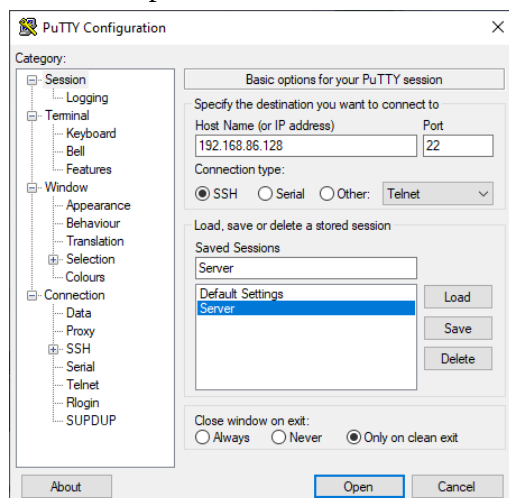
```
$ ip a et $ systemctl status ssh
```

```
hayro@hayroserver: ~  
hayro@hayroserver:~$ ip a  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default  
    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  
        valid_lft forever preferred_lft forever  
2: enp3s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000  
    link/ether 74:85:2a:1e:ca:fc brd ff:ff:ff:ff:ff:ff  
    inet 192.168.86.128/24 metric 100 brd 192.168.86.255 scope global dynamic enp3s0  
        valid_lft 84030sec preferred_lft 84030sec  
    inet6 fe80::7685:2aff:fe1e:cafc/64 scope link  
        valid_lft forever preferred_lft forever  
3: wlp2s0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000  
    link/ether 60:6d:c7:70:66:b1 brd ff:ff:ff:ff:ff:ff  
hayro@hayroserver:~$
```

```
hayro@hayroserver:~$ systemctl status ssh  
● ssh.service - OpenBSD Secure Shell server  
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: en  
   Active: active (running) since Thu 2023-11-30 15:53:46 UTC; 42min ago  
     Docs: man:sshd(8)  
           man:sshd_config(5)  
   Process: 838 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)  
    Main PID: 917 (sshd)  
      Tasks: 1 (limit: 4428)  
     Memory: 6.7M  
        CPU: 131ms  
   CGroup: /system.slice/ssh.service  
           └─917 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"  
  
nov. 30 15:53:42 hayroserver systemd[1]: Starting OpenBSD Secure Shell server...  
nov. 30 15:53:46 hayroserver sshd[917]: Server listening on 0.0.0.0 port 22.  
nov. 30 15:53:46 hayroserver sshd[917]: Server listening on :: port 22.  
nov. 30 15:53:46 hayroserver systemd[1]: Started OpenBSD Secure Shell server.  
nov. 30 16:22:55 hayroserver sshd[1605]: Accepted password for hayro from 192.168.86.128 port 22.  
nov. 30 16:22:55 hayroserver sshd[1605]: pam_unix(sshd:session): session opened for hayro (uid=0)  
lines 1-19/19 (END)
```

grâce à cette dernière commande je peux savoir le statut de SSH, voir si il est bien actif, et voir quel port utiliser pour me connecter.

Maintenant il me suffit de rentrer l'adresse ip et le port sur Putty, choisir le type de connexion SSH, et je suis dessus, je rentre mon identifiant et mot de passe, et j'ai l'accès complet au serveur depuis mon ordinateur personnel.



Je n'ai maintenant plus besoin de monopoliser la télévision, et je peux positionner mon serveur dans l'entrée, à côté de mon routeur afin de le connecter au réseau, je n'aurai donc, normalement, plus besoin d'y toucher.

3.2 Transfert de fichiers

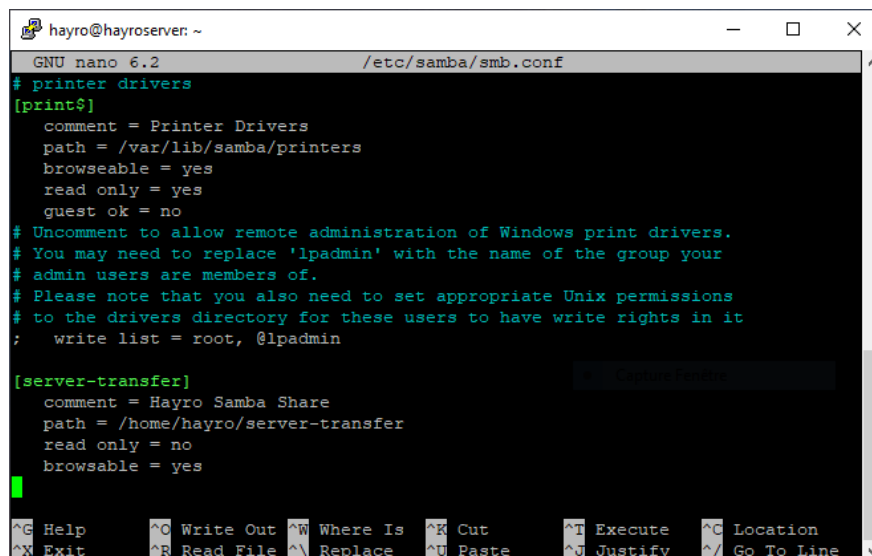
En plus de cela, je voudrais un moyen de transférer des fichiers de mon pc personnel au serveur, pour cela j'installe samba :

```
$ sudo apt install samba
```

Une fois installé, je crée un nouveau répertoire, que j'appelle server-transfer qui va servir d'intermédiaire entre mon PC et le serveur, je dois maintenant changer le fichier de configuration de samba pour lui dire quel répertoire utiliser, pour accéder au fichier et le modifier :

```
$ sudo nano /etc/samba/smb.conf
```

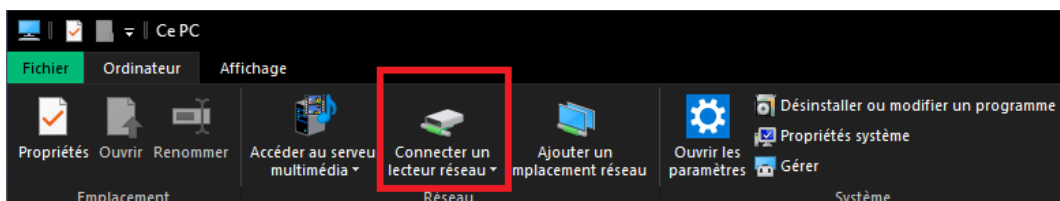
auquel j'ajoute ces 5 lignes à la fin :



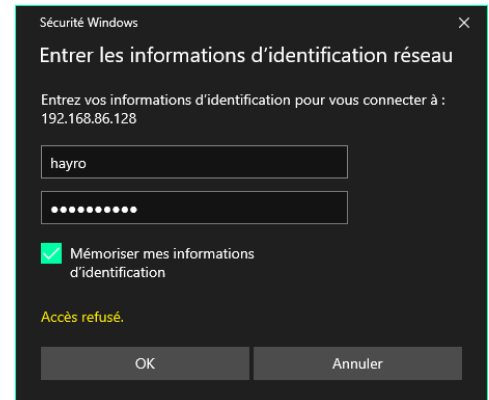
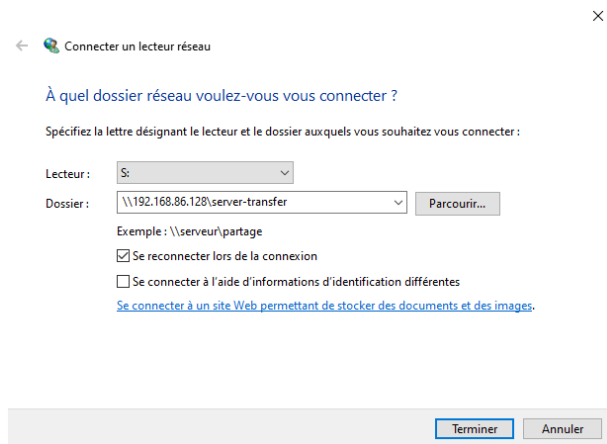
```
GNU nano 6.2 /etc/samba/smb.conf
# printer drivers
[print$]
    comment = Printer Drivers
    path = /var/lib/samba/printers
    browseable = yes
    read only = yes
    guest ok = no
# Uncomment to allow remote administration of Windows print drivers.
# You may need to replace 'lpadmin' with the name of the group your
# admin users are members of.
# Please note that you also need to set appropriate Unix permissions
# to the drivers directory for these users to have write rights in it
; write list = root, @lpadmin

[server-transfer]
    comment = Hayro Samba Share
    path = /home/hayro/server-transfer
    read only = no
    browsable = yes
```

Ensuite, depuis mon PC, j'ouvre l'explorateur des fichiers, où je connecte un nouveau lecteur réseau :



Je rentre l'ip ainsi que le nom du répertoire et je choisi S: pour « Serveur » en tant que lecteur, l'accès m'est refusé, mais c'est normal, il faut que je me connecte au serveur :



Un nouveau répertoire est apparu dans mon explorateur des fichiers, il fait office de passerelle entre mon PC et le serveur, tout fichier que je mets dedans, depuis le serveur ou depuis le PC, est accessible depuis les 2.

4 Les avantages et inconvénients du Multiboot

4.1 Des avantages

Le principal avantage du Dualboot est la polyvalence, Linux peut être plus pratique dans certains cas, il est largement plus sécurisé et consomme moins de ressources, mais Windows dispose de certaines applications que Linux n'a pas, et dans mon cas, la plupart des jeux auquel je joue ne sont pas sur Linux, donc garder un accès à Windows est obligatoire.

Le fait d'avoir accès à ces 2 OS sur un seul PC est donc un avantage énorme, c'est un certain gain pratique (simple de passer d'un OS à l'autre), et surtout économique, pas besoin d'acheter 1 PC par OS.

D'autres avantages existes, par exemples si l'un des deux OS tombe en panne, on peut démarrer avec l'autre afin de réparer sa partition, ou au moins sauvegarder les données.

4.2 Pas que des avantages

Certes utile, le Dualboot a aussi ses défauts, une mise à jour d'un des OS est susceptible d'écraser le chargeur de démarrage, et si le démarrage est endommagé... il est possible que l'on perde l'accès à tous les OS, inutile de mentionner que reconfigurer le démarrage sans accès à un système d'exploitation n'est pas une mince affaire.

Autre problème, Linux est certes connu pour sa sécurité, mais un virus téléchargé sur Windows n'affectera pas que Windows, Linux, et toutes nos données se trouvant sur le même disque dur.

4.3 Une alternative

Une alternative au Dualboot est la machine virtuelle, un programme qui virtualise le matériel d'un PC et permet d'y installer un OS.

La machine n'ayant pas accès aux composants physiques du PC, on peut dire au revoir aux performances, mais une machine virtuelle présente certains avantages :

Premièrement, une machine virtuelle est très simple d'utilisation, car un programme s'en charge pour nous (Virtual Box par exemple).

Deuxièmement, elle est extrêmement sécurisée, car même si l'on télécharge et exécute un virus, les dégâts se font à l'intérieur de la machine virtuelle et ne mettra pas en danger notre PC.

Finalement, il est possible d'avoir deux OS différents ouverts en même temps, ce qui peut être pratique pour plein de raisons.

5 Le serveur Minecraft

Mon objectif principal avec ce dualboot est de mettre en place un serveur Minecraft, je vais donc, rapidement parler du processus pour cela.

5.1 L'installation et l'initialisation du serveur

La première étape pour cela est de télécharger java, afin de pouvoir exécuter des fichiers .jar, Ainsi que screen, qui permet d'ouvrir un « terminal secondaire », qui s'avère être très utile pour s'occuper du serveur.

```
$ sudo apt install openjdk-19-jre-headless et $ wget screen
```

Deuxièmement, il faut activer ufw (Uncomplicated FireWall) tout en laissant passer certains ports, le port pour se connecter au serveur minecraft (25565) et le port de SSH :

```
$ sudo ufw allow OpenSSH , $ sudo ufw allow 25565 et enfin $ sudo ufw enable
```

Ensuite il suffit de télécharger la version serveur de minecraft, directement sur leur site, l'envoyer au serveur via le répertoire créé avec samba, créer un répertoire où l'on déplace le fichier et où les autres fichiers seront créés et L'exécuter avec java une première fois pour générer les fichiers importants, comme le eula.txt et le server.properties.

Comme on veut pouvoir rapidement lancer le serveur, autant créer un fichier start.sh afin d'avoir simplement à l'exécuter pour lancer le serveur, le fichier contient :

```
$ java -Xms1024M -Xmx4G -jar minecraft_server.1.20.1.jar nogui
```

- *Xms définit la RAM minimale allouée au serveur (ici 1024Mo)*
- *Xmx définit la RAM maximale allouée au serveur (ici 4Go)*

Après la première exécution, le serveur ne va pas pouvoir se lancer, car il faut accepter le EULA, les conditions générales d'utilisations, celui-ci est simplement un document txt ou il faut changer le « eula=false » en « eula=true » grâce à la commande nano.

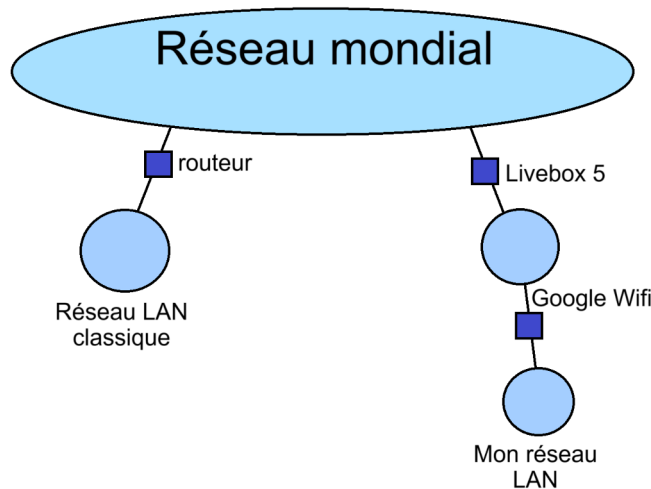
Maintenant que l'on a accepté le EULA, on peut à nouveau exécuter le fichier start.sh, mais avant cela, il est préférable d'ouvrir un sous-terminal avec screen et d'exécuter le fichier dedans.

Après un chargement plus ou moins long, le serveur est accessible, mais seulement en LAN, c'est à dire pour les machines sur le même réseau que le serveur, ce qui n'est pas réellement ce que je voudrais, ayant pour objectif de jouer avec mes amis dessus, et mes amis n'habitent pas chez moi.

Il faut alors changer la table de routage de mon routeur afin de permettre à quelqu'un qui possède mon adresse Ip WAN et le port du serveur de se connecter.

5.2 Problème de routage

Ma configuration réseau actuelle pose souci car j'ai, en plus de ma Livebox 5 d'Orange, acheté des google wifi afin de connecter mon pc en Ethernet, tout en étant à plus de 10 mètres de la box, (la connexion Wifi et Ethernet est bien meilleure avec eux) ce qui fait deux réseaux LAN.



On peut voir un réseau LAN classique comme un îlot rattaché au continent qu'est le réseau mondial par un pont : son routeur.

Mon réseau, lui, est semblable à un îlot, rattaché à un îlot, lui même rattaché au continent.

Ce qui a rendu le routage bien plus compliqué, sans entrer dans les détails.

Mais une fois le routage terminé, j'ai pu tester avec un ami à moi, et il a pu se connecter sans aucun souci grâce à l'adresse WAN et le port du serveur.

Dernier problème, les performances du serveur ne sont pas géniales, j'ai donc commandé une seconde barrette de RAM de 4Go dans l'espoir de voir les problèmes diminuer, mais je ne peux encore rien dire à ce sujet, j'attends encore le colis.

C'est la fin, merci d'avoir lu mon rapport.