

Mode opératoire activité 4 :

Matériel et équipement :

Installation de la Machine Virtuelle sur la machine :

Dell G15 5510	
Nom de l'appareil	DESKTOP-T7J0I15
Processeur	Intel(R) Core(TM) i7-10870H CPU @ 2.20GHz 2.21 GHz
Mémoire RAM installée	16,0 Go (15,8 Go utilisable)
Type du système	Système d'exploitation 64 bits, processeur x64

Un seul disque dur non fragmenté de 500 Go.

Chipset : HM 470



Note : Comme pour les autres activités, produire un mode opératoire avec vos réponses aux questions posées, les commandes systèmes utilisées, etc ...,

1. Montage et exploitation d'un dispositif USB par le système invité

- 1.1 Les périphériques de stockage:

Pour simplifier, il existe quatre types de périphériques de stockage :

1. les périphériques internes au boîtier (de type disque dur)
2. les périphériques amovibles avec lecteurs/enregistreurs internes (CD/DVD)
3. les périphériques externes amovibles (généralement USB) : c'est le cas d'une clé usb de stockage, mais de la souris, du clavier, aussi des lecteurs/enregistreurs externes, ...

4. les volumes distants montés via le réseau grâce au protocole NFS ; c'est le cas de votre compte situé dans le disque du serveur grappa (alias di-homes).
5. Il faudrait ajouter ici ce qu'on appelle "le cloud", qui vous permet de stocker des informations ailleurs, dans des centres de stockage de données (data-centers) via Internet (dropbox, etc.).

Les systèmes de fichiers des périphériques de type disque dur interne et généralement les média amovibles des lecteurs IDE sont montés au démarrage selon les consignes données dans le fichier **/etc/fstab**

-Afficher le contenu de /etc/fstab dans le terminal (Donner la commande utilisée)

```
debian@debian:~$ ls -la /etc/fstab
-rw-r--r-- 1 root root 876 22 oct. 07:01 /etc/fstab
debian@debian:~$
```

-L'usb est il pris en compte ?

Non elle n'est pas prise en compte

- 1.2 USB:

USB = Universal Serial Bus

Les ports USB sont actuellement massivement utilisés pour connecter toute sorte de périphériques qui peuvent se brancher "à chaud", sans risque . La transmission des données est plus rapide sur un port USB. Un port USB peut recevoir un hub USB, qui permet de brancher plusieurs périph. sur un seul port ; mais les besoins en énergie du port sont plus élevés, et il faut y prendre garde.

Ainsi les ports USB ont la plupart du temps remplacé les ports série et parallèle. mais ces périphériques USB-série et USB-parallèle renferment un convertisseur USB-série ou un convertisseur USB-parallèle, ils fonctionnent en interne avec les caractéristiques électroniques et informatiques d'un périphérique série ou parallèle. Pour débrancher à chaud, il est préférable de s'assurer qu'aucun flux de données ne circule entre la machine et le périphérique. Pour les périphériques de stockage, il est prudent et recommandé de "démonter" le périphérique, c'est à dire de l'ôter de l'arborescence des fichiers, avant de le déconnecter physiquement.

Un des autres avantages de l'USB est que le périphérique est reconnu et est pris en charge automatiquement dès qu'il est branché (sans intervention de l'utilisateur).

Cette automaticité n'est pas magique ; elle tient d'une part à la technologie matérielle USB, d'autre part au paramétrage d'un fichier de correspondance qui associe à la détection du matériel particulier l'action précise à entreprendre. Ainsi, quand on connecte une clé de stockage USB, les pilotes sont chargés, le volume est monté automatiquement, et son contenu est automatiquement affiché (Linux), ou encore une liste de tâches est proposée à l'utilisateur (Windows).

Dans la plupart des distributions Linux récentes, la clé de stockage est montée dans un sous-répertoire du répertoire situé dans la racine : **/media** . Ce sous répertoire porte le nom de la clé.

-Que contient actuellement /media ?

```
debian@debian:~$ ls -la /media
total 32
drwxr-xr-x  8 root root  4096 28 oct.  08:27 .
drwxr-xr-x 18 root root  4096 27 oct.  15:05 ..
lrwxrwxrwx  1 root root    6 22 oct.  07:01 cdrom -> cdrom0
drwxr-xr-x  2 root root  4096 22 oct.  07:01 cdrom0
drwxr-xr-x  2 root root  4096 22 oct.  07:01 cdrom1
drwxr-x---  3 root root  4096 28 oct.  11:17 debian
drwxr-x---  2 root root  4096 28 oct.  07:41 sf_Cours_IUT
drwxrwx---  1 root vboxsf 4096 28 oct.  08:33 sf_Travail
drwxr-x---  2 root root  4096 28 oct.  07:46 sf_untest
debian@debian:~$
```

Sous linux, les règles de fabrication automatique des fichiers de périphériques sont rendues possibles depuis l'avènement du noyau 2.6 par le démon udev et le répertoire associé /sys présent dans l'espace utilisateur (dans les noyaux 2.4, sysdev était incorporé au noyau). sys est le nom d'un répertoire de la racine de l'arborescence qui contient les informations sur les périphériques détectés "à chaud", "hotplug". l'utilisateur voir le contenu du répertoire /sys. Nous nous contentons ici d'appliquer les règles par défaut, mais sachez que l'utilisateur peut énoncer des règles pour donner des noms personnalisés différents aux fichiers de périphériques.

-Dans quel sous répertoire de /sys se trouve les périphériques de stockage ?

```
debian@debian:~$ ls -la /sys/
total 4
dr-xr-xr-x 13 root root  0 28 oct.  09:10 .
drwxr-xr-x 18 root root 4096 27 oct.  15:05 ..
drwxr-xr-x  2 root root  0 28 oct.  09:10 block
drwxr-xr-x 34 root root  0 28 oct.  09:10 bus
drwxr-xr-x 52 root root  0 28 oct.  09:10 class
drwxr-xr-x  4 root root  0 28 oct.  09:10 dev
drwxr-xr-x 15 root root  0 28 oct.  09:10 devices
drwxr-xr-x  5 root root  0 28 oct.  09:10 firmware
drwxr-xr-x  7 root root  0 28 oct.  09:10 fs
drwxr-xr-x  2 root root  0 28 oct.  11:21 hypervisor
drwxr-xr-x 15 root root  0 28 oct.  09:10 kernel
drwxr-xr-x 146 root root  0 28 oct.  09:10 module
drwxr-xr-x  3 root root  0 28 oct.  09:10 power
debian@debian:~$ ls -la /sys/^C
debian@debian:~$
```

Remarque importante : ne confondez pas liaison série RS232 et liaison USB (le mot Bus de USB est important): si vous essayez de relier deux ports usb de deux machines par un simple cordon de raccordement usb-usb (priseA-priseA), vous risquez fort de griller une partie de l'usb sur une des deux machines.

Si vous voulez vraiment le faire, il faut un cordon spécial qui contient un circuit en son milieu, ce circuit permet la conversion : signaux usb <-> signaux ethernet<->signaux usb.

La technologie de communication est alors celle d'un réseau ethernet "au-dessus de usb" (ethernet over usb), et la liaison se configure comme celle d'un réseau point à point.

- 1.3 inspection de la configuration du noyau:

- Vérifiez l'intégration des pilotes USB dans le noyau en cherchant la chaîne CONFIG_USB dans le fichier de configuration du noyau `boot/config-$(uname -r)`

```
cat /boot/config-$(uname -r) | grep 'CONFIG_USB'
```

```
CONFIG_USB_FUNCTIONFS_GENERIC=y
# CONFIG_USB_MASS_STORAGE is not set
# CONFIG_USB_GADGET_TARGET is not set
CONFIG_USB_G_SERIAL=m
# CONFIG_USB_MIDI_GADGET is not set
# CONFIG_USB_G_PRINTER is not set
# CONFIG_USB_CDC_COMPOSITE is not set
# CONFIG_USB_G_NOKIA is not set
# CONFIG_USB_G_ACM_MS is not set
# CONFIG_USB_G_MULTI is not set
# CONFIG_USB_G_HID is not set
# CONFIG_USB_G_DBGP is not set
# CONFIG_USB_G_WEBCAM is not set
# CONFIG_USB_RAW_GADGET is not set
CONFIG_USB_ROLE_SWITCH=m
CONFIG_USB_ROLES_INTEL_XHCI=m
# CONFIG_USB_LGM_PHY is not set
CONFIG_USB4=m
# CONFIG_USB4_DEBUGFS_WRITE is not set
debian@debian:~$
```

Y : Le module est résident dans le noyau

M : Le module est chargé par le noyau à la demande (amovible)

- L'USB1,2 et 3 sont ils pris en charge par le noyau (chercher les pilotes correspondants, voir l'activité 3 pour le nom des pilotes)

USB 1 :

```
debian@debian:~$ sudo cat /boot/config-$(uname -r) | grep 'CONFIG_USB_OHCI'
CONFIG_USB_OHCI_LITTLE_ENDIAN=y
CONFIG_USB_OHCI_HCD=m
CONFIG_USB_OHCI_HCD_PCI=m
# CONFIG_USB_OHCI_HCD_SSB is not set
# CONFIG_USB_OHCI_HCD_PLATFORM is not set
debian@debian:~$
```

USB 2 :

```

debian@debian:~$ sudo cat /boot/config-$(uname -r) | grep 'CONFIG_USB_EHCI'
CONFIG_USB_EHCI_HCD=m
CONFIG_USB_EHCI_ROOT_HUB_TT=y
CONFIG_USB_EHCI_TT_NEWSCHED=y
CONFIG_USB_EHCI_PCI=m
# CONFIG_USB_EHCI_FSL is not set
# CONFIG_USB_EHCI_HCD_PLATFORM is not set
debian@debian:~$

```

USB 3 :

```

debian@debian:~$ sudo cat /boot/config-$(uname -r) | grep 'CONFIG_USB_xHCI'
debian@debian:~$

```

- Rechercher aussi la chaîne CONFIG_USB_STORAGE (périphérique USB de stockage).

```

debian@debian:~$ sudo cat /boot/config-$(uname -r) | grep 'CONFIG_USB_STORAGE'
CONFIG_USB_STORAGE=m
# CONFIG_USB_STORAGE_DEBUG is not set
CONFIG_USB_STORAGE_REALTEK=m
CONFIG_USB_STORAGE_DATAFAB=m
CONFIG_USB_STORAGE_FREECOM=m
CONFIG_USB_STORAGE_ISD200=m
CONFIG_USB_STORAGE_USBAT=m
CONFIG_USB_STORAGE_SDDR09=m
CONFIG_USB_STORAGE_SDDR55=m
CONFIG_USB_STORAGE_JUMPSHOT=m
CONFIG_USB_STORAGE_ALAUDA=m
CONFIG_USB_STORAGE_ONETOUCH=m
CONFIG_USB_STORAGE_KARMA=m
CONFIG_USB_STORAGE_CYPRESS_ATACB=m
CONFIG_USB_STORAGE_ENE_U6250=m
debian@debian:~$

```

- Quel est le statut de ce module vis-à-vis du noyau, est-il résident, amovible, ou jamais présent ?

[Le module est amovible dans le noyau, il est chargé par le noyau à la demande](#)

- 1.4 clé usb de stockage:

- Que fait la commande dmesg ? (man dmesg)

```

NOM
    dmesg - Afficher et contrôler le tampon circulaire du noyau

SYNOPSIS
    dmesg [options]

    dmesg --clear
    dmesg --read-clear [options]
    dmesg --console-level niveau
    dmesg --console-on
    dmesg --console-off

DESCRIPTION
    dmesg est utilisé pour examiner ou contrôler le tampon circulaire du
    noyau.

    L'action par défaut est d'afficher tous les messages du tampon circu-
    laire du noyau.

```

- Afficher les 20 dernières lignes (man tail)

sudo dmesg > tampnoy.txt

tail -n 20 tampnoy.txt


```

debian@debian:~$ ls -la /media/
total 32
drwxr-xr-x  8 root root  4096 28 oct.  08:27 .
drwxr-xr-x 18 root root  4096 27 oct.  15:05 ..
lrwxrwxrwx  1 root root    6 22 oct.  07:01 cdrom -> cdrom0
drwxr-xr-x  2 root root  4096 22 oct.  07:01 cdrom0
drwxr-xr-x  2 root root  4096 22 oct.  07:01 cdrom1
drwxr-xr-x  2 root root  4096 28 oct.  12:05 debian
drwxr-xr-x  2 root root  4096 28 oct.  07:41 sf_Cours_IUT
drwxrwxr-x  1 root vboxsf 4096 28 oct.  08:33 sf_Travail
drwxr-xr-x  2 root root  4096 28 oct.  07:46 sf_untest
debian@debian:~$

```

- Lire les fichiers de périphérique associés au volume correspondant à la clé usb (afficher pour cela le contenu du dossier /dev/... à ajuster avec le nom de la clé)
`sudo cat /dev/sdb`

2. Système de fichiers et formatage d'une clé USB (en local)

- 2.1 Introduction

Le but est de formater une clé, puis de partitionner et formater comme on le souhaite la ou les partitions effectuées.

Il faut être sudoer pour faire cela ...

Dans un premier temps, on va utiliser le formatage en FAT qui est économe en terme d'emplacement mémoire et adaptée aux petites capacités de stockage, et compatible Windows.

Le système de fichiers FAT est donc recommandé pour une clé usb. Quand on achète une nouvelle clé usb, elle est formatée en fat.

Ce système de fichier possède cependant des inconvénients : pas de journalisation, une faible robustesse (c'est pourquoi on stocke automatiquement par sécurité deux exemplaires de la FAT, car si la table d'allocation des enregistrements est endommagée on ne peut plus récupérer les enregistrements). De plus, sous Linux, on ne peut créer un lien symbolique avec un fichier appartenant à ce système de fichiers.

Sous Windows c'est NTFS= New Technology File System, le Système de fichiers de windows NT (ne pas confondre avec NFS = Network File System, le protocole de montage des fichiers sur machine distantes), sous Linux c'était ext2, mais c'est maintenant ext3 ou mieux ext4 (plus performant, extension de capacité de ext3), qui sont nativement utilisés pour des disques durs, ou des disques usb de plus grande capacité. Sur les CD/ROM en lecture seulement, le système de fichiers est différent, puisque tous les enregistrements d'un fichiers sont adjacents sur le disque (iso9600).

Remarque : les manip sont les mêmes pour formater un disque de stockage USB de grande capacité, mais les systèmes de fichiers des partitions sont bien sûr différents de la FAT.

- Vérifier que le paquet **dosfstools** est installé. Il charge les commandes pour formater en FAT= File Allocation Table) (FAT veut dire table d'allocation des fichiers).

Indication : utiliser la commande apt (man apt) avec l'option appropriée.

```

debian@debian:~$ sudo apt install dosfstool
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
E: Impossible de trouver le paquet dosfstool
debian@debian:~$ █

```

- 2.2 Examen de la table des partitions, destruction de la partition existante et repartitionnement avec reformatage

Un périphérique de masse est muni d'une table de partitions située dans le premier secteur du périphérique de stockage.

Ce secteur est appelé MBR : Master Boot Record.

Ce secteur contient un petit bout de code écrit en assembleur (qui sera exécuté à la demande du BIOS seulement si la séquence de boot a pris en compte le périphérique), et des données sur la table de partition.

On peut aller jusqu'à 4 partitions par unité de stockage, avec au maximum une partition étendue (elle même partitionnable). Les autres partitions que la partition étendue s'appellent des partitions primaires. Mais dans une clé usb de données, on trouve habituellement une seule partition primaire en FAT32, et pas de partition étendue.

Chaque partition doit être formatée selon un système de fichier bien précis.

Lorsqu'une des partitions est amorçable (bootable), elle contient un système d'exploitation dans une partition, un drapeau (flag) l'indique dans le code du MBR.

Cependant, une partition n'est pas forcément bootable, elle peut ne contenir que des données, et non pas un système d'exploitation à charger.

C'est le cas ici : la clé USB qui vous est remise a pour capacité 4 Go ou 8 Go,

Nous allons nous servir de la commande **fdisk** pour détruire/recréer la table de partition (dans le MBR) en jouant sur le fichier /dev/sd? (? = a, b, c ... selon les cas)

Puis de la commande **mkfs.vfat** (mkfs=make file system, type de système de fichier : virtual fat) pour formater la ou les partitions établies en jouant sur le fichier /dev/sd?? (le second ? est le numéro de la partition dans /dev/sd?)

- Brancher une clé usb

- Utiliser fdisk (man fdisk) pour identifier la clé usb

Le nom de fichier de périphérique doit être exact, sinon on risque de détruire une autre partition, par exemple celle d'un disque dur !

En général, sur les stations, on trouve /dev/sdb (le fichier de périphérique qui permet de définir la table de partitions) et /dev/sdb1 (le fichier propre à la première et la seule partition) ; vérifier avant de faire la suite !

Nous avons changé de clé pour cette partie de l'activité


```

Disk /dev/sdb: 14,72 GiB, 15805479424 bytes, 30870077 sectors
Disk model: Ultra Line
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: dos
Disk identifier: 0x0a0dff65

Device      Boot      Start        End    Sectors    Size Id Type
/dev/sdb1             1869881445 3571221465 1701340021 811,3G 7a unknown
/dev/sdb2             1634566756 3553817813 1919251058 915,2G 72 unknown
/dev/sdb3              0              0          0      0B  0 Empty
/dev/sdb4             28049408    28049849      442    221K  0 Empty

Partition table entries are not in disk order.
debian@debian:~$

```

Le nom de fichier de périphérique doit être exact, sinon on risque de détruire une autre partition, par exemple celle d'un disque dur !

En général, sur les stations, on trouve /dev/sdb (le fichier de périphérique qui permet de définir la table de partitions) et /dev/sdb1 (le fichier propre à la première et la seule partition) ; vérifier avant de faire la suite !

- Lire la partition de la clé. Faire la différence entre la clé et un disque dur avec la taille. Notez le périphérique concerné : /dev/sd.

Quel est le système de fichier actuel de la clé ? Notez-le.

```

debian@debian:~$ sudo fdisk -l
Disk /dev/sda: 8 GiB, 8589934592 bytes, 16777216 sectors
Disk model: VBOX 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: dos
Disk identifier: 0x35613218

Device      Boot      Start        End    Sectors    Size Id Type
/dev/sda1    *           2048    14776319 14774272      7G 83 Linux
/dev/sda2             14778366 16775167 1996802    975M  5 Extended
/dev/sda5             14778368 16775167 1996800    975M 82 Linux swap / Solaris

Disk /dev/sdb: 14,72 GiB, 15805479424 bytes, 30870077 sectors
Disk model: Ultra Line
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: dos
Disk identifier: 0x0a0dff65

Device      Boot      Start        End    Sectors    Size Id Type
/dev/sdb1             1869881445 3571221465 1701340021 811,3G 7a unknown
/dev/sdb2             1634566756 3553817813 1919251058 915,2G 72 unknown
/dev/sdb3              0              0          0      0B  0 Empty
/dev/sdb4             28049408    28049849      442    221K  0 Empty

Partition table entries are not in disk order.
debian@debian:~$

```

Le disque sda est celui du disque de Debian et le sdb est la clé USB

Le système de fichier est dos

!!!!!! Attention à partitionner la clé, et pas un disque dur !!!!!

- 1) Se familiariser avec fdisk (man fdisk, recherche sur le net) pour détruire toutes les partitions sur la clé usb
- 2) Créer une nouvelle partition primaire 1 en fat 32 LBA

3) Enregistrer les modifications

```
Partition number (1-4, default 1): 1
First sector (2048-30870076, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-30870076, default 30870076): +5G

Created a new partition 1 of type 'Linux' and of size 5 GiB.

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.

debian@debian:~$ sudo fdisk -l
Disk /dev/sda: 8 GiB, 8589934592 bytes, 16777216 sectors
Disk model: VBOX 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: dos
Disk identifier: 0x35613218

Device      Boot    Start        End    Sectors    Size Id Type
/dev/sda1   *          2048    14776319    14774272     7G 83 Linux
/dev/sda2                14778366    16775167     1996802    975M  5 Extended
/dev/sda5                14778368    16775167     1996800    975M 82 Linux swap / Solaris

Disk /dev/sdb: 14,72 GiB, 15805479424 bytes, 30870077 sectors
Disk model: Ultra Line
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: dos
Disk identifier: 0xf2df868b

Device      Boot    Start        End    Sectors    Size Id Type
/dev/sdb1                2048    10487807    10485760     5G 83 Linux
debian@debian:~$
```

```
Command (m for help): t
Selected partition 1
Hex code or alias (type L to list all): 0c
Changed type of partition 'Empty' to 'W95 FAT32 (LBA)'.

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.

debian@debian:~$ sudo fdisk -l
Disk /dev/sda: 8 GiB, 8589934592 bytes, 16777216 sectors
Disk model: VBOX 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: dos
Disk identifier: 0x35613218

Device      Boot    Start        End    Sectors    Size Id Type
/dev/sda1   *          2048    14776319    14774272     7G 83 Linux
/dev/sda2                14778366    16775167     1996802    975M  5 Extended
/dev/sda5                14778368    16775167     1996800    975M 82 Linux swap / Solaris

Disk /dev/sdb: 14,72 GiB, 15805479424 bytes, 30870077 sectors
Disk model: Ultra Line
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: dos
Disk identifier: 0xf2df868b

Device      Boot    Start        End    Sectors    Size Id Type
/dev/sdb1                2048    10487807    10485760     5G  c W95 FAT32 (LBA)
debian@debian:~$
```

4) formater :

le système de fichier est fat32, LBA

- En utilisant la commande **mkfs.vfat** (man mkfs.vfat, recherche sur le net), reformater la clé en fat32 en l'appelant cle01:

Si un message dit que la clé est déjà montée, démonter la clé par exemple par action sur clic droit de l'icône sur le bureau, puis refaire la commande

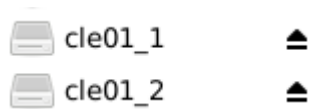
- enlever la clé
- réintroduire la clé
- enregistrer un fichier texte, le lire, utilisez la clé pour lire et écrire ...

```
debian@debian:~$ sudo mkfs.vfat -n cle01 /dev/sdb1
mkfs.fat 4.2 (2021-01-31)
mkfs.fat: Warning: lowercase labels might not work properly on some systems
debian@debian:~$ sudo fdisk -l
```

- 2.3 Création de plusieurs partitions sur une clé

- détruire la partition précédente, partitionner la clé en deux partitions primaires FAT 32 (LBA) à peu près égales et enregistrer.

- Formater chacune des partitions avec le système fat32. Nom de la première partition : cle01.1 et pour la deuxième cle01.2



- Peut-on monter ou démonter les deux partitions ? Le faire et voir le résultat ...

[Si on démonte 1 partition, l'autre est également démontée](#)

- Remettre enfin la clé dans son état initial : une partition unique fat32.

Remarque : il existe aussi la commande plus récente cfdisk, en mode pseudo-graphique.

- 2.4 Ajout d'une partition NTFS (Windows) et EXT4 (Linux) à la partition FAT32 de la clé01

- une partition fat32, nom cle01.1

- une partition NTFS, nom cle01.2

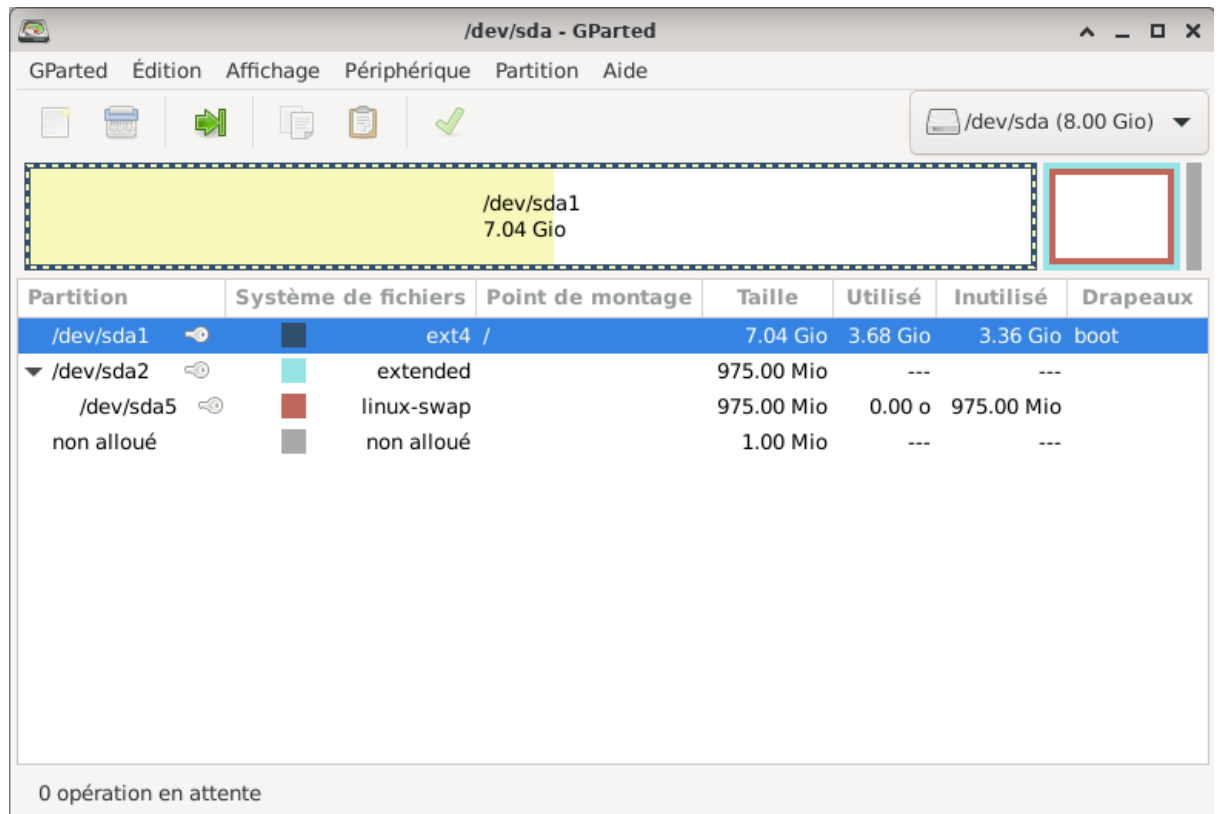
- une partition EXT4, nom cle01.3

```
debian@debian:~$ sudo mkfs.vfat -n cle01_1 /dev/sdb1
mkfs.fat 4.2 (2021-01-31)
mkfs.fat: Warning: lowercase labels might not work properly on some systems
debian@debian:~$ sudo mkfs.vfat -n cle01_2 /dev/sdb2
mkfs.fat 4.2 (2021-01-31)
mkfs.fat: Warning: lowercase labels might not work properly on some systems
debian@debian:~$ sudo mkfs.vfat -n cle01_3 /dev/sdb3
mkfs.fat 4.2 (2021-01-31)
mkfs.fat: Warning: lowercase labels might not work properly on some systems
debian@debian:~$
```

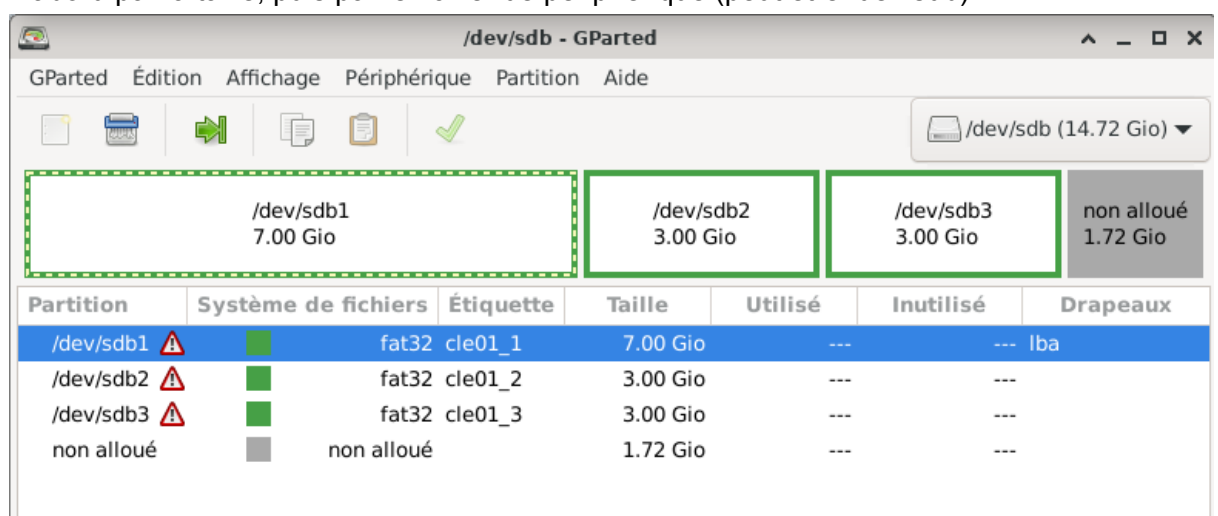
Vérifier comme précédemment, en faisant des tests sur le montage et l'utilisation des trois partitions

- 2.5 Partitionnement avec l'outil graphique gparted

- Installer le gestionnaire de partition "gparted"
- Lancer le dans un terminal ou à partir du menu système
- Découvrir le système de fichiers des partitions du disque virtuel en fonctionnement /dev/sda



- Choisir dans le menu à droite de gparted la clé usb. Comment elle est identifiée ? D'abord par la taille, puis par le fichier de périphérique (peut être /dev/sdb)



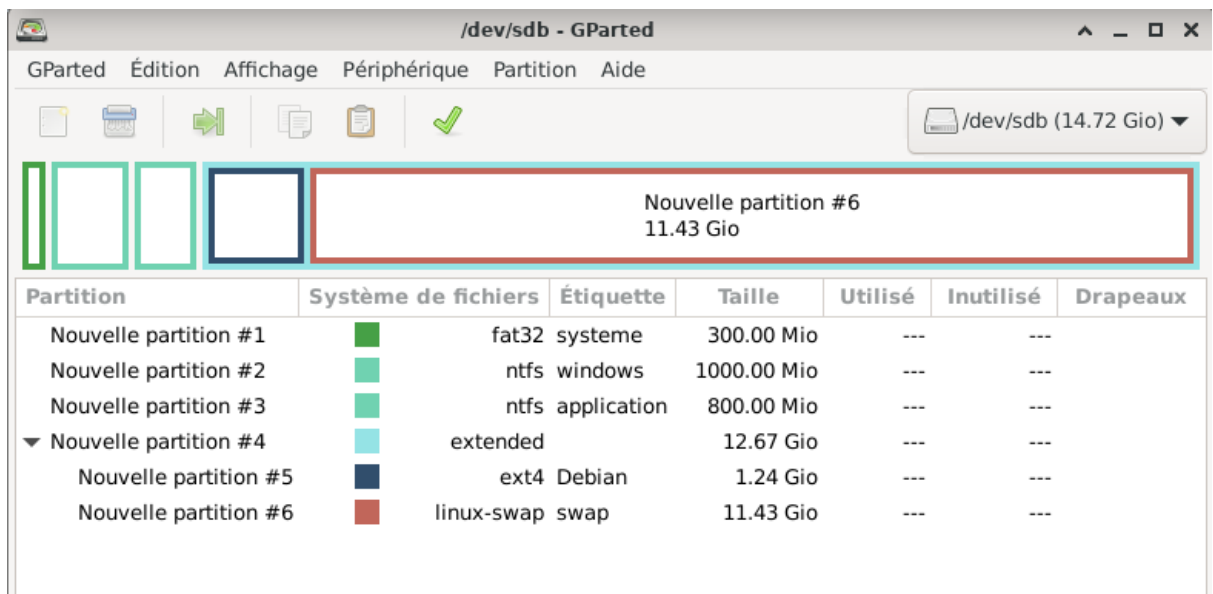
- Supprimer toutes les partitions du dispositif usb /dev/sdb
- Créer 4 partitions :

partition 1 : primaire, fat32, étiquette : systeme, 300Mi, bootable
 partition 2 : primaire, ntfs, étiquette : windows, 1Gi
 partition 3 : primaire, ntfs, étiquette : application, 800Mi
 partition 4 : étendu avec 2 partitions (sur le reste d'espace mémoire

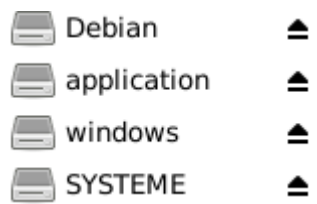
disponible :

- première partition : ext4, étiquette : Debian, 1270Mi
- seconde partition : swap-linux, étiquette : swap, sur le reste

de l'espace disque disponible



- Fermer gparted
- Vérifier avec le gestionnaire de fichier la présence de toutes les partitions. Faites quelques tests, en ajoutant des dossiers, des fichiers dans les partitions



- Démonter la clé usb, et vérifier le montage automatique de la clé dans la machine hôte, ...