Systèmes de gestion de fichiers et gestion des disques sous Linux

S.berraho@emsi.ma
Berraho.sanae@gmail.com

- Introduction
- Système de fichier
- Représentation d'un fichier
- Organisation générale d'un disque
- Organisation d'un disque sous Linux
- Partitionnement d'un disque sous Linux
- Initialisation du système de fichier
- Montage d'une partition

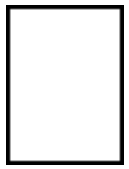
Système de fichiers (1)

- **Un système de fichiers** (file system) définit **l'organisation d'un volume physique ou logique.**
- C'est une structure de données permettant de **stocker les informations et de les organiser dans des fichiers** sur ce que l'on appelle des **mémoires secondaires** (physiquement des **mémoires de masse** comme disque dur, disquette, CD-ROM, clé USB, disques SSD, etc.).
- Un système de fichiers offre à l'utilisateur une vue abstraite sur ses données (**une arborescence de fichiers et de répertoires**) et permet de les **localiser** à partir d'**un chemin d'accès.**

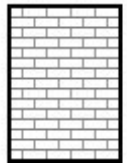
Système de fichier (2)

- **Le fichier** est la plus petite entité logique de stockage. **Un nom** lui est associé pour accéder à **son contenu**. Les données du fichier sont stockées dans des suites de **blocs** (la plus petite unité du périphérique de stockage).
- Un système de gestion de fichiers (**SGF**) ou file system (**fs**) est chargé de gérer l'organisation des informations mémorisées sur **les périphériques de bloc** tels que les disques durs, clé USB..
- **Le formatage** (action de formater c'est-à-dire **créer un système de fichiers**) prépare un support de données de stockage en y inscrivant un système de fichiers, de façon à ce qu'il soit **reconnu** par un système d'exploitation.
- Il existe de **nombreux systèmes de fichiers** différents : FAT, NTFS, HFS, ext, UFS, ISO 9660, ReiserFS, APFS, NFS, etc

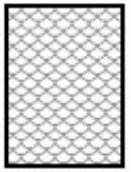
Système de fichier (3)



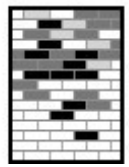
Disque dur non utilisé
(Non formaté)



Disque dur avec un système de fichiers



Disque dur avec un système de fichiers différent



Disque dur sur lequel figurent les données

Extension

- **Le nom d'un fichier** peut posséder un suffixe (**une extension**) séparé par un point qui est fonction du contenu du fichier : .txt pour un fichier texte par exemple.
- **Sous Windows**, cette extension permettra de choisir les applications qui prendront en charge ce format de fichier. Historiquement, l'extension était codée sur 3 caractères dans le système de fichiers FAT utilisé par Microsoft.
- **Sous GNU/Linux**, l'extension fait **simplement partie du nom de fichier**.

Le fichier dans un système de fichiers

- Chaque **fichier** est vu par le système de fichiers de plusieurs façons :
  - ⇒ **un descripteur** de fichier (souvent un entier unique) permettant de l'identifier
  - ⇒ **une entrée dans un répertoire** permettant de le situer et de le nommer
  - ⇒ **des métadonnées sur le fichier** permettant de le définir et de le décrire (attributs)
  - ⇒ **un ou plusieurs blocs** (selon sa taille) permettant d'accéder aux données du fichier (son contenu)

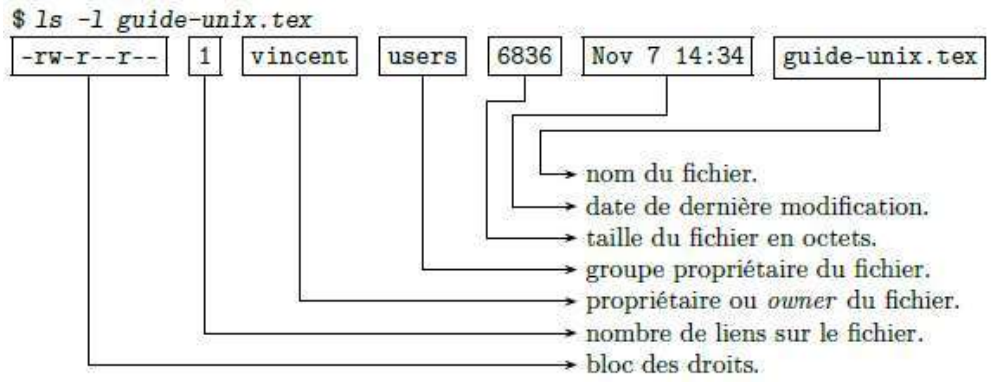
Attributs d'un fichier (1)

Attributs d'un fichier: Constituent les propriétés du fichier

Exemples d'attributs:

- ⇒ **Nom:** pour permettre aux personnes d'accéder au fichier
- ⇒ **Identificateur:** Un nombre permettant au SE d'identifier le fichier
- ⇒ **Type:** binaire, ou texte; lorsque le SE supporte cela
- ⇒ **Position:** Indique le disque et l'adresse du fichier sur disque
- ⇒ **Taille:** en bytes ou en blocs
- ⇒ **Protection:** Détermine qui peut écrire, lire, exécuter…
- ⇒ **Date:** pour la dernière modification, ou dernière utilisation
- ⇒ Autres…

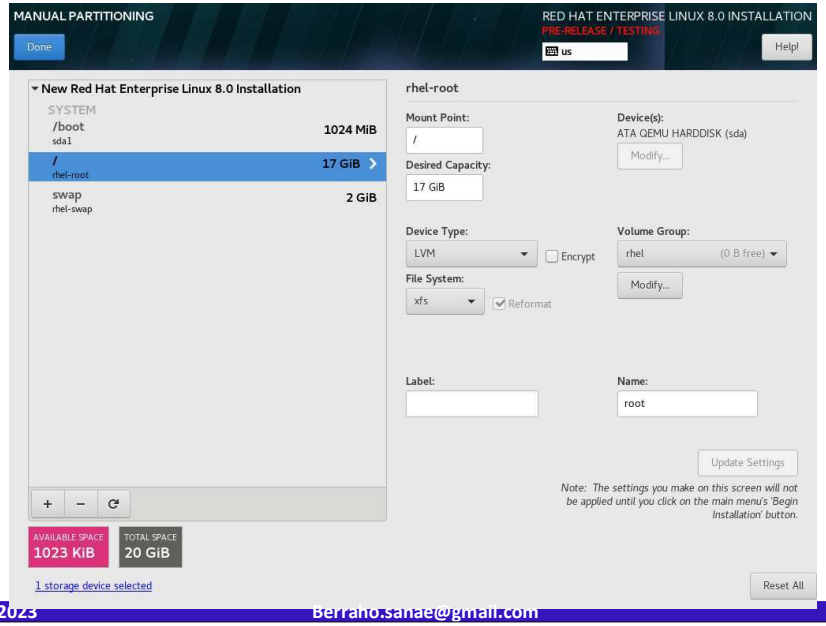
Attributs d'un fichier (2)



Systèmes de fichiers GNU/Linux (1)

- **L'extended file system** ou **ext**, est le premier S.F. créé en avril **1992** spécifiquement pour GNU/Linux.
- **1993** : Version 2, **ext2**, évolution de ext. Ext2 est très répandu, robuste et performant. Conçu de manière évolutive: il prévoit l'ajout de nouvelles capacités sans réécriture du code
- **2001 :** Version 3, **ext3**, ajoute la journalisation à ext2. La journalisation améliore la récupération des données en cas d'arrêt brutal du système.
- **2008 :** Version 4, **ext4,** amélioration de ext3.

Systèmes de fichiers par défaut dans RHEL 8



Système de gestion de fichiers: Définition

- **Système de gestion de fichiers:** ensemble de principes et de règles selon lesquels les fichiers sont **organisés et manipulés.** Chaque système d'exploitation possède son système de fichier privilégié, même s'il peut en utiliser d'autres.

OS	Système de fichier
MSDOS	FAT
Windows 95/98	FAT32 (File Allocation Table 32bits)
Windows NT	NTFS (New Technology File System) standard pour les ordinateurs sous Windows ¿ depuis VISTA.
MAC OS	HFS+ (High Performance File System)
Linux	ext4 (Extended File System)

Représentation de fichiers sur disque (1)

- Les informations d'un fichier sont sauvegardées dans **l'INODE** du fichier (**nœud d'index**) avec d'autres données.
- Deux concepts fondamentaux:
  - ⇒ **Le bloc :** unité de transfert entre le disque et la mémoire (souvent 4096 octets)
  - ⇒ **L'inode (index node)** : descripteur d'un fichier

Un fichier Unix est une suite finie de bytes (octets) Matérialisée par des **blocs disques**, et une **inode** qui contient les propriétés du fichier (**mais pas son nom)**

Représentation des fichiers sur disque (2)

Concept de l'inode

- Le terme **inode** (contraction de **« index »** et **« node »,** en français : nœud d'index) désigne le **descripteur d'un fichier** sous UNIX.
- L'inode d'un fichier est identifié par un numéro (**i-number**) dans le système de fichiers et contient les **métadonnées du fichier**:
  - ⇒ **Type de l'inode** (fichier ordinaire, répertoire, autres)
  - ⇒ **Propriétaire, droits, dates de création/modification/accès, Taille**
  - ⇒ **Liste des blocs du contenu du fichier**
  - ⇒ **…**
- Tout fichier possède son **unique i-node.**
- Les i-nodes sont tous **de même taille**.
- Donc, dans ce cours : **fichier = inode + blocs du fichier**

```
$ ls -i asup # num d'inode du fichier asup
10224747 asup
```


Représentation des fichiers sur disque (3)

- La commande **stat** permet d'afficher l'intégralité du contenu de l'**inode**

$ stat *nom_fichier*

- Le nom du fichier
- Sa taille
- Son nombre de blocs
- Son inode
- Ses permissions
- L'UID de son propriétaire
- Le GID de son propriétaire
- Ses mtime, atime et ctime
- ….

```
[root@192 bsanae]# stat file1
  File: file1
  Size: 36              Blocks: 8          IO Block: 4096   regular file
Device: fd00h/64768d    Inode: 38204600    Links: 1
Access: (0644/-rw-r--r--)  Uid: (    0/    root)   Gid: (    0/    root)
Context: unconfined_u:object_r:user_home_t:s0
Access: 2023-12-03 10:07:43.104753012 +0100
Modify: 2023-12-03 10:07:43.101503099 +0100
Change: 2023-12-03 10:07:43.101503099 +0100
 Birth: 2023-12-03 10:06:19.742330157 +0100
```


Représentation des fichiers sur disque (4)

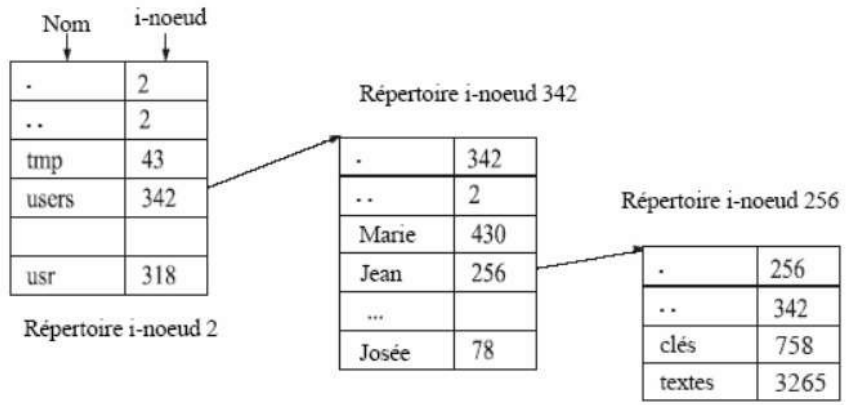
Sous Unix, **un i-node** contient les informations suivantes :

- **Taille du fichier** (en octets) ;
- **Identifiant du disque** qui contient le fichier ;
- **UID :** User IDentifier du propriétaire du fichier ;
- **GID :** Group IDentifier du propriétaire du fichier ;
- **Droits d'accès** (en lecture, écriture, exécution) ;
- **Nombres de liens** (physiques) ;
- **atime (access time) :** Date de dernière ouverture du fichier ;
- **mtime (modification time) :** Date de dernière modification du fichier
- **ctime (change Time) :** Date de dernière modification de l'i-node lui-même ;
- **Adresse :** pointeur sur les blocs du disque sur lesquels le fichier est stocké.

⇒ **Les inodes des fichiers ne contiennent pas les noms des fichiers**

Représentation des fichiers sur disque (5)

Structure d'un répertoire sous Unix



Représentation des fichiers sur disque (6)

Notion de chemin

- **Chemins relatifs :**
  . : répertoire courant ("ici")
  .. : répertoire parent ("au dessus")
  ~ ou **$HOME** : répertoire personnel ("chez moi")

- **Chemin absolu :**
  / : la racine (root)

Représentation des fichiers sur disque (7)

Notion de lien

- **Un lien** est un type spécial de fichiers qui **fait référence à un autre fichier**

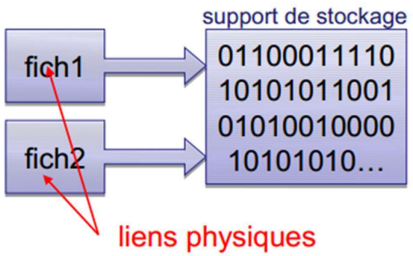
- Le lien permet:
  ⇒ De créer **des raccourcis** vers des fichiers existants
  ⇒ **D'éviter de stocker plusieurs fois le même fichier** dans des répertoires différents

Représentation des fichiers sur disque (8)

Lien physique

- **Un lien physique** est associé à **un emplacement sur le support de stockage.**
- Similaire à la notion de **pointeurs du langage c**
- Deux liens physiques sont considérés comme **deux fichiers indépendants.**
  ⇒ un lien physique permet d'avoir plusieurs noms de fichiers qui partagent exactement le même contenu (**le même i-node**)

- **Avantage :** On peut changer le contenu ou la localisation du fichier original, le lien physique restera valide.
- **Inconvénient :** un lien physique référence un numéro d'inode ce qui impose de rester dans la même partition (en effet : deux disques logiques différents peuvent avoir chacun une inode 47).
- **syntaxe** : `ln toto ../R2/titi`

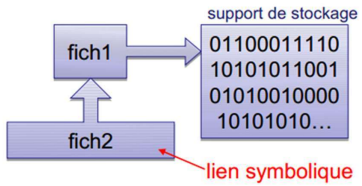


Représentation des fichiers sur disque (9)

Lien symbolique

- **Un lien symbolique** est une référence vers un fichier cible
- Utile pour simplifier et réduire l'utilisation du disque quand deux fichiers ont le même contenu.
- Ressemblent plus **aux raccourcis**
- Le principe du lien symbolique est que l'on crée un lien vers un **autre nom de fichier**
- Cette fois le lien pointe **vers le nom du fichier et non pas vers le i-node directement**
- Avec un lien symbolique on peut **traverser les partitions** (plus aucun risque que deux inodes différents aient le même numéro).

- **syntaxe** : `ln -s toto ../R2/titi`
- Par défaut, le lien symbolique doit être **crée dans le répertoire courant**. Si on veut spécifier aussi un chemin relatif pour le lien symbolique, utiliser l'option **r** .



Organisation générale d'un disque dur

- Les fichiers sont enregistrés sur des **disques**.
- Les disques sont découpés **en partitions,**
- Les systèmes de fichiers sont **indépendants** sur chaque partition.
- Chaque disque du système contient **au moins une partition**
- Intérêt d'allouer de l'espace disque sur des **partitions séparées** :

  ⇒ **Séparation logique des données** du système d'exploitation à partir des données utilisateur
  ⇒ Capacité à utiliser **différents systèmes de fichier**
  ⇒ Capacité à exécuter **plusieurs systèmes d'exploitation sur une machine**

Partitionnement (1)

- Les disques peuvent être **divisés en partitions**. Chaque partition est accessible comme s'il s'agissait d'un **disque distinct**. Ceci est possible grâce à l'ajout d'une table des partitions.
- **Une table de partitions** permet définir les partitions, avec leur début et fin sur le support de stockage.
- Il existe deux types de tables de partitions:

  ⇒ le type **MBR (Master Boot Record)**
  ⇒ le type **GPT (GUID Partition Table).**

Partitionnement (2)

Tables de partitions: MBR vs GPT

- **MBR** est une ancienne méthode de partitionnement de disque utilisée avec des ordinateurs basés **BIOS**.
- **GPT** est un agencement de partitionnement plus récent faisant partie de l'**UEFI** (Unified Extensible Firmware Interface)

Organisation générale du disque (1)

Partitionnement MBR (Master Boot Record)

- **Le MBR** est situé dans les 1er secteurs du disque
- Sa taille est de 512 octets, Il est constitué de 2 parties :
  ⇒ **La table des partitions (**adresse début et fin de chaque partition**).**
  ⇒ **Le programme d'amorçage** qui charge le noyau du système
- Au démarrage de l'ordinateur, **le BIOS** lit le contenu du MBR du disque placé en première position dans l'ordre d'amorçage.
- Après la lecture du MBR du disque, le BIOS va exécuter **le code d'amorçage** situé dedans.
- L'objectif du **code d'amorçage** est de lancer le chargeur d'amorçage (**bootloader**) du système d'exploitation situé sur la partition du disque marquée comme active.

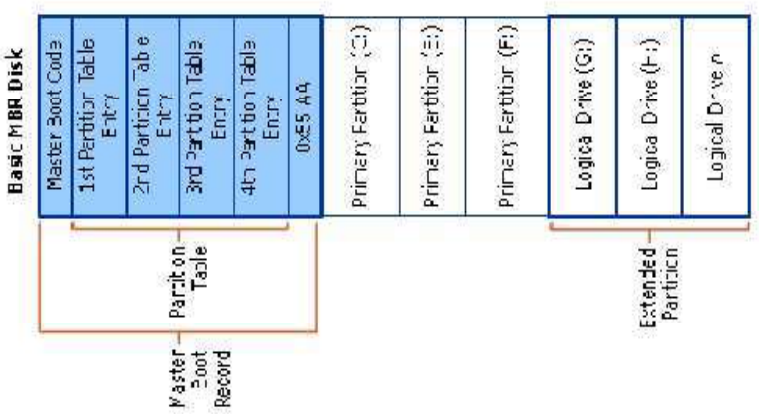
Organisation générale du disque (2)

Partitionnement MBR (Master Boot Record): Types de partitions

- **Partition primaire:**
  ⇒ Une obligatoire et au maximum 4.
- **Partition étendue:**
  ⇒ Une par disque qui remplace une partition primaire (max : 3 primaires + 1 étendue).
  ⇒ Uniquement comme **conteneur** de partitions logiques.
- **Partition logique:**
  ⇒ Contenue dans une partition étendue.

Organisation générale du disque (3)

Structure d'un disque au format MBR



Organisation générale du disque (4)

Partitionnement MBR: Limitations

- Ce système comporte quelques **limitations** qu'il faut connaître si on souhaite encore l'utiliser :
  ⇒ **La taille des partitions** est limitée à 2,2To.
  ⇒ On ne peut créer que **quatre partitions principales** (primaires)
  ⇒ **Incompatible** avec le mode **UEFI**

Organisation générale du disque (5)

Partitionnement GPT (GUI Partition Table)

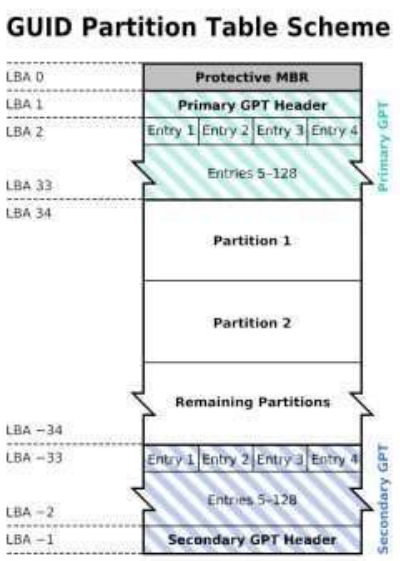
- Chaque partition possède son propre identifiant unique (**GUID**)
- Permet de dépasser les contraintes sur le nombre de partitions
  ⇒ Il supporte jusqu'à 128 partitions
  ⇒ La taille des partitions est très grande
  ⇒ **Compatible** avec le mode **UEFI** et **BIOS**

Si l'on prévoit de créer de **nombreuses partitions physiques** sur le même disque, il convient de créer une table de partition au **format GPT** lors de l'étape du partitionnement.

Organisation générale du disque (6)

Structure d'un disque au format GPT

- Pour garantir une compatibilité avec les logiciels gérant uniquement le MBR, le GPT possède **un MBR protecteur (Protective MBR).**
- **Deux GPT sont présents** sur le disque : l'un **primaire**, l'autre **secondaire** (qui est une sauvegarde du premier). Le primaire se situe au début du disque alors que le secondaire se situe à la fin du disque.
- La table des partitions est précédée d'un **entête** qui contient **les informations sur le disque** : blocs utilisables, GUID, nombre et taille des partitions…



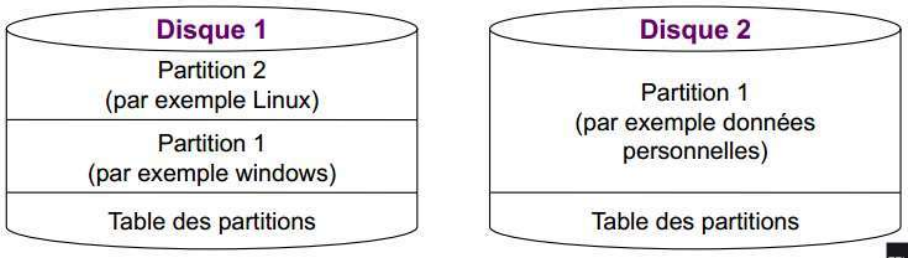
Organisation générale du disque (7)

MBR ou GPT ???

MBR : BIOS	GPT : BIOS/UEFI
Windows XP/Vista/7/8/10/11	Windows 7/8/10/11
32 bits / 64 bits	64 bits
4 partitions maximum	128 partitions maximum
2 To maximum par partition	Jusqu'à 9.4 Zo. Sur Windows c'est 256 To

Organisation des disques sous Linux (1)

- Une machine peut posséder **plusieurs disques**
- Et chaque disque peut être **scindé en plusieurs partitions**
  ⇒ Utile pour installer plusieurs systèmes d'exploitation ou pour augmenter l'indépendance entre les données utilisateurs et le système d'exploitation
- Chaque partition **possède son système de fichiers indépendant**



Organisation des disques sous Linux (2)

- Le pointeur spécial **/dev** permet l'accès aux disques
- Format des pointeurs sur disque :

- **Types de bus:**
  ⇒ **hd** : Périphériques IDE
  ⇒ **sd :** Périphériques SATA, SCSI, SSD, etc.
  ⇒ **nvme :** Périphériques NVME

- **Exemples:**
  ⇒ **/dev/hda1** : Partition 1 sur le 1er disque IDE
  ⇒ **/dev/sdb2** : Partition 2 sur le 2ème disque Sata,SCSI, etc.
  ⇒ **/dev/nvme0p1**: première partition sur le premier disque NVMe

Informations sur les disques et les partitions

- Afficher les informations sur les disques et les partitions:

```
# lsblk
```

```
[root@192 bsanae]# lsblk
NAME              MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                 8:0    0   20G  0 disk
sdb                 8:16   0   20G  0 disk
sr0                11:0    1  905M  0 rom  /run/media/bsanae/RHEL-8-8-0-BaseOS-x86_64
nvme0n1           259:0    0   20G  0 disk
├─nvme0n1p1       259:1    0    1G  0 part /boot
└─nvme0n1p2       259:2    0   19G  0 part
  ├─rhel_192-root 253:0    0   17G  0 lvm  /
  └─rhel_192-swap 253:1    0    2G  0 lvm  [SWAP]
```


Organisation des disques sous Linux (3)

- Au lieu de nommer les partitions ainsi, on préfère maintenant les identifier par **un numéro unique** écrit en hexadécimal

  ⇒ **Universal Unique Identifier:** Identifiant Universel Unique: Il s'agit d'une suite plus ou moins longue de caractères alpha-numériques qui permet d'identifier de façon absolument sûre chaque périphérique de stockage et partition.

  ⇒ Ce n° est inscrit au début de chaque partition

- Pour obtenir l'UUID:

```
# blkid
```

```
[root@192 bsanae]# blkid
/dev/nvme0n1p1: UUID="b8636bc2-1ea6-4ed6-beb0-8acf40e15a1a" BLOCK_SIZE="512" TYPE="xfs" PARTUUID="b242a2b4-01"
/dev/nvme0n1p2: UUID="tN3i0N-Ck2f-5oUV-cpeu-P2AT-vQa4-PxFCiJ" TYPE="LVM2_member" PARTUUID="b242a2b4-02"
/dev/sr0: BLOCK_SIZE="2048" UUID="2023-04-10-21-33-23-00" LABEL="RHEL-8-8-0-BaseOS-x86_64" TYPE="iso9660" PTUUI
/dev/mapper/rhel_192-root: UUID="46c078e5-ad69-43a8-b02f-f0231c9c6440" BLOCK_SIZE="512" TYPE="xfs"
/dev/mapper/rhel_192-swap: UUID="d5075efe-5b93-4774-af03-07e4f6b72de3" TYPE="swap"
/dev/nvme0n1: PTUUID="b242a2b4" PTTYPE="dos"
```


Informations sur les disques et les partitions

- Afficher la liste des partitions:

```
# ls /dev/
```

```
[root@192 bsanae]# ls /dev/sd*
/dev/sda  /dev/sdb
[root@192 bsanae]# ls /dev/nvme*
/dev/nvme0  /dev/nvme0n1  /dev/nvme0n1p1  /dev/nvme0n1p2
[root@192 bsanae]#
```

- Afficher la liste des partitions et leur taux de remplissage:

```
# df -h
```

```
[root@192 bsanae]# df -h
Filesystem                Size  Used Avail Use% Mounted on
devtmpfs                  1.8G     0  1.8G   0% /dev
tmpfs                     1.8G     0  1.8G   0% /dev/shm
tmpfs                     1.8G  9.8M  1.8G   1% /run
tmpfs                     1.8G     0  1.8G   0% /sys/fs/cgroup
/dev/mapper/rhel_192-root   17G  6.7G   11G  40% /
/dev/nvme0n1p1            1014M  271M  744M  27% /boot
tmpfs                     364M   24K  364M   1% /run/user/1000
/dev/sr0                  905M  905M     0 100% /run/media/bsanae/RHEL-8-8-0-BaseOS-x86_64
```


Informations sur les disques et les partitions

- Affiche l'espace occupé par une arborescence:

```
# du <chemin>
```

-s: uniquement la somme
-h: un format plus lisible

```
[root@192 bsanae]# du -h /home/bsanae/
0       /home/bsanae/.mozilla/extensions/{ec8030f7-c20a-464f-9b0e-13a3a9e97384}
0       /home/bsanae/.mozilla/extensions
0       /home/bsanae/.mozilla/plugins
0       /home/bsanae/.mozilla/firefox/okv7msix.default-default/storage/permanent/chrome/idb/3870112724rs
0       /home/bsanae/.mozilla/firefox/okv7msix.default-default/storage/permanent/chrome/idb/3561288849sd
0       /home/bsanae/.mozilla/firefox/okv7msix.default-default/storage/permanent/chrome/idb/1451318868nt
0       /home/bsanae/.mozilla/firefox/okv7msix.default-default/storage/permanent/chrome/idb/2823318777nt
0       /home/bsanae/.mozilla/firefox/okv7msix.default-default/storage/permanent/chrome/idb/1657114595Am
0       /home/bsanae/.mozilla/firefox/okv7msix.default-default/storage/permanent/chrome/idb/2918063365pi
9.3M    /home/bsanae/.mozilla/firefox/okv7msix.default-default/storage/permanent/chrome/idb
9.3M    /home/bsanae/.mozilla/firefox/okv7msix.default-default/storage/permanent/chrome
9.3M    /home/bsanae/.mozilla/firefox/okv7msix.default-default/storage/permanent
0       /home/bsanae/.mozilla/firefox/okv7msix.default-default/storage/temporary
0       /home/bsanae/.mozilla/firefox/okv7msix.default-default/storage/default/moz-extension+++f0b0be9b-
ContextId=4294967295/idb/3647222921wleabcEoxlt-eengsairo.files
44K     /home/bsanae/.mozilla/firefox/okv7msix.default-default/storage/default/moz-extension+++f0b0be9b-
ContextId=4294967295/idb
48K     /home/bsanae/.mozilla/firefox/okv7msix.default-default/storage/default/moz-extension+++f0b0be9b-
```

```
[root@192 bsanae]# du -s /home/bsanae/
103100  /home/bsanae/
[root@192 bsanae]# du -sh /home/bsanae/
101M    /home/bsanae/
[root@192 bsanae]#
```

Création de partitions (1)

- Pour créer des partitions sous Linux, plusieurs outils existent:
  - ⇒ **parted,**
  - ⇒ **fdisk,**
  - ⇒ **sfdisk,**
  - ⇒ **gdisk,**
  - ⇒ **sgdisk**

Création de partitions (2)

- Ces outils différent dans le **type de partitionnement**:
  - ⇒ **fdisk, sfdisk, parted** sont compatibles avec le partitionnement **MBR** et **GPT**
  - ⇒ **gdisk et sgdisk** sont compatibles avec le partitionnement **GPT** seulement, (mais permettent de transformer un partitionnement MBR en GPT et réciproquement)
- Ils différents aussi dans le **mode de fonctionnement**:
  - **fdisk, gdisk** fonctionnent en **mode interactif**
  - **sfdisk, sgdisk** fonctionnent en **mode script**
  - **parted** est utilisable en **mode interactif, script et mixte**

Création de partitions (3)

Création d'une partition: Précautions

- Le partitionnement a pour effet de **créer ou modifier la table de partition et son contenu**, il **ne modifie pas le contenu des partitions.**
- Mais l'écrasement ou la modification malheureuse de la table de partition a pour effet de rendre **inaccessible l'accès aux données**. Il est donc fortement recommandé avant toute opération portant sur la table de partition :
  - ⇒ de sauvegarder les données
  - ⇒ d'identifier correctement les unités de disque ou les partitions à modifier

Création de partitions (4)

Commande fdisk:

- Manipuler la table de partitions d'un disque en mode interactif.

```
# fdisk <disque>
```

- Afficher les partitions d'un disque :

```
# fdisk -l <disque>
```

- Commandes de **fdisk**:
  - ⇒ **m:** afficher l'aide.
  - ⇒ **p :** lister les partitions.
  - ⇒ **n :** créer une nouvelle partition.
  - ⇒ **d :** supprimer une partition.
  - ⇒ **t :** modifier le type d'une partition.
  - ⇒ **w :** sauvegarder la table de partitions
  - ⇒ **g:** créer une table de partitions de type GPT

- Type de partition: Code hexadécimal.
- Exemple :
  - ⇒ **5 :** Partition étendue.
  - ⇒ **7 :** NTFS.
  - ⇒ **b :** FAT32
  - ⇒ **82** : Linux swap
  - ⇒ **83 :** Linux
  - ⇒ **8e :** LVM

Initialisation du système de fichiers

Initialisation du système de fichiers = Formatage

- La création d'un système de fichier (formatage) sur une partition peut se faire avec la commande **mkfs**.

```
# mkfs -t <type> <partition>
```

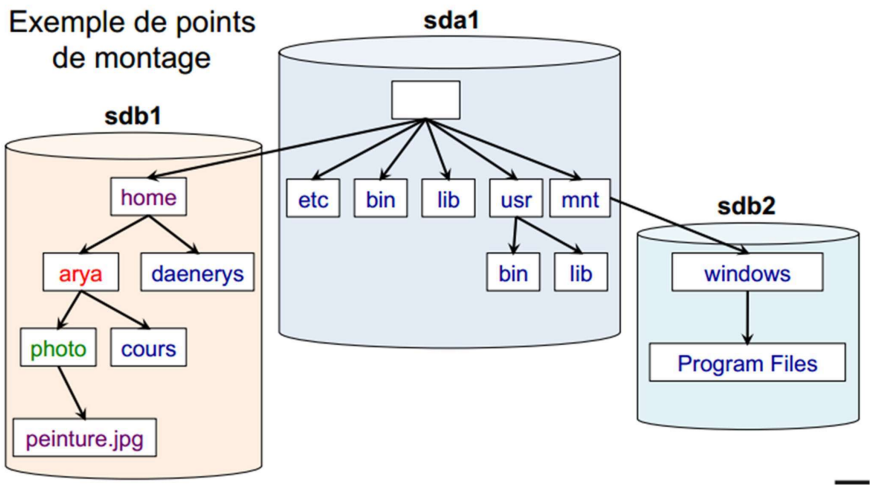
- Le type détermine la commande à exécuter :
  - ⇒ **ext3** : mkfs.ext3
  - ⇒ **ext4** : mkfs.ext4
  - ⇒ **reiserfs** : mkfs.reiserfs
  - ⇒ **vfat** : mkfs.vfat
  - ⇒ **ntfs** : mkfs.ntfs
- Vérifier le type de système de fichier:

```
# lsblk -f
```


Montage d'une partition (1)

- L'arborescence Unix peut être construite à partir de **diverses partitions** qui peuvent être situées **sur plusieurs disques**
- Le processus de **montage**, est le moyen de **faire correspondre parties de l'arborescence et partitions physiques de disque**.
- **Un point de montage** est un répertoire à partir duquel sera **accessible** le système de fichiers
- Il suffira ensuite de se déplacer à ce répertoire, appelé **point de montage**, en fait **un répertoire "d'accrochage",** pour accéder à ses fichiers

Montage d'une partition (2)



Montage d'une partition (3)

Points de montage

- Tant qu'ils ne sont pas effectués, **le système de fichiers est inaccessible**
- Il est possible de créer **un point de montage manuellement**
  - ⇒ Pour les clés USB ou les disques durs ou un CDROM par exemple
  - ⇒ En utilisant la commande **« mount »**
  - ⇒ Et **« umount »** pour supprimer le point de montage

Pour résumer:

⇒ Pour accéder aux fichiers d'une partition, on doit la « **monter** » (**mount**)

⇒ Cela fait **apparaître son contenu dans l'arbre** des fichiers Unix

Montage d'une partition (4)

- **Mount:** Monter un système de fichiers.

```
# mount     partition   point_de_montage
```

- Le point de montage doit être existant. Son contenu (éventuel) deviendra invisible.
- Sans aucun argument **mount** affiche les systèmes de fichiers actuellement montés.
- Options (de la commande) :
  - ⇒ **– t** : Type du SF (détection automatique si pas indiqué).
  - ⇒ **– a** : Monter tous les SF du fichier **/etc/fstab**
  - ⇒ **– L** : Montage par label (étiquette).
  - ⇒ **– U** : Montage par UUID.

Montage d'une partition (4)

- Démonter un système de fichiers.

```
# umount      point_de_montage
```

- On ne peut pas démonter une partition si :
  - ⇒ Une commande s'exécute dans la partition ;
  - ⇒ Un fichier dans cette partition est ouvert ;
  - ⇒ Un répertoire de la partition est le répertoire courant.
- Solution : **lsof**, **fuser**.

Montage d'une partition (5)

Montage automatique: Fichier /etc/fstab

- Contient une configuration statique des différents montages des systèmes de fichiers.
- Permet de faire le montage des systèmes de fichiers **lors du démarrage.**
- Chaque ligne indique :
  - ⇒ Le périphérique à monter ;
  - ⇒ Le point de montage ;
  - ⇒ Le type du système de fichiers ;
  - ⇒ Les options de montage ;
  - ⇒ Fréquences de sauvegarde et de vérification.

Montage d'une partition (6)

Structure du fichier /etc/fstab

Les lignes de **/etc/fstab** contiennent 6 « mots » :

1. Le périphérique associé à la partition, ex : **/dev/sdb1**
2. Le point de montage, ex : **/**
3. Le format de la partition, ex : **ext4**
4. Les options, ex : **defaults, noauto, user**
5. **dump :** activer ou non la sauvegarde du disque avec l'utilitaire dump (1 ou 0)
6. **fsck :** activer ou non la vérification du système de fichier avec l'utilitaire fsck (0, 1, 2 ou 3)

Montage d'une partition (7)

Structure du fichier /etc/fstab

```
#
# /etc/fstab
# Created by anaconda on Sun Oct  1 16:43:12 2023
#
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
/dev/mapper/rhel_192-root                /                       xfs     defaults        0 0
UUID=b8636bc2-1ea6-4ed6-beb0-8acf40e15a1a /boot                  xfs     defaults        0 0
/dev/mapper/rhel_192-swap                none                    swap    defaults        0 0
```