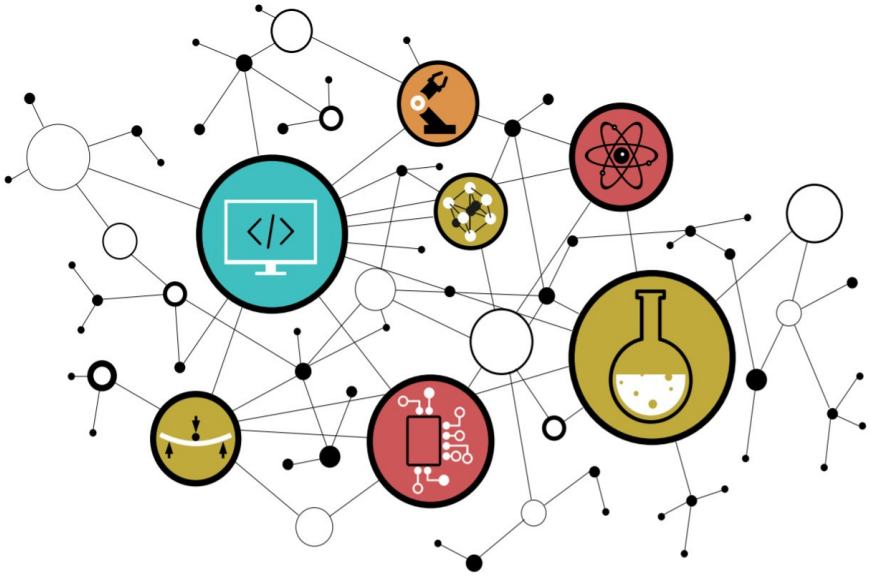
TRAVAUX PRATIQUES

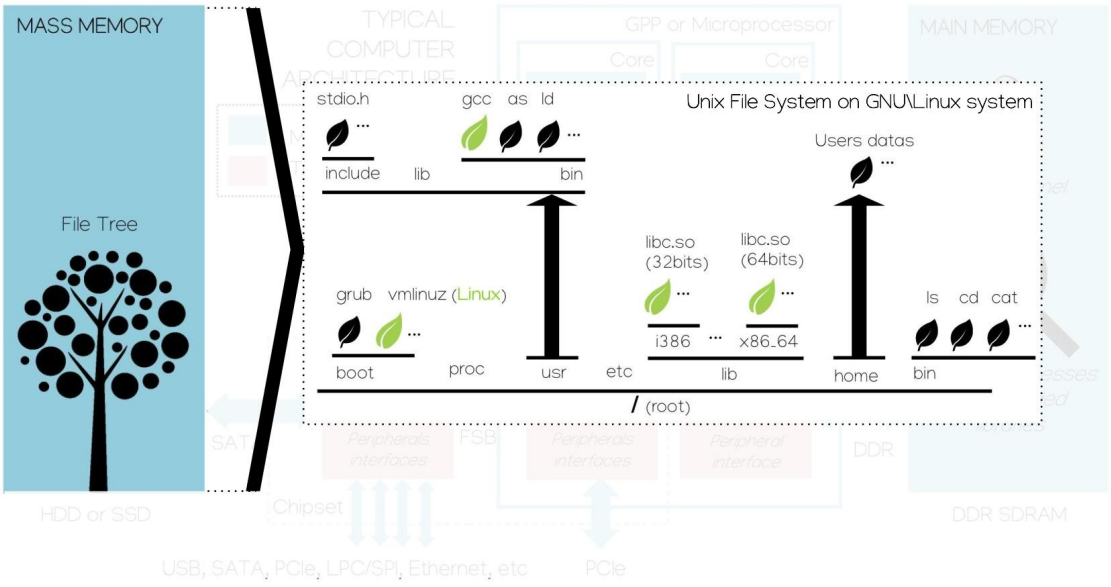
MÉMOIRE DE MASSE ET SYSTÈME DE FICHIERS



Hugo Descoubes – neku – GNU\Linux 5.8 - Ubuntu 20.04 LTS – LibreOffice 6.4.6.2 - 2021

SOMMAIRE

9. MÉMOIRE DE MASSE ET SYSTÈME DE FICHIERS



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 (opération 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. :

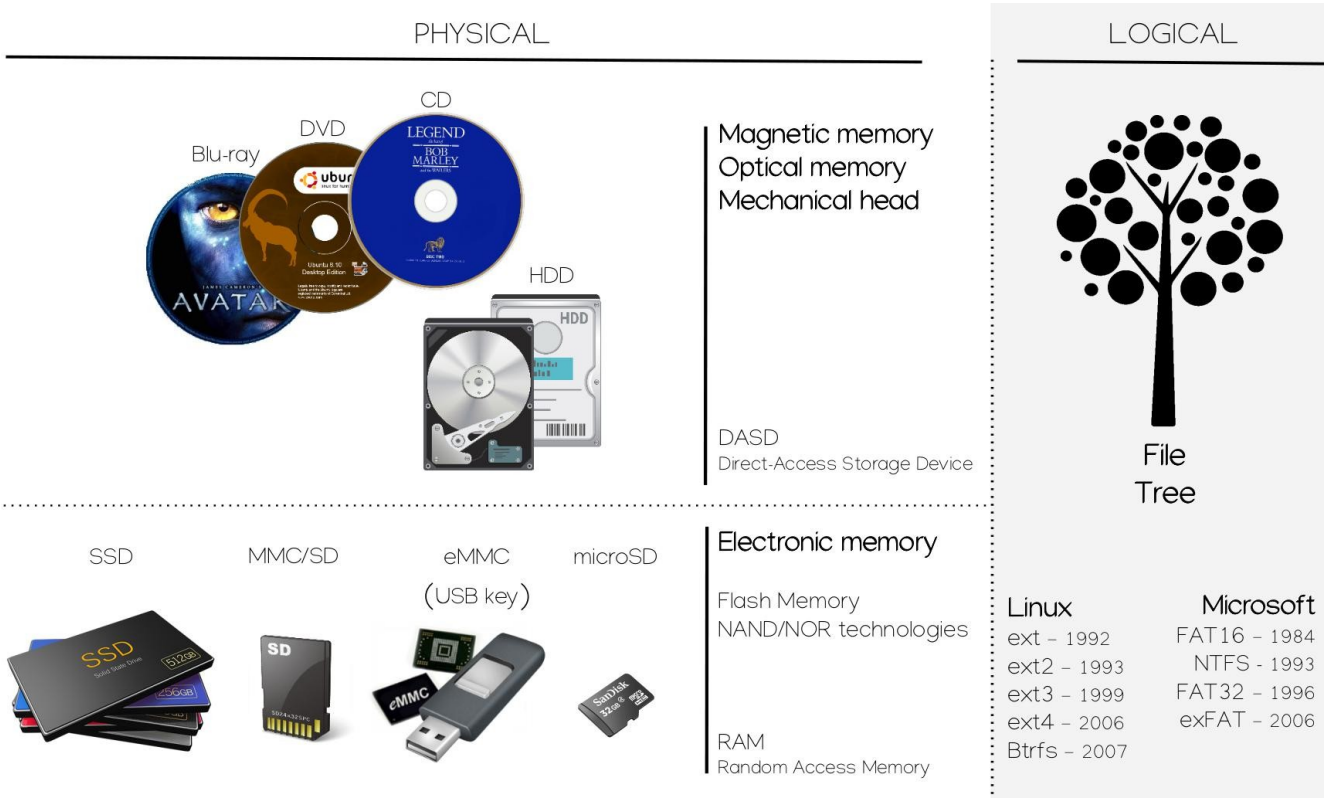
- Mémoire cache : *mémoire adressable par association (associative) de technologie SRAM*. A chaque octet copié en cache (niveaux L1\L2\L3) depuis la mémoire principale est associé un emplacement (par indexage) dans une ligne de cache.
- Mémoire principale : *mémoire adressable par octet de technologie DRAM*. Chaque octet possède une adresse mémoire unique typiquement représentée au format hexadécimal (32bits, 64bits, exemple 39bits physical et 48bits virtual, etc – *cat /proc/cpuinfo*). Sur processeur intégrant une MMU (GPP, AP, etc) 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, etc*. Cette mémoire stock 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, notamment :

- Linux : firmware brut de qqMo dans */boot* . Vous pouvez vérifier et constater que Linux n'est pas un fichier ELF, mais sait exploiter des fichiers ELF une fois chargé et exécuté en mémoire principale.
- GCC : Chaîne de compilation ayant servi à générer le système lui-même dans */usr/bin*
- LIBC : Bibliothèques standards du langage C (32bits et 64bits) dans */lib*

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 !

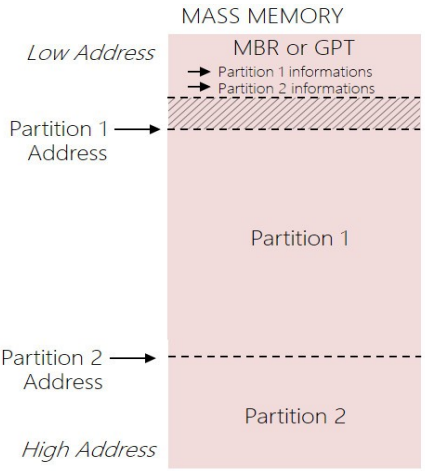


Plusieurs technologies de mémoire de masse sont actuellement en usage (HDD, SSD, MMC SD, MMC microSD, CD, DVD, Blu-ray, etc), tout comme certaines sont maintenant tombées en désuétude (K7 audio, disquette, VHS, VHS-c, carte perforée, tore magnétique, etc). Chaque technologie offre son lot hérité d'avantages, d'inconvénients, de compromis et se trouve adapté 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 (SSD, MMC SD, MMC microSD, eMMC par exemple sur clé USB, etc)

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, etc sous Linux. NTFS, FAT32, VFAT, etc sous Windows. Bien d'autres technologies existent, notamment pour d'autres systèmes d'exploitation (BSD, Mac OS X, etc). Chaque technologie offre sont 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 4Go
- Taille maximale théorique d'une partition 16To
- Nombre maximal de fichiers 268M
- Nombre maximal de fichiers par répertoire 65534
- etc

9.1. Table des partitions



Une partition représente une zone logique contigu de la mémoire physique. Pour un support donné et en fonction des besoins, il est bien entendu 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 dominent le marché :

- MBR (Master Boot Record, https://doc.ubuntu-fr.org/mbr). Nous ne pouvons créer que 4 partitions primaires maximum, toutes de tailles inférieures à 2To. Technologie encore très utilisée à notre époque, notamment dans les Systèmes Embarqués et ordinateur personnel

- 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 !***

- Se placer dans le répertoire */disco/mass/* puis connecter la clé USB à votre machine. Identifier son nom (*/dev/sdx*) et ses caractéristiques.

```
Lsblk

lsblk -f

sudo sfdisk -l

export DISK=/dev/<your_device_name>
```

- Identifier le paramètre système *block size* relatif à votre support (valeur dépendant de multiples paramètres), puis effacer l'ancienne table des partitions. Analyser 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 ?

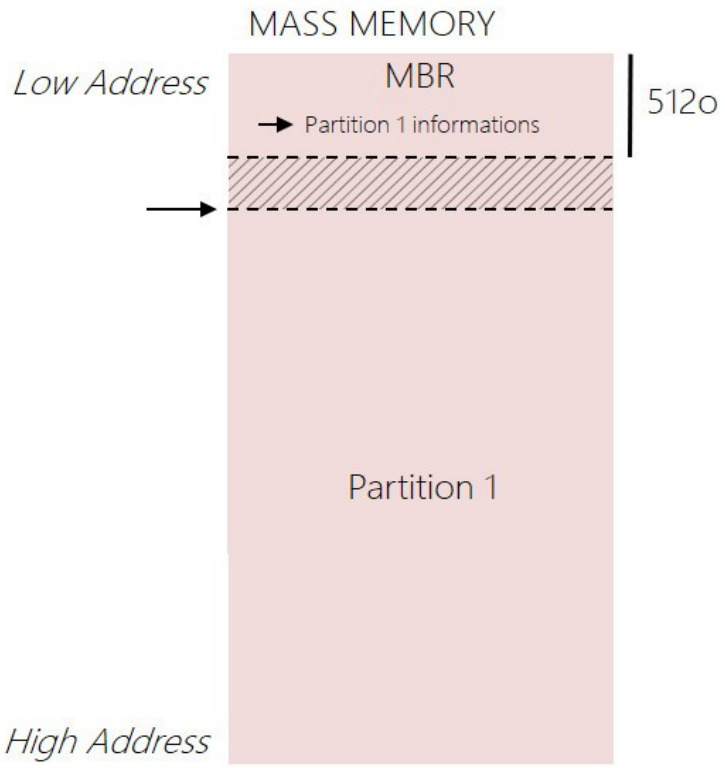
- Nous allons créer un nouvelle table des partitions en utilisant l'utilitaire *sfdisk*. Néanmoins, d'autres utilitaires existent pour créer des tables des partitions (*parted, fdisk, etc*). Créer une nouvelle table des partitions et analyser la sortie.

```
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
```

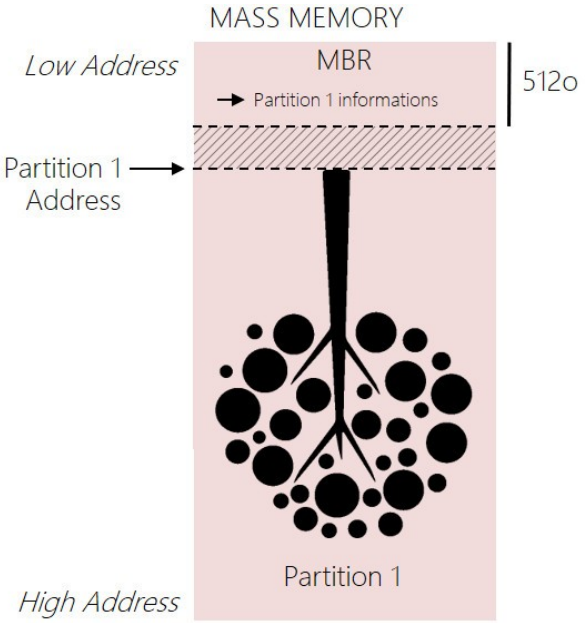
- En s'aidant d'internet, préciser la taille d'un MBR ? Par quelle suite d'octets se termine toujours un MBR ? Vérifier que cette suite binaire est bien présente après création de la table des partitions.

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

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



9.2. 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 peu 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, mûres et offrant du support comme FAT par exemple. Sur ordinateur ou système embarqué (SoC AP ou System on Chip Application Processor) supervisé par Linux et couplé au réseau, préférer par exemple Btrfs, la quintessence des FS sous Linux (ext2, ext3, ext4 étant des technologies transitoires).

- Déployer un système de fichiers Virtual FAT (extension à FAT12, FAT16 et FAT32) sur la partition n°1 et nommer 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
```

- Observer l'implémentation technologique Virtual FAT du système de fichiers à l'adresse de début de la partition. Retrouver votre label ?

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

xxd ./fs_fat32.bin > fs_fat32.txt
```

9.3. 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 (clé USB, MMC SD, etc). 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 !

- Créer un point de montage et respecter le nom du label présent dans la partition n°1 précédemment créée (néanmoins, ce nom pourrait être différent). Vérifier et valider avant de réaliser l'opération, le chemin relatif à votre nom d'utilisateur dans */media* . Valider les opérations réalisées.

```
export MEDIA=/media/<user_name>

sudo mkdir -p ${MEDIA}/root/

sudo mount ${DISK}1 ${MEDIA}/root/

lsblk -f
```

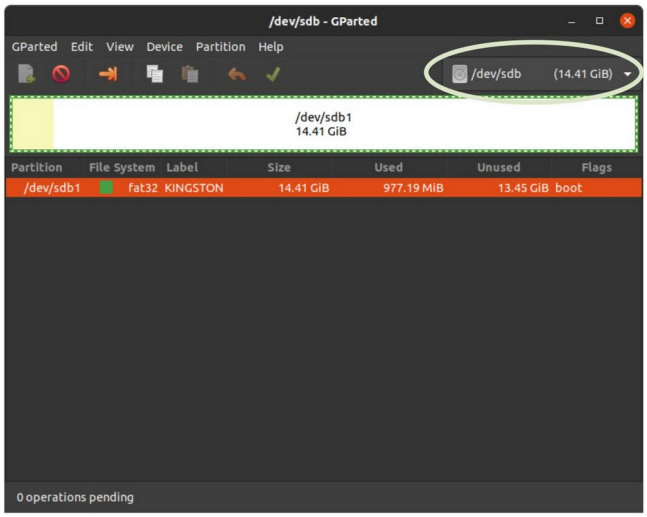
- Réaliser une écriture sur le point de montage, synchroniser point de montage et support physique puis retirer manuellement du système le point de montage. Vous pouvez alors retirer physiquement la clé USB de la machine et valider son bon fonctionnement sur un autre ordinateur ! *Ne pas oublier l'étape de synchronisation, sinon l'écriture sur le média cible ne sera pas active*.

```
echo "Hello World Bro's !" > hello.txt

sudo cp hello.txt ${MEDIA}/root/

sync

sudo umount ${MEDIA}/root
```

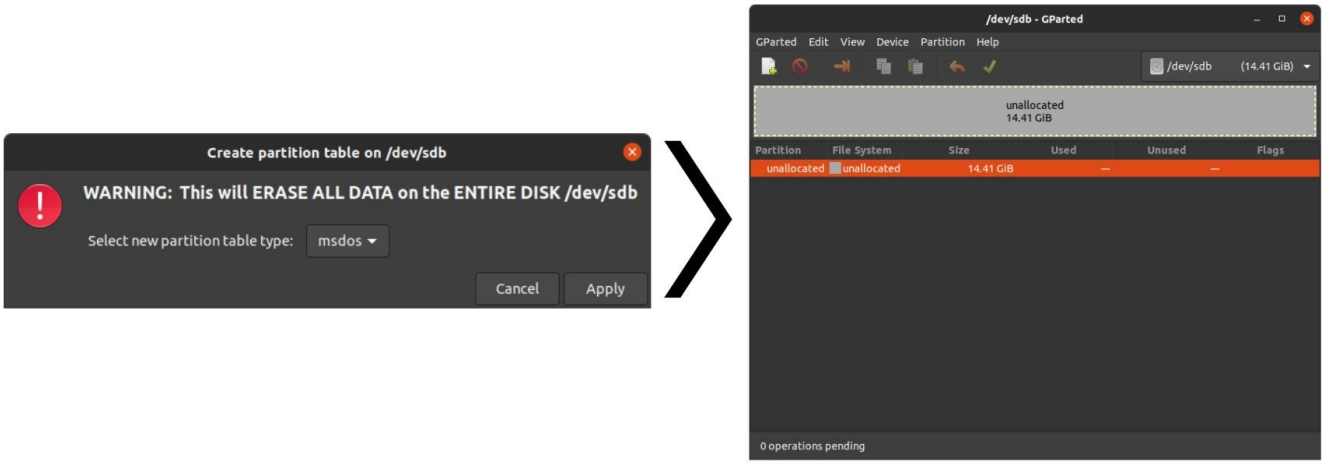
9.4. Outil graphique GParted

Nous allons finaliser l'exercice avec l'utilisation d'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 graphique les plus standard sur système GNU\Linux.

- Sélectionner le périphérique matériel à partitionner et retirer le point de montage existant
    - GParted > Refresh Devices
    - GParted > Devices > /dev/sd? (your device name)
    - Clic droit sur la partition du device (Partition > /dev/sdb1 - ci-dessous) > Unmount

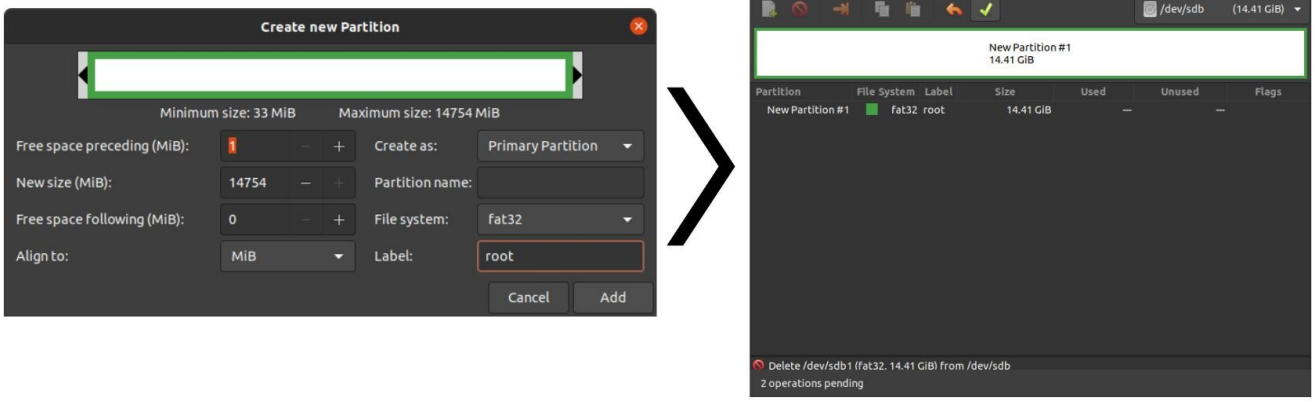


- Créer une table de partitions MSDOS (technologie MBR)
    - Device > Create Partition Table…
    - msdos > Apply



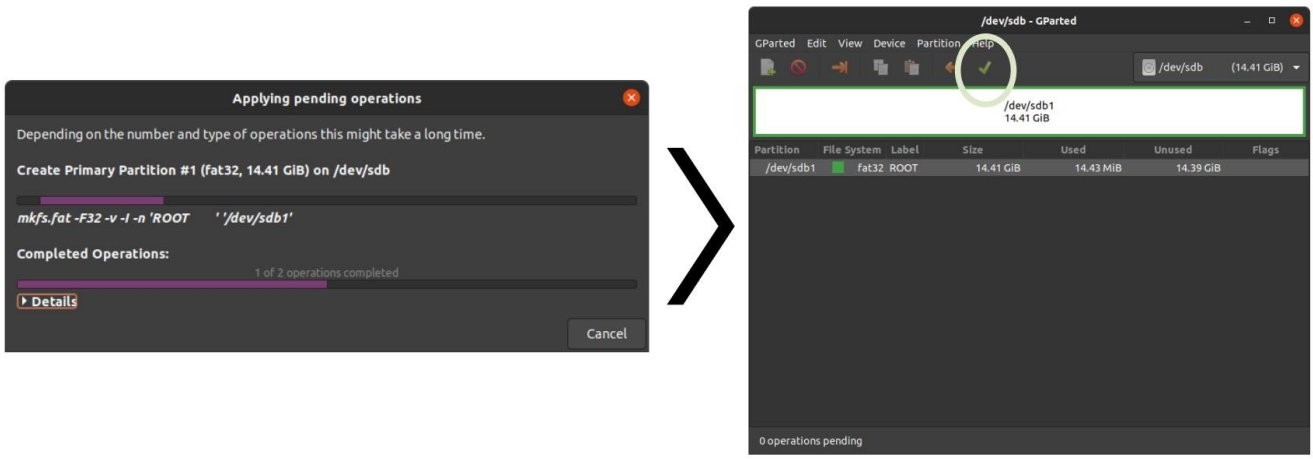
- Créer une nouvelle partition

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



- Appliquer les configurations précédentes

    - Cliquer sur l'icône ''Apply all operations'' entouré ci-dessous
    - Et c'est terminé !



9.5. Kali sur clé USB bootable

Pour celles et ceux étant 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 (https://www.kali.org/) 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, etc).

- Télécharger l'ISO d'un Kali Linux Light 64bits :

  https://www.kali.org/downloads/

- A 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, redémarrer votre ordinateur en interrompant la phase de boot à l'amorçage (appuyer sur F12 sur ordinateur en salles A203/A201 ou F10,F2, etc dépend du fabricant de carte mère), spécifier que vous souhaitez booter (démarrer) sur un support USB et un Kali Linux va se démarrer en quelques seconde

Et voilà, les TP sont terminés !


```
neku$ Pour conclure sur quelques mots bien personnels ...
neku$ Je dirais longue vie à Linux, au projet GNU, aux systèmes Unix ...
neku$ A l'open source, au travail collaboratif décentralisé ...
neku$ A l'agilité dans le travail en équipe et dans les entreprises ...
neku$ A la liberté de s'informer et de partager l'information ...
neku$ A vous et à vos rêves …
neku$ Que nous ne cessions jamais d'œuvrer pour la liberté de tous ...
neku$ My name is nobody
neku$
```