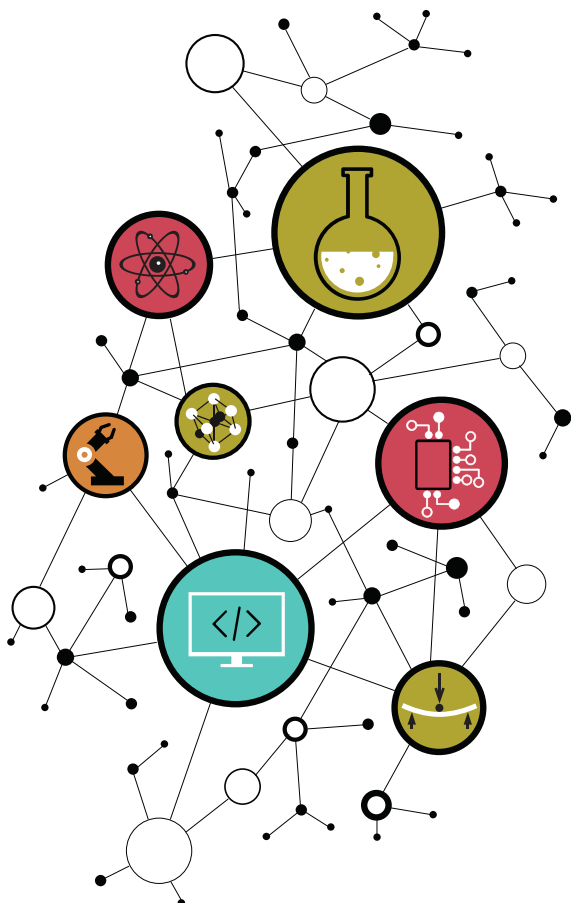


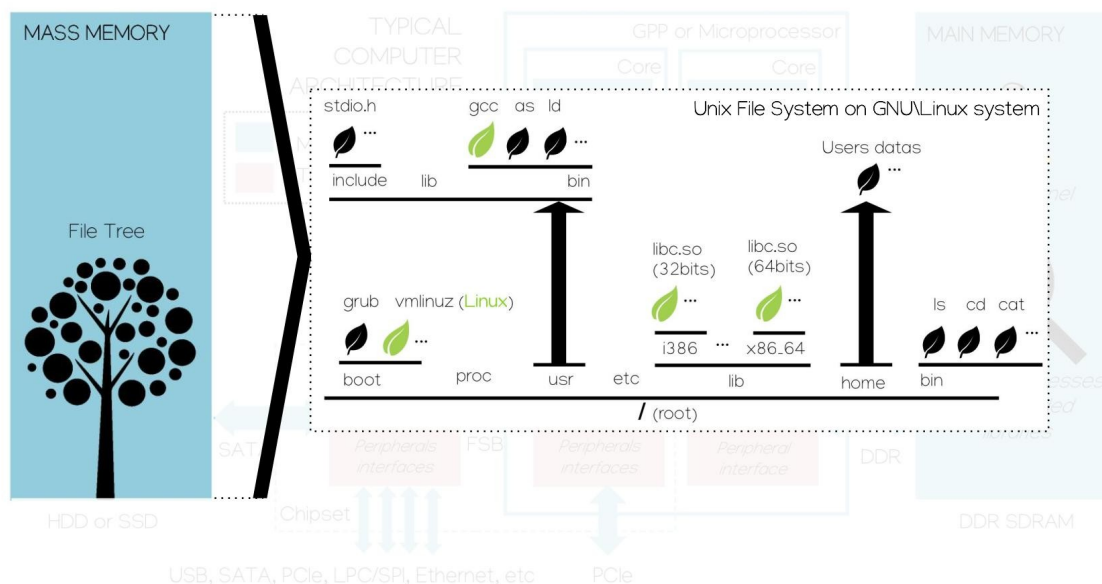
## PARTIE 10

# MÉMOIRE DE MASSE ET STOCKAGE DES FICHIERS



### I. Du support physique à la représentation logique

Les médias physiques de stockage de masse offre une stratégie de représentation et de classification de l'information sous forme d'arborescence de fichiers. Il est à noter qu'une mémoire physique (espace de stockage) est toujours pilotée par un périphérique matériel d'interface nommé contrôleur. Celui-ci est chargé d'écouter les requêtes (opérations de lecture ou d'écriture, adresse, nombre d'octets, etc) et de répondre à celle-ci en délivrant ou en stockant l'information demandée.

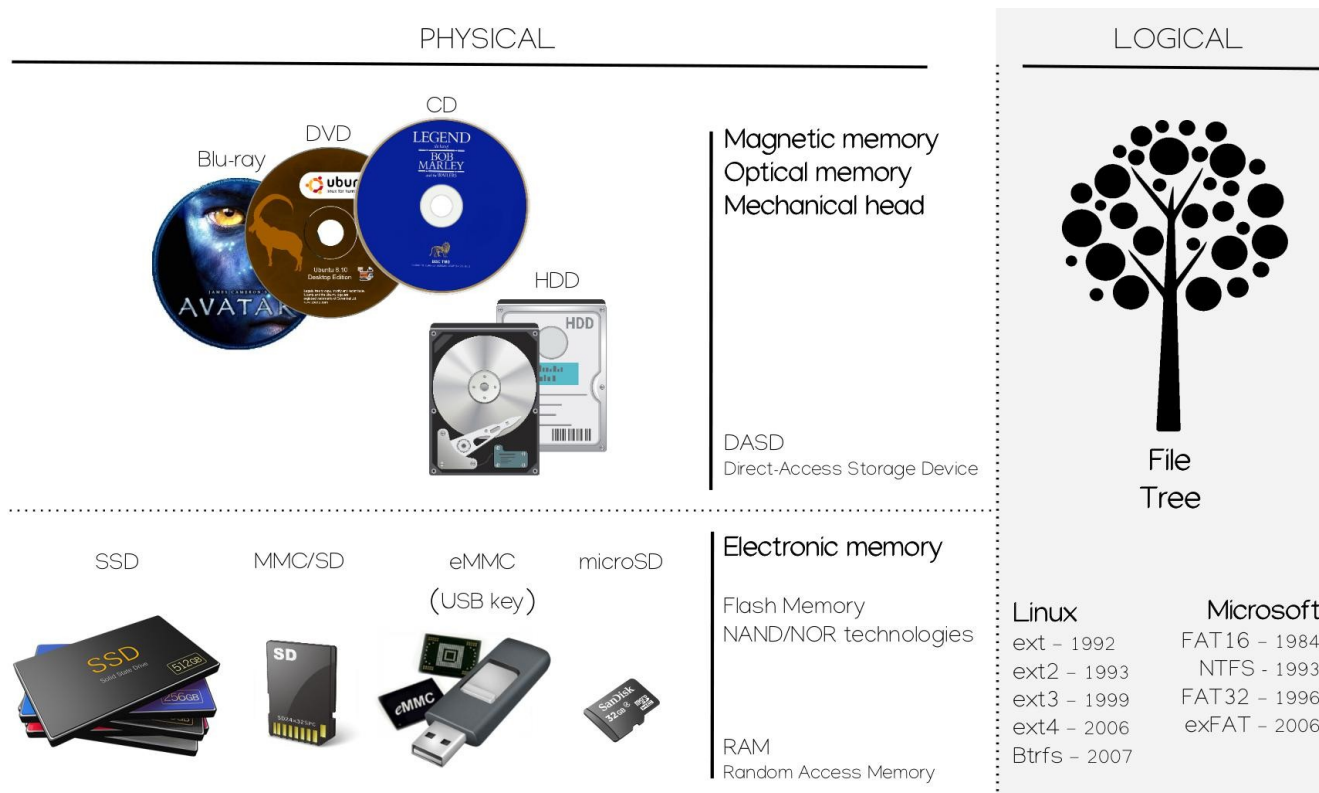


Rappelons les 3 familles de mémoires rencontrées sur ordinateur :

- **Mémoire cache** : mémoire adressable par association (associative) de technologie SRAM. À chaque octet copié en cache (niveaux L1\L2\L3) depuis la mémoire principale est associé un emplacement (par indexage) dans une ligne de cache.
- **Mémoire principale** : mémoire adressable par octet de technologie DRAM. Chaque octet possède une adresse mémoire unique typiquement représentée au format hexadécimal (32-bit, 64-bit, exemple *39-bit physical* et *48-bit virtual*, ...; `cat /proc/cpuinfo`). Sur processeur intégrant une MMU (GPP, AP, ...) nous distinguons une adresse virtuelle (vision système, CPU et développeur) d'une adresse physique (limitation matérielle sur la machine). Dans les langages de programmation offrant une abstraction aux langages machines, les adresses en mémoire principale sont le plus souvent nommées pointeurs voire références. Ces mémoires sont souvent nommées mémoires vives.
- **Mémoire de masse** : mémoire adressable par chemin dans une arborescence de fichiers de technologies SSD, HDD, MMC, ... Cette mémoire stocke l'information en implémentant les concepts de classification de fichiers dans des répertoires. Un fichier possède donc une adresse nommée chemin (*path*) dans une arborescence dont la base est nommée racine (`/` ou *root* sur système Unix-like).

Les systèmes *Unix-like*, comme les systèmes d'exploitation GNU/Linux, héritent pour la plupart d'une arborescence de fichiers Unix (`/boot`, `/proc`, `/usr`, `/lib`, `/bin`, `/home`, etc). Rappelons que le système original Unix se voulait d'une philosophie simple et minimaliste. Sous Unix, tout est fichier avec une gestion élémentaire (*open*, *read*, *write* et *close*). Le système se veut également implémenter un modèle à 3 couches (*kernel*, *shell* et *utilities*). Nous pouvons observer dans l'illustration ci-dessus quelques un des programmes et bibliothèques les plus connus du système.

Dans les systèmes numériques de traitement de l'information actuels, une mémoire de masse est une mémoire persistante dite non volatile (persistance de l'information sans apport d'énergie électrique). Une mémoire de masse est accessible en lecture voire en écriture. Elle sont souvent de plus grandes capacités que les mémoires vives mais restent plus lentes (mémoires vives de technologies volatiles SRAM ou DRAM). Hors communication extérieure au système (Internet, réseau Ethernet, clé USB, etc), une mémoire de masse stocke et représente l'ensemble des savoirs et savoirs-faire statiques d'une machine !



Plusieurs technologies de mémoire de masse sont actuellement en usage (HDD, SSD, MMC SD, MMC microSD, CD, DVD, Blu-ray, ...), tout comme certaines sont maintenant tombées en désuétude (K7 audio, disquette, VHS, VHS-c, carte perforée, tore magnétique, ...). Chaque technologie offre son lot hérité d'avantages, d'inconvénients, de compromis et se trouve adaptée à des besoins et des marchés ciblés. Les mémoires de masse actuellement les plus rapides sur le marché, mais toujours les plus coûteuses pour de grandes capacités de stockage, sont les mémoires électroniques de technologie Flash NOR ou NAND.

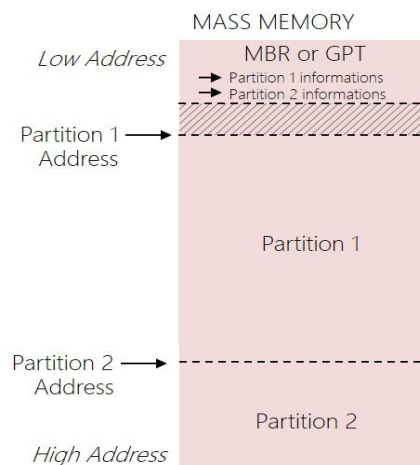
Indépendamment des technologies physiques de stockage utilisées, la représentation logique de l'information offerte par le système se base sur les concepts de partitions (zones logiques contiguës de la mémoire physique) et d'adressage de l'information par arborescence de fichiers. Plusieurs technologies de **systèmes de fichiers (FS ou File System)** sont en usage à notre époque. *Btrfs*, *ext4*, *SquashFS*, ... sous Linux ; *NTFS*, *FAT32*, *VFAT*, ... sous Windows ; et encore bien d'autres notamment pour d'autres systèmes d'exploitation (BSD, Mac OS X, etc). Chaque technologie offre son lot d'avantages et d'inconvénients. Prenons l'exemple de la technologie FAT32 encore très utilisée à notre époque (systèmes embarqués, clé USB, etc) :

- Taille maximale d'un fichier : 4 GB
- Taille maximale théorique d'une partition : 16 TB
- Nombre maximal de fichiers : 268 M
- Nombre maximal de fichiers par répertoire : 65534

## II. Table des partitions

Une partition représente une zone logique contiguë de la mémoire physique. Pour un support donné et en fonction des besoins, il est possible de définir plusieurs partitions, de différentes tailles, natures et à différents emplacements en mémoire. Pour ce faire, deux technologies représentant la table des partitions d'un média dominant le marché :

- **MBR (Master Boot Record)**<sup>27</sup>, le support physique peut contenir quatre partitions primaires maximum, toutes de tailles inférieures à 2 TB. Technologie encore très utilisée dans les systèmes embarqués mais en voie d'extinction dans les ordinateurs personnels ;
- **GPT (GUID Partition Table)**, rétrocompatible MBR et offrant moins de limitations mais manquant parfois de support sur certains systèmes.



Dans cet exercice, nous allons travailler sur une clé USB 16 Go en supprimant l'ancienne table des partitions et en la remplaçant par un nouveau MBR. Attention, certaines étapes exécutées en tant que super utilisateur **root (sudo)** sont critiques et risqueraient notamment de supprimer la table des partitions du disque système de façon irréversible. Ne surtout pas se tromper dans le choix du périphérique matériel `/dev/sdX` durant les exercices qui suivent !

Placez-vous dans le répertoire `/disco/mass/` puis connectez la clé USB à votre machine. Identifiez son nom (de la forme `/dev/sdX`) et ses caractéristiques.

```
lsblk
lsblk -f

sudo sfdisk -l

export DISK=/dev/<your_device_name>    # À modifier selon le nom de votre composant
```

Identifiez le paramètre système `block size` relatif à votre support (valeur dépendant de multiples paramètres), puis effacer l'ancienne table des partitions. Analysez la sortie.

```
sudo blockdev --getbsz ${DISK}

sudo dd if=/dev/zero bs=1K count=10K of=${DISK}

lsblk

sudo dd if=${DISK} bs=1K count=1 of=./mbr.bin

xxd ./mbr.bin > mbr.txt
```

Que constatons-nous ?

<sup>27</sup> <https://doc.ubuntu-fr.org/mbr>

Nous allons créer une nouvelle table des partitions en utilisant l'utilitaire `sfdisk`. Néanmoins, d'autres utilitaires existent pour créer des tables des partitions (`parted`, `fdisk`, ...). Créez une nouvelle table des partitions et analysez la sortie.

```
sudo sfdisk ${DISK}
>>> help -> [ENTER]
>>> ,, -> [ENTER]
>>> quit -> [ENTER]
>>> Y -> [ENTER]

lsblk

sudo dd if=${DISK} bs=1K count=1 of=./mbr.bin

xxd ./mbr.bin > mbr.txt

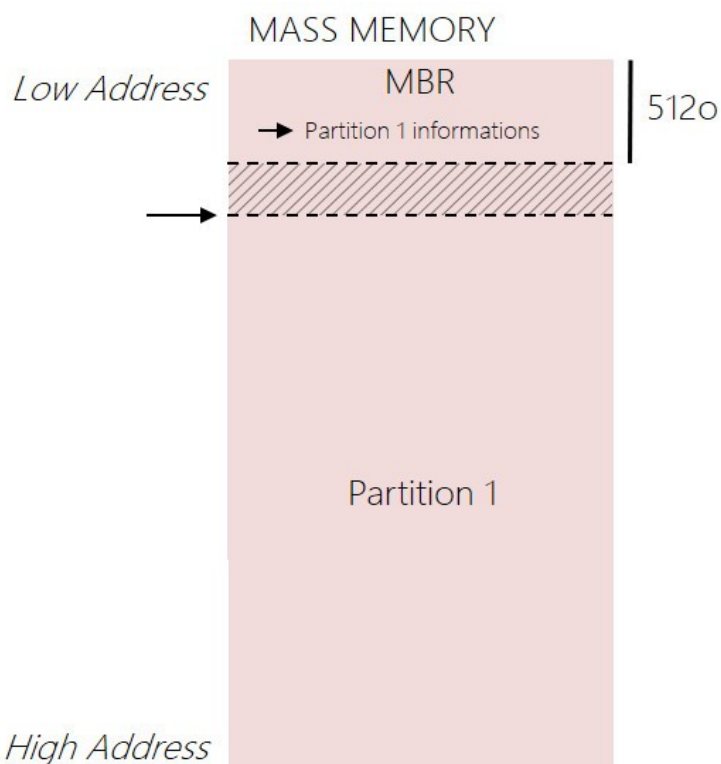
file mbr.bin
```

En s'aidant d'Internet, précisez la taille d'un MBR ?

Par quelle suite d'octets se termine toujours un MBR ? Vérifiez que cette suite binaire est bien présente après création de la table des partitions.

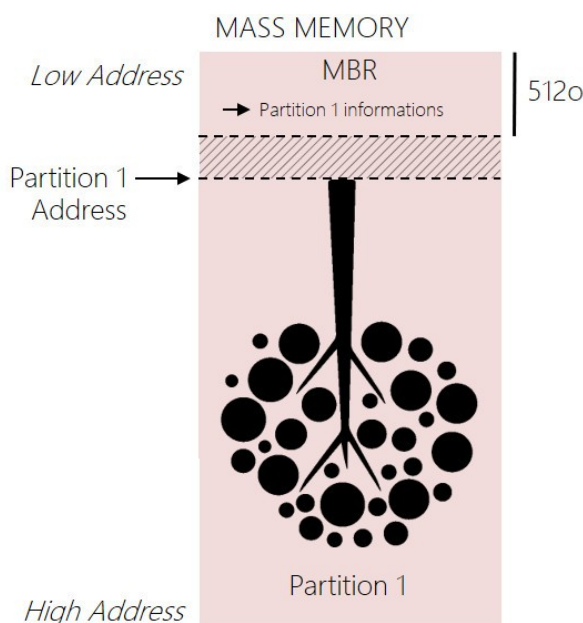
Quelle est la taille d'un bloc logique sur notre clé USB ?

Sur le schéma suivant, précisez à quelle adresse (en octet) débute la partition n°1 précédemment créée.



### III. Système de fichiers

Nous allons maintenant déployer un système de fichier sur la partition précédemment créée. Le choix de la technologie du FS (*File System*) choisi peut conditionner les performances voire le bon fonctionnement du système ou du média. Par exemple, une clé USB sera probablement amenée à être utilisée sur machine supervisée par Windows comme par Linux. Windows étant une solution propriétaire et fondamentalement fermée, mieux vaut préférer une FS propriétaire comme **NTFS** ou **FAT** afin d'éviter toute mauvaise surprise et une bonne compatibilité à l'usage. Sur système embarqué (MCU), préférer des FS légers, mûrs et offrant du support comme **FAT** par exemple. Sur ordinateur ou système embarqué (SoC AP) supervisé par Linux et couplé au réseau, préférer par exemple **Btrfs**.



Déployez un système de fichiers **Virtual FAT** (extension à FAT12, FAT16 et FAT32) sur la partition n°1 et nommez cette partition à l'aide d'un label (option **-n**). Ce label sera à l'avenir utilisé par le système pour nommer les futurs points de montage.

```
lsblk -f
sudo mkfs.vfat -n root ${DISK}1
lsblk -f
```

Observez l'implémentation technologique **Virtual FAT** du système de fichiers à l'adresse de début de la partition. Retrouvez votre label ?

```
sudo dd if=${DISK} bs=1K count=2K of=./fs_fat32.bin
xxd ./fs_fat32.bin > fs_fat32.txt
```

### IV. Point de montage

Un **point de montage** est un répertoire dans le FS de la machine *host* (ordinateur) représentant l'image logique du contenu du média de stockage de masse externe. Sous Unix, les points de montage sont généralement présents dans les répertoires racine `/mnt/` (point de montage manuel) et `/media/` (points de montage automatiques). En effet, contrairement à Windows qui considère les périphériques externes comme des lecteurs différenciés du lecteur principal, Linux traite les partitions et périphériques de stockage comme des fichiers. Rappelons que sous Unix, tout est fichier !

Créez un point de montage et respectez le nom du label présent dans la partition n°1 précédemment créée (néanmoins, ce nom pourrait être différent). Vérifiez et validez avant de réaliser l'opération, le chemin relatif à votre nom d'utilisateur dans `/media/`. Validez les opérations réalisées.

```
export MEDIA=/media/<user_name>
sudo mkdir -p ${MEDIA}/root/
sudo mount ${DISK}1 ${MEDIA}/root/
lsblk -f
```

Réalisez une écriture sur le point de montage, ...

```
echo "Hello World Bro's !" > hello.txt
sudo cp hello.txt ${MEDIA}/root/
```

... synchronisez point de montage et support physique, ...

```
sync
```

... puis retirez manuellement du système le point de montage.

```
sudo umount ${MEDIA}/root
```

Vous pouvez alors retirer physiquement la clé USB de la machine et valider son bon fonctionnement sur un autre ordinateur ! Ne pas oublier l'étape de synchronisation, sinon l'écriture sur le média cible ne sera pas active.



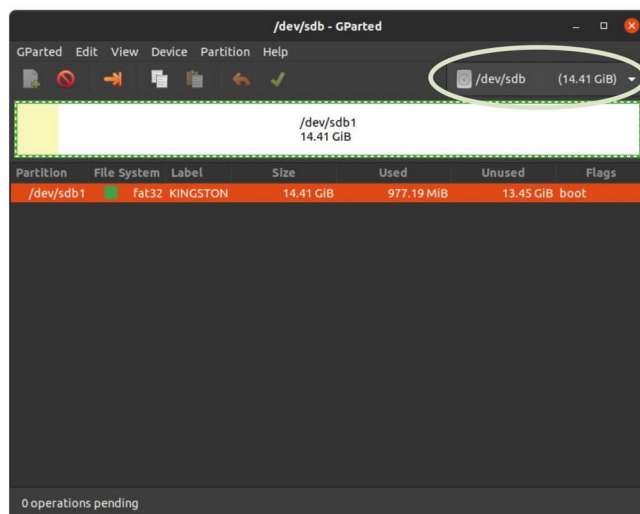
### V. Outil graphique GParted

Ce n'est pas parce qu'on travaille sous Linux qu'on doit tout faire à la console, dans l'ombre en sentant le café et les cacahuètes (ne nous jugez pas). Tout ce que nous avons fait ici est réalisable avec un outil graphique, automatisant les étapes précédemment présentées (création de la table des partitions et déploiement d'un système de fichiers). Nous utiliserons **GParted** (*GNOME Partition Editor*) basé sur GNU Parted, l'un des outils graphiques les plus standards sur système GNU/Linux.

Ouvrez GParted.

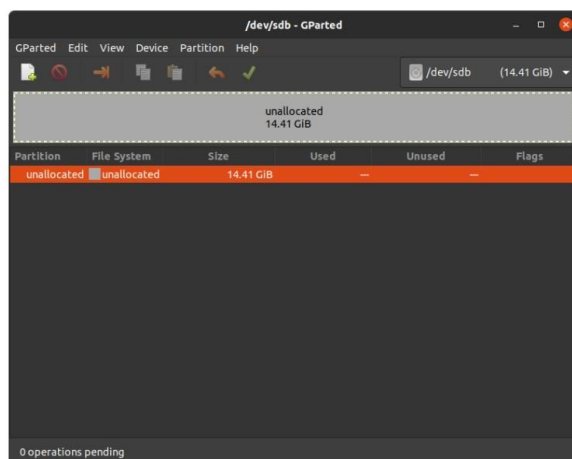
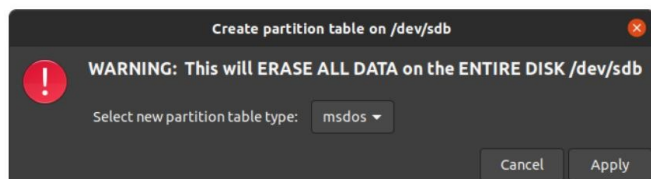
Sélectionnez le périphérique matériel à partitionner et retirez le point de montage existant :

- GParted > Refresh Devices
- GParted > Devices > /dev/sdX (*your device name*)
- Clic droit sur la partition du device (Partition > /dev/sdb1 - ci-dessous) > Unmount



Créez une table de partitions MSDOS (technologie MBR) :

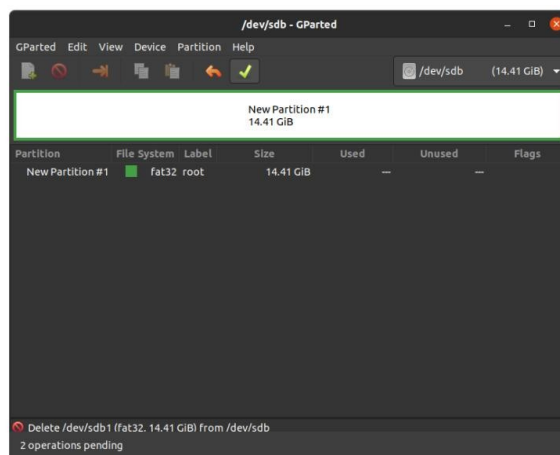
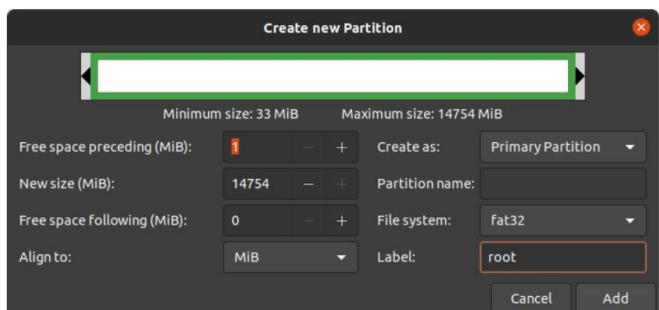
- Device > Create Partition Table...
- msdos > Apply





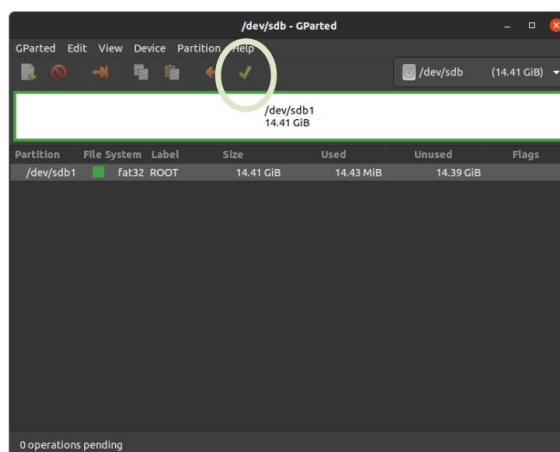
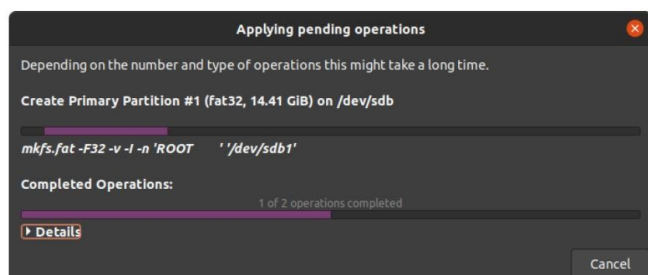
Créez une nouvelle partition :

- Partition > New
- Valeurs par défaut sauf les champs suivants
  - File system : > fat32
  - Label : > root
- Add



Appliquez les configurations précédentes :

- Cliquez sur l'icône "Apply all operations" entourée ci-dessous
- Et c'est terminé !



### VI. Kali sur clé USB bootable



Pour celles et ceux arrivés jusqu'ici, cet ultime exercice permet de déployer une image disque ISO sur une clé USB afin de la rendre *bootable* au démarrage et donc de charger dans notre cas au *boot* (phase d'amorçage) un système Kali Linux<sup>28</sup> dédié à la pénétration des systèmes.

Une image disque est historiquement une archive correspondant à la copie conforme d'un disque optique ou magnétique. Le format le plus répandu à notre époque est ISO (norme ISO 9660), même si d'autres standards existent (ISZ, IMG, UIF, ...).

Téléchargez l'ISO d'un Kali Linux Light 64-bit : <https://www.kali.org/downloads/>

À ce stade de l'enseignement, vous devez être apte à comprendre le tutoriel proposé sur le site officiel Kali afin de préparer un clé USB bootable :

```
sudo sfdisk -l

export DISK=/dev/<your_device_name>

dd if=kali-linux-<your_version>.iso of=${DISK} bs=4M
```

Une fois l'installation réalisée, redémarrez votre ordinateur en interrompant la phase de *boot* à l'amorçage (appuyez sur **F12** sur ordinateur en salles A203/A201 ou **F10**, **F2**, ... cela dépend du fabricant de carte mère), spécifiez que vous souhaitez démarrer sur un support USB et un Kali Linux va se démarrer en quelques secondes.

<sup>28</sup> <https://www.kali.org/>





