# Gestion des disques

# Introduction

Les administrateurs système doivent généralement faire face à de nombreux défis en matière de gestion des disques. On retrouve parmi ces défis ...

* Partitionnement des disques ;
* Mise en place de systèmes de fichiers ;
* Montage de systèmes de fichiers ;
* Partage de systèmes de fichiers ;
* Surveillance de l'espace libre dans les systèmes de fichiers ;
* Sauvegarde (et parfois restauration) des systèmes de fichiers ;

# BIOS et UEFI

Après avoir allumé un ordinateur, le premier programme qui démarre est, soit le BIOS, soit l’UEFI. Ceci est le point de départ de l'ordinateur. Il effectue une série de tests de diagnostic afin de détecter et connecter le processeur, la mémoire, le clavier, les disques durs et autres périphériques.   
Ce processus est connu sous le nom de Power on Self-Test(POST).   
Si tous les périphériques sont connectés sans aucun problème, le BIOS/UEFI trouvera et exécutera le programme du chargeur de démarrage.

## Différences entre le BIOS et l'UEFI

|  |  |
| --- | --- |
| BIOS | UEFI |
| BIOS signifie Basic Input Output System | **UEFI signifie Unified Extensible Firmware Interface** |
| Le BIOS est l'approche classique pour détecter et connecter des périphériques dans le système | **UEFI est la manière moderne de démarrer le système Il est développé en remplacement du BIOS** |
| Le BIOS a été inventé pour les ordinateurs personnels au début de 1982 | **L'UEFI (basé sur l'EFI d'Intel) est devenu disponible pour tous les fabricants en 2007** |
| Le BIOS fournit des fonctionnalités très basiques Il ne fournit aucun utilitaire de dépannage | **UEFI fournit plusieurs fonctionnalités avancées, y compris la compatibilité descendante UEFI fournit un environnement de pré-démarrage qui a son propre shell, ses pilotes et ses applications Cet environnement comprend des fonctionnalités de base pour le dépannage telles que le diagnostic à distance, le mode d'urgence, la connectivité Internet et la sauvegarde du stockage** |
| Tous les paramètres sont stockés dans la mémoire CMOS non volatile | **Tous les paramètres sont stockés dans la mémoire Flash** |
| Le BIOS ne comprend pas de pilote pour les cartes réseau modernes, les périphériques vidéo et de stockage | **Généralement, UEFI peut détecter tous les appareils modernes** |
| Le BIOS ne peut pas démarrer à partir d'une partition  supérieure à 2 téraoctets | **Le micrologiciel UEFI peut démarrer à partir d'un lecteur d'une taille allant jusqu'à 9 zébioctets** |
| Le BIOS s'exécute en mode processeur 16 bits et ne dispose  que de 1 Mo d'espace pour s'exécuter | **UEFI peut s'exécuter en mode 32 bits ou 64 bits** |
| Le BIOS prend en charge uniquement le schéma de partition MBR | **UEFI prend en charge les schémas de partition MBR et GPT** |

Du point de vue de la gestion des disques, la différence importante entre le BIOS et l'UEFI est que le BIOS prend uniquement en charge le schéma de partition MBR tandis que l'UEFI prend en charge à la fois le schéma de partition MBR et GPT.   
Il n'y a aucun moyen de passer du BIOS à l'UEFI sur un ordinateur existant. Il faut acheter un nouveau matériel qui prend en charge et inclut UEFI. Toutefois, la majorité des nouveaux ordinateurs utilisent UEFI au lieu du BIOS traditionnel.

Un disque dur a besoin d'une table de partition avant de pouvoir être utilisé. Il existe deux types de table de partition MBR et GPT. Si un système est équipé d’un BIOS, on ne peut utiliser que la table de partition MBR tandis que si un système est équipé de l'UEFI, nous pouvons utiliser n'importe quelle table de partition.

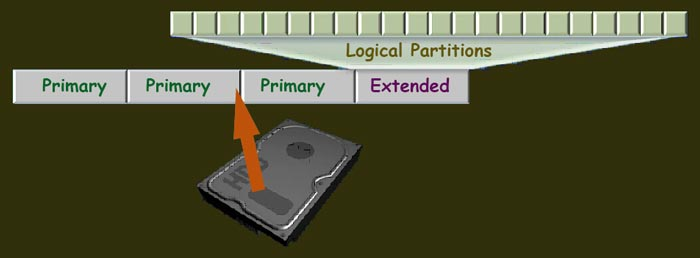
## Schéma de partition MBR

Le schéma de partition MBR a été inventé avec l'invention de l'ordinateur personnel en 1982.  
Ce schéma utilise le premier secteur du disque dur pour stocker toutes les informations qui sont nécessaires afin de démarrer le système, y compris le chargeur de démarrage et la table de partition.

Le chargeur de démarrage est un petit script utilisé pour charger le système d'exploitation.   
La table de partition est la disposition du disque dur.

La taille par défaut d'un secteur est de 512 octets. De ces 512 octets, seuls 64 octets sont utilisés pour stocker les informations de la table de partition.   
Les informations de la table de partition sont stockées dans des valeurs 32 bits.   
En raison de cet espace limité, la table de partition ne peut stocker des informations que sur quatre partitions avec une limitation de 2 térabytes dans chaque partition.

Peu à peu, la limite de quatre partitions est devenue un problème. Afin de trouver une solution à cette linite de quatre partitions, un petit changement a été apporté au schéma de partition MBR. Dans cette modification, une partition a été autorisée à **utiliser une partition étendue**.   
Dans cette partition étendue, un certain **nombre de partitions logiques peuvent être créées**. Le nombre de partitions logiques pouvant être créées dépend du système d'exploitation utilisé. Sous Linux, il est possible de **créer jusqu'à 15** **partitions**.   
Quel que soit le nombre de partitions logiques que l’on crée dans la partition étendue, la **taille de la partition étendue**, y compris toutes ses partitions logiques, **ne peut pas dépasser 2 téraoctets**.



Le schéma de partition MBR peut être utilisé sur un **disque dur de 8 téraoctets ou moins**.   
Pour utiliser un **disque dur supérieur à 8 téraoctets** ou pour créer une partition supérieure à 2 téraoctets, il faut un **schéma de partition GPT**.

## Schéma de partition GPT

Le schéma de partition GPT système a été inventé avec les normes UEFI afin de répondre aux exigences des périphériques de stockage modernes. Avec GPT, la taille maximale d'une partition est de 8 zébioctets.   
GPT autorise un maximum de 128 partitions.   
GPT utilise un ID unique global (GUID) 128 bits pour identifier la partition.   
Étant donné que GPT fournit un grand nombre de partitions (128) et un espace beaucoup plus grand dans chaque partition, il n'est pas nécessaire de diviser les partitions en partitions principales, étendues et logiques.

## Différence entre MBR et GPT

|  |  |
| --- | --- |
| MBR | GPT |
| MBR signifie Master Boot Record | **GPT signifie GUID Partition Table** | |
| MBR prend en charge un maximum de 4 partitions principales.  Une partition principale peut être utilisée  une partition étendue  La partition étendue peut être utilisée pour créer des partitions logiques | **GPT prend en charge 128 partitions maximum La méthode de partition principale, étendue et logique n'est pas utilisée dans GPT Toutes les partitions sont égales** | |
| La taille de partition maximale est de 2 téraoctets | **La taille de partition maximale est de 8 zébioctets** | |
| Utilise la norme 32 bits pour stocker la  table de partition | **Utilise la norme 64 bits pour stocker les informations de partition** | |
| Toutes les informations sont stockées dans le premier secteur du disque dur Si ce secteur est corrompu, le démarrage échouera | **Les informations sont stockées dans deux emplacements Si un emplacement est corrompu, le système démarrera à partir d'un autre emplacement** | |

## Noms de périphériques de disque courants

On retrouve dans un **écosystème Linux une nomenclature normalisée** du nom des périphériques.

* **/dev/hda**  
  Disque dur IDE (peu commun de nos jours) ;
* **/dev/sda**  
  Disque SCSI ou disque dur SATA ;
* **/dev/vda**  
  Disque dur d'une machine virtuelle qui utilise le pilote de disque virtio
* **/dev/xvda**  
  Disque dur de la machine virtuelle qui utilise le pilote de disque Xen

Le nom du premier disque commence par la lettre a.   
Par exemple, tous les noms de disques répertoriés dans le tableau ci-dessus représentent le premier disque de leur type.   
Si le système possède plusieurs disques, la lettre disponible suivante sera utilisée.   
Par exemple sdb représente le deuxième disque SCSI ou SATA, sdc représente le troisième disque SCSI ou SATA et sdf représente le sixième disque SCSI.

Les systèmes Linux modernes proposent toutefois une nouvelle méthode d’identification pour les périphériques de type bloc.

UUID (*Universal Unique IDentifier*) est une suite plus ou moins longue de caractères alphanumériques qui permet d'identifier de façon unique chaque périphérique de stockage et partition.

Le chiffre de l'UUID est calculé automatiquement au moyen d'un algorithme intégrant notamment certaines données de l'ordinateur hôte, au moment de la création ou du formatage de la partition ou de la table des partitions.   
Ce mode de calcul ne présente aucun risque de sécurité crédible.

Un UUID est de la forme : 1124d9e8-6266-4bcf-8035-37a02ba75c69.

Il est à noter que l'UUID d'une partition est stockée dans le système de fichier.   
Une partition vierge (non formatée) ne peut donc avoir d'UUID et une partition clonée a le même UUID que l'originale. Ceci est vrai même avec un partitionnement GPT.   
Bien que GPT définisse un UUID qui est stocké dans la table de partition, ce n'est pas ce dernier qui est utilisé par linux pour identifier les partitions.

Les UUIDs se retrouvent notamment dans deux fichiers de configuration ...

* Le fichier de configuration de GRUB, le chargeur de démarrage (/etc/default/grub) ;
* Le fichier récapitulant les partitions à monter au démarrage … (/etc/fstab).

À chaque formatage d'une partition, un nouvel UUID lui est attribué, il est alors parfois nécessaire de modifier ces fichiers.

Dans les fichiers de configuration, vous trouverez les UUID écrits de la façon suivante …   
UUID=0c22e844-4ecb-48d4-a3c3-f083ddb1df3b. mais on pourra trouver aussi la forme UUID=079B6B49295F7AD1.  
**>>** **ls -l /dev/disk/by-uuid  
total 0  
lrwxrwxrwx. 1 root root 10 mai 16 18:08 09ed2e23-8686-4f48-97bf-f96c7be8b15d -> ../../dm-1  
lrwxrwxrwx. 1 root root 10 mai 27 13:22 7e66eb31-5804-48b1-a1f1-7d2b21ad5ce1 -> ../../sda3  
lrwxrwxrwx. 1 root root 10 mai 16 18:08 8fdc3882-0700-45be-b692-2dc2f3735eaf -> ../../dm-2  
lrwxrwxrwx. 1 root root 10 mai 27 13:22 aef2e959-3d15-4842-ae1f-260d4188abb2 -> ../../sda2  
lrwxrwxrwx. 1 root root 10 mai 27 13:22 B9BD-AE5C -> ../../sda1**

**>>** **lsblk -fs**

**>>** **lsblk -f --size**

**NAME FSTYPE LABEL UUID FSAVAIL FSUSE% MOUNTPOINT**

**loop0 squashfs 0 100% /snap/core/8268**

**loop1 squashfs 0 100% /snap/core/8213**

**loop2 squashfs 0 100% /snap/hello-world/2**

**sda**

**├─sda1 vfat B9BD-AE5C 505,8M 1% /boot/efi**

**├─sda2 ext2 aef2e959-3d15-4842-ae1f-260d4188abb2 118,1M 45% /boot**

**└─sda3 crypto\_LUK 7e66eb31-5804-48b1-a1f1-7d2b21ad5ce1**

**└─sda3\_crypt LVM2\_membe J0FgfC-eubU-Q4Dl-p3jM-I1g9-my8a-Ikjk0t**

**├─debian10--vg-root**

**│ ext4 09ed2e23-8686-4f48-97bf-f96c7be8b15d 69,3G 63% /**

**└─debian10--vg-swap\_1**

**swap 8fdc3882-0700-45be-b692-2dc2f3735eaf [SWAP]**

De nos jours, il est préférable d’utiliser les UUID au lieu du schéma classique comme /dev/sda.

Avec le schéma classique, l’ordre des disques est donné en fonction de la découverte d’un disque. Si, par exemple, on ajoute un nouveau disque et que ce dernier est découvert avant le disque existant, il est possible que GRUB ne soit pas en mesure de trouver la partition BOOT.   
Comme résultat, le système ne sera pas en mesure de démarrer.

Puisque le UUID est unique à chaque disque, cette situation a peu de chance de se produire.

# Partitionnement

Les raisons pour partitionner un disque incluent ...

* la protection de certains systèmes de fichiers contre le **manque d'espace** (comme par exemple, on souhaite peut-être que la partition du **système d'exploitation soit séparée des répertoires personnels** ou **des applications** afin d’éviter qu'elle ne soit affectée si les fichiers des utilisateurs commencent à occuper beaucoup trop d'espace disque) ;
* l'amélioration des performances ;
* l'allocation d'espace d’échange (swap) ;
* la maintenance et les sauvegardes ;
* la gestion plus efficace (et ciblé) de la vérification des volumes ;
* le **maintien** (en particulier sur les systèmes de test) de **plusieurs systèmes d'exploitation** ;
* la gestion de l’espace disque pour l'extension du système de fichiers ;
* le partage de certains systèmes de fichiers avec d'autres systèmes.

Traditionnellement, les fichiers résidaient dans des systèmes de fichiers résidant dans des partitions de disque qui n'étaient elles-mêmes que des tranches de disques. Cette organisation domine encore aujourd'hui, bien que les serveurs des grands centres de données adoptent souvent une structure entièrement différente.

|  |
| --- |
| Fichiers et répertoires |
| Système de fichiers |
| Partition |
| Disque |

Cette vue simpliste fonctionne pour de nombreux systèmes, mais de nos jours, il existe de nombreuses complexités qui rendent la gestion des disques plus difficile à certains égards et plus facile à d'autres.

Un système de fichiers peut être virtuel - ne résidant plus sur un seul disque et plus complexe à gérer, mais beaucoup plus facile à redimensionner selon les besoins.

En fait, l'ensemble du système pourrait être virtuel. Ce que l'on pourrait gérer comme s'il s'agissait d'un seul disque pourrait en fait être une partie d'une très grande matrice de disques.

La gestion de la partition du disque dur d’un serveur Linux est une tâche critique.  
Les différents utilitaires vont aider l’administrateur système à gérer les partitions de votre disque dur.

Pour la plupart des serveurs Linux, le partitionnement est effectué avant le déploiement des serveurs.   
D'un autre côté, il est toujours possible d’ajouter des disques ultérieurement ou conserver une quantité importante d'espace disque libre pour une utilisation future.

Les commandes fdisk et parted sont utilisées à cette fin.

## Utilitaire fdisk

fdisk est une autre option courante pour les administrateurs système.   
Il répertorie les différentes partitions (qui sont liées aux disques durs car un disque dur peut être divisé en plusieurs partitions) du système.

fdisk renverra la totalité de l'espace (en Go ou en Mo), la quantité totale d'octets et la quantité totale de secteurs par chaque partition et, en résumé, elle donnera également les secteurs de début et de fin, la quantité d'espace disque   
( en octets) et le type de partition.

**Astuce** …  
Habituellement, un disque SATA est étiqueté avec sd*n*.

Pour apporter des modifications ou vérifier des partitions …  
**>>** **fdisk /dev/sdb**

fdisk entrera en mode interactif.   
On peut entrer la lettre m pour afficher une liste des opérations que l’on peut effectuer avec la commande fdisk.

**Commande (m for help): m**

|  |  |
| --- | --- |
| Action | Action |
| a | toggle a bootable flag |
| b | edit bsd disklabel |
| c | toggle the dos compatibility flag |
| d | delete a partition |
| l | list known partition types |
| m | print this menu |
| n | add a new partition |
| o | create a new empty DOS partition table |
| p | print the partition table |
| q | quit without saving changes |
| s | create a new empty Sun disklabel |
| t | change a partition's system id |
| u | change display/entry units |
| v | verify the partition table |
| w | write table to disk and exit |
| x | extra functionality (experts only) |

Comme on peut le constater, la commande fdisk fournit de nombreuses fonctionnalités.

## Utilitaire parted

parted est similaire aux autres outils de gestion des disques. Il répertorie toutes les partitions et permet de les gérer.   
Sa principale différence est qu'il informe également de la marque et du modèle des disques durs et même du type de connectivité utilisé (scsi, sata, ...) et de la taille totale du disque.

parted peut gérer à la fois les tables de partitionnement MBR et GPT.

**>>** **parted /dev/sdb**

Il est préférable de vérifier deux fois afin de s'assurer d'ajouter le périphérique de bloc que l'on souhaite travailler; sinon, parted peut s'exécuter sur sda ou sur le lecteur sur lequel le système d'exploitation est installé ce qui serait fort dommageable.   
parted est extrêmement puissant et le choix du mauvais disque peut entraîner une perte de données précieuse.   
Il faut donc faire preuve de prudence lors du travail sur un disque.

Après avoir accédé à la commande, on entre en mode interactif et (parted) est ajouté à l’invite de commande (*prompt*).

Il faut maintenant créer une nouvelle table de partition. Il y a un bon vieux MBR (enregistrement de démarrage principal) et un GPT plus récent (table de partition guid) …  
**(parted)** **mklabel msdos**

Ici, mklabel crée la table de partition et msdos utilisera MBR.   
On peut maintenant créer des partitions.   
Le format de base de la commande est …  
**(parted)** **mkpart «type de partition» «système de fichiers» début fin**

Si on souhaite utiliser tout l'espace disque disponible et créer une grande partition …  
**(parted)** **mkpart primary ext4 1MiB 100%**

Ici, 100% signifie qu'il utilisera tout l'espace disponible.

Toutefois, si on veut créer plusieurs partitions …  
**(parted)** **mkpart primary ext4 1MiB 20GB**

Ici, il créera une partition avec 20 Go de stockage. Ensuite, on va créer une autre partition, mais comme il en existe déjà une, le point final de la partition précédente est maintenant le point de départ de la partition suivante …  
**(parted)** **mkpart primary ext4 20GB 50GB**

Cette commande créera une deuxième partition de 30 Go. Si on souhaite créer une partition supplémentaire pour l'espace restant, on connaît le point d'arrivée et le point de départ …  
(**parted)** **mkpart primary ext4 50GB 100%**

Il est possible de remplacer ext4 par le type de fichier souhaité -- ntfs, vfat, btrfs, …

Pour vérifier si le partitionnement est conforme, on utilise la commande print …  
**(parted)** **print**

ISi tout se passe comme prévu, on quittr l'outil de partitionnement …  
**(parted)** **quit**

L'exécution de la commande lsblk affichera les partitions nouvellement créées.   
Comme dernière étape, il faut formater ces partitions avant de les monter et de les utiliser.   
On retrouve maintenant trois partitions sur sdb : sdb1, sdb2, sdb3.   
On formate chacune d’elle avec ext4 …  
**>>** **sudo** **mkfs.ext4 /dev/sdb1**

Il suffit de répéter cette dernière pour chaque volume- il suffit de simplement modifier le nom et le numéro du périphérique de bloc.

## Utilitaire sfdisk

sfdisk est similaire à l’utilitaire fdisk, mais sfdisk permet de voir les volumes physiques et logiques et donne également un résumé des partitions des volumes physiques réels avec les cylindres (début et fin), les secteurs, la taille et le type.

**>>** s**fdisk -l**

Toutes ces commandes devraient permettre de **voir au moins les volumes logiques**, les **partitions** et les **disques durs** disponibles sur un système

La plupart de ces commandes offrent également des capacités de gestion pour modifier et manipuler les partitions.

# Formatage d’une partition ou d’un volume

Maintenant que l’on a une partition, on peut **créer un système de fichiers à l'aide de l’utilitaire mkfs**.   
Par exemple, pour créer un système de fichiers ext4 sur la partition /dev/sdb1, on exécute la commande **mkfs** …

**>>** **sudo** **mkfs -t ext4 /dev/sdb1**  
**mke2fs 1.43.4 (31-Jan-2017)**  
**En train de créer un système de fichiers avec 5242624 4k blocs et 1310720 i-noeuds.**  
**UUID de système de fichiers=29ce3b50-824c-425a-87da-c0f682fb2183**  
**Superblocs de secours stockés sur les blocs :**  
 **32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208, 4096000**  
  
**Allocation des tables de groupe : complété**  
**Écriture des tables d'i-noeuds : complété**  
**Création du journal (32768 blocs) : complété**  
**Écriture des superblocs et de l'information de comptabilité du système de fichiers : complété**

La commande mkfs est une commande frontale de commandes conçues pour créer des systèmes de fichiers spécifiques. Par exemple, la commande mkfs -t ext4 exécute la commande mkfs.ext4.   
on peut voir quels systèmes de fichiers la commande **mkfs** peut créer en entrant **mkfs** et enappuyant deux fois sur la touche **TAB** …

**>>** **sudo mkfs TAB TAB**  
**mkfs mkfs.cramfs mkfs.ext3 mkfs.minix mkfs.vfat**  
**mkfs.bfs mkfs.exfat mkfs.ext4 mkfs.msdos mkfs.btrfs**  
**mkfs.ext2 mkfs.fat mkfs.ntfs**

On peut exécuter n'importe laquelle de ces commandes directement, mais souvent, il est simplement plus facile d'exécuter la commande mkfs.

La commande mkfs en elle-même a peu d'options.   
Cependant, chacune des commandes mkfs.\* dispose de plusieurs options qui modifient le mode de création du système de fichiers.   
Lors de la spécification d’options avec la commande mkfs, toute option ne faisant pas partie de la commande **mkfs** est transmise à la commande **mkfs.\***.

Par exemple, l'option -m de la commande mkfs.ext4 spécifie la quantité de système de fichiers (pourcentage du système de fichiers) réservée au super-utilisateur (la valeur par défaut est 5%).   
La commande suivante finit par exécuter la commande **mkfs.ext4** et passer l'option **-m** à cette commande.  
**>>** **sudo mkfs -t ext4 -m 10 /dev/sdb1**

**Astuce** …  
Les systèmes de fichiers ont de nombreuses options.   
Les options activées peuvent faire une grande différence pour les performances du système et parfois pour la sécurité.

Par exemple, l'**option -m** pour les systèmes de fichiers de type **ext** est utilisée pour **spécifier la quantité de système de fichiers réservée au super-utilisateur**. Le but de cette opération est de permettre à l'utilisateur **root** (ou aux démons s'exécutant en tant que root) de pouvoir stocker des fichiers sur un système de fichiers, même si des utilisateurs normaux remplissent le système de fichiers.

Lors du formatage, un **système de fichiers de 5 To séparé pour /home se voit attribué par défaut la valeur réservée au super-utilisateur de 5%**, sauf si on utilise l'option **-m**. Si on n'utilise pas cette option, lors de la création de ce système de fichiers, cela signifie que 250 Go seront réservés pour l'utilisateur root, utilisateur qui ne placera probablement jamais un seul fichier dans le système de fichiers /home.

# Liste des périphériques de type bloc

Dans le répertoire racine ou /, on retrouve un **ensemble** de répertoires systèmes dont la présence de certains est intéressants pour les périphériques de bloc.

* **/dev**  
  C'est dans ce répertoire que sont stockés les fichiers de configuration des périphériques (réels et virtuels) - (partitions, disques, cartes son, ports SCSI, ports USB, etc.).   
  Ces fichiers sont en mode "données brutes". Il est souvent nécessaire d'utiliser des programmes pour interpréter ces contenus (par exemple, en montant la partition /dev/hda1 dans /mnt/hda1 pour accéder aux fichiers).
* **/mnt**  
  C'est en général dans ce répertoire qu'on accède aux autres systèmes de fichiers   
  (autres partitions, CD/DVD, clés USB, serveurs de fichiers...).

Voici **quelques commandes différentes qui peuvent répertorier les disques durs**.  
Ce sont probablement les plus utilisées et les plus faciles pour effectuer le travail.

## Commande df

La commande df sous Linux est probablement l'une des plus utilisées. Il répertorie l'utilisation réelle de l'espace disque et peut fournir des informations sur les disques durs (ou l'espace disque actuel) utilisés dans l'ensemble du système.

La façon la plus courante de l'utiliser est avec l'argument -h qui signifie lisible par l'homme.

**>>** **df -h  
Sys. de fichiers Taille Utilisé Dispo Uti% Monté sur  
udev 967M 0 967M 0% /dev  
tmpfs 198M 6,1M 191M 4% /run  
/dev/sda1 28G 8,8G 18G 34% /**

Comme on peut le constater …

* la première colonne est le **nom logique du disque** (ou le nom avec lequel on peut le trouver sur le système) ;
* la deuxième colonne est la **taille de chacun d'eux** ;
* la troisième colonne est la **quantité actuellement utilisée** (en octets) ;
* la quatrième colonne indique la **quantité actuellement disponible** ;
* la cinquième colonne indique la **quantité utilisée** (en%)   
  et
* la sixième et dernière colonne indique **où est-elle** **physiquement montée dans le système Linux**.

Commutateurs les plus fréquents …

* **-a**  
   Affiche tous les systèmes de fichiers, y compris ceux de 0 blocs   
  (par exemple : proc, sysfs, usbfs et tmpfs) ;
* **-h**  
   Ajoute aux valeurs un M pour mébioctet (2^20 octets) pour que ce soit plus lisible ;
* **-H**
* **S**imilaire à l’option -h mais en mégaoctets (10^6 octets) ;
* **-T**   
   Affiche le type du système de fichier.

**Exemples d'utilisation**

**>>** **df –h**Affiche la **quantité d'espace disque utilisé en mébioctets** par les systèmes de fichiers.

**vdf /home**Affiche la **quantité d'espace disque utilisé par la partition /home** (si elle existe)

**>>** **df -T –h**   
Affiche le **nom des partitions et leur point de montage**.

## Commande lsblk

La commande lsblk (*list block devices*) illustre graphiquement la relation entre les disques et leurs partitions et fournit également les numéros et les points de montage principaux et mineurs.   
sblk est un peu plus sophistiqué car il répertorie tous les périphériques de bloc.   
Il permet d’afficher une liste simple de tous les disques disponibles.

**>>** **lsblk**

Dans la sortie précédente …

* **NAME** donne le nom du disque   
  (il est possible que cela ne soit pas cohérent et peut changer en fonction du disque qui a été monté en premier). sda, sdb, sdc,... sont les noms de périphérique de bloc et sda1, sda2, ... désignent les partitions sur chaque périphérique. ;
* **MAJ:MIN** désigne le numéro de périphérique majeur et mineur.  
  Le numéro majeur représente principalement le type de périphérique ou de bus utilisé par le matériel tandis que le numéro mineur indique une instance.  
  Par exemple pour les périphériques de type bloc le numéro majeur 8 correspond aux disques SCSI et le numéro mineur 0 associé indique le 1er disque détecté dans la chaîne SCSI. Le numéro mineur 16 et quant à lui associé au second disque dur et le numéro mineur 17 à la 1re partition de ce 2D disque dur. Le troisième disque aura le numéro mineur 32 et ainsi de suite ;
* **RM** indique si le périphérique est amovible (0 indique un périphérique fixe et 1 un périphérique mobile) ;
* **SIZE** indique l'espace de stockage utilisable sur le disque ;
* **RO** indique si le périphérique est en lecture seule, tel qu'un lecteur de DVD ou un lecteur protégé en écriture. ;
* **TYPE** indique s'il s'agit d'un disque ou d'une partition ;
* **MOUNTPOINT** indique le point de montage.

## Commande cfdisk

cfdisk est probablement l’utilitaire le plus avancé en terme graphique, car il est beaucoup plus visuel et interactif. Il permet dans un premier temps de lister tous les disques/partitions du système mais il permet également de les gérer en les sélectionnant puis en appliquant des actions telles que Supprimer, Redimensionner, Type (pour changer le type de partition) et Écrire Modifications apportées aux partitions.

**>>** **sudo cfdisk**

**Disque : /dev/sda**

**Taille : 30 GiB, 32212254720 octets, 62914560 secteurs**

**Étiquette : dos, identifiant : 0x008d738c**

**Périphérique Amorçage Début Fin Secteurs Taille Id Type**

**>> /dev/sda1 \* 2048 58722303 58720256 28G 83 Linux**

**/dev/sda2 58724350 62912511 4188162 2G 5 Étendue**

**└─/dev/sda5 58724352 62912511 4188160 2G 82 partition d'échan**

**lqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqk**

**x Type de la partition : Linux (83) x**

**x Attributs : 80 x**

**xUUID du système de fichiers : d2f61d08-627e-47df-a347-c45c2d0080b8 x**

**x Système de fichiers : ext4 x**

**x Point de montage : / (démonté) x**

**mqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqj**

**[ Amorçable ] [ Supprimer ] [Redimensionner] [ Quitter ]**

**[ Type ] [ Aide ] [ Écrire ] [ Sauvegarder ]**

**Quitter le programme sans écrire les modifications**

cfidsk fournit également des informations conviviales sur chaque partition et disque car il indique où commence et se termine chaque cylindre de partition, la quantité de secteurs utilisés par chacun et la taille complète de chacun avec son type.

## Commande du (*disk usage*)

La commande du affiche l'espace disque utilisé par les fichiers et/ou les répertoires.

Commutateurs les plus fréquents …

* **-a**  
  Affiche tous les fichiers et pas uniquement les répertoires ;
* **-s**  
  Affiche le total sans lister les différents fichiers ;
* **-c**  
   Effectue un total après avoir tout affiché ;
* **-h**  
  Ajoute un suffixe correspondant à l'unité (K, M, G) ;
* **-H**  
  Idem que -h mais en puissance de 10.

**Exemple d'utilisation**

**>>** **du -hs dir**   
Affiche la **taille du répertoire dir ou du répertoire courant** si dir est omis.

**>>** **du -ch /home/MonUtilisateur**Affiche la **taille des répertoires contenus dans /home/<Utilisateur>** en utilisant un suffixe puis le total ;

**>>** **du -sm ~/Images/\*.jpg**Affiche la **taille totale des fichiers JPEG contenus dans le répertoire ~/Images**.

## Commande pydf

La commande pydf (python df car c'est un script python) fournit également un affichage d'utilisation du disque très utile montrant des points de montage.

## Commande fdisk

La commande fdisk peut **afficher des détails sur les partitions de disque** et **utilise des nombres très différents**.   
**>>** **fdisk -l**

## Commande parted

La commande parted **affiche les informations de partition dans un format différent** …  
**>>** **parted -l**

## Commande blkid

La commande blkid permet de **localiser et afficher les attributs de périphérique de bloc**.

Enfin, il est toujours possible d’afficher le contenu du **fichier /proc/partitions** …  
**>>** **cat /proc/partitions**

# Espace d’échange

L'espace d'échange sous Linux est utilisé lorsque la quantité de mémoire physique (RAM) est pleine.

Si le système a besoin de plus de ressources mémoire et que la RAM est pleine, des pages inactives de la mémoire sont déplacées vers l'espace de swap.

Bien que l'espace d'échange puisse aider les hôtes avec une petite quantité de RAM, il ne doit pas être considéré comme un remplacement pour plus de RAM. L'espace d'échange est situé sur les disques durs, qui ont un temps d'accès beaucoup plus lent que la mémoire physique.

L'espace d'échange peut être une partition d'échange dédiée (recommandée), un fichier d'échange ou une combinaison des deux.

Anciennement, la quantité recommandée d'espace de swap a augmenté linéairement avec la quantité de RAM dans le système. Mais comme la quantité de mémoire dans les systèmes modernes a augmenté en centaines de gigaoctets, il est désormais reconnu que la quantité d'espace de swap dont un système a besoin est fonction de la charge de travail de la mémoire qui s'exécute sur ce système. Cependant, étant donné que l'espace d'échange est généralement désigné au moment de l'installation et qu'il peut être difficile de déterminer à l'avance la charge de travail de la mémoire d'un système.

Il est généralement recommandé de déterminer l'échange système à l'aide du tableau suivant ...

## Recommandations

|  |  |
| --- | --- |
| Quantité de mémoire vide du système | Espace d’échange recommandé |
| 4 gigaoctets de mémoire vive ou moins | **Un minimum de 2 gigaoctets d’espace d’échange** |
| 4 gigaoctets à 16 gigaoctets de mémoire vive | **Un minimum de 4 gigaoctets d’espace d’échange** |
| 16 gigaoctets à 64 gigaoctets de mémoire vive | **Un minimum de 8 gigaoctets d’espace d’échange** |
| 64 gigaoctets à 256 gigaoctets de mémoire vive | **Un minimum de 16 gigaoctets d’espace d’échange** |
| 256 gigaoctets à 512 gigaoctets de mémoire vive | **Un minimum de 32 gigaoctets d’espace d’échange** |

# Point de montage

Un point de montage est un répertoire à partir duquel sont accessibles les données se trouvant sous forme d'un système de fichiers sur une partition de disque dur ou un périphérique.   
Plus simplement, c'est le répertoire qui permet d'accéder au contenu d'un disque dur, clé USB, lecteur DVD, ou autre périphérique de stockage.

## Commande mount

La commande **mount** peut être utilisée pour **monter** un système de fichiers et pour afficher les systèmes de fichiers actuellement montés. Lorsqu'elle est utilisée sans argument, la commande **mount** affiche les systèmes de fichiers montés ainsi que certains attributs de montage (également appelés options de montage).

**>>** **mount**(certaines lignes ont été omises)  
**sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)  
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)  
udev on /dev type devtmpfs (rw,nosuid,relatime,size=1004720k,nr\_inodes=251180,mode=755)  
/dev/sda1 on / type ext4 (rw,relatime,errors=remount-ro,data=ordered)**

Chaque ligne décrit un périphérique monté et est divisée en quatre champs, soulignés en gras …  
périphérique **sur le** **point de montage** **de type** **type** **avec les** **options de montage**

### Synopsis de la commande mount

**mount [options] <-t type> [-o options de montage] <périphérique> <point de montage>**

En l'occurrence, on peut monter une partition sur **/mnt** ou tout autre point de montage que l’on a choisi   
(ne pas oublier que le point de montage doit exister) …  
**>>** **mount -t ext2 /dev/hdb1 /mnt**

L'**option -t** sert à spécifier quel type de système de fichiers la partition est censée héberger.   
Parmi les systèmes de fichiers que l’on rencontrera le plus souvent, on trouve …

* **ext2FS/ext3FS/ext4FS** ;
* **VFAT** (pour toutes les partitions DOS/windows : FAT 12, 16 ou 32)   
  et
* **ISO9660** (système de fichiers des cédéroms).

Si on ne spécifie aucun type, mount essaiera et trouvera quel système de fichier est hébergé pour la partition en lisant le superblock. Il échoue rarement.

L'**option -o** sert à spécifier une ou plusieurs options de montage.   
Ces options dépendent du système de fichiers utilisé.

## Commande umount

Pour démonter un système de fichiers, on peut spécifier la commande **umount** suivie du **point de montage**   
ou du **nom** **du périphérique**.

**Synopsis de la commande umount**  
**umount <point de montage | périphérique>**

Par exemple, si le périphérique **/dev/sda1** est monté dans le répertoire **/boot**, l’une des commandes suivantes doit démonter le système de fichiers …

**>>** **umount /boot**  
**>>** **umount /dev/sda1**

## Montage du système de fichiers manuellement

Pour monter un système de fichiers, il faut spécifier deux arguments à la commande **mount ...**

* le périphérique à monter   
  et
* le point de montage.

Par exemple, les commandes suivantes montrent d'abord que /dev/sda1 est monté, démonté pour enfin être remonté sous la partition /boot …

**>>** **mount | grep /dev/sda1  
/dev/sda1 on /boot type ext4 (rw,usrquota,grpquota)**  
**>>** **umount /dev/sda1  
>>** **mount /dev/sda1 /boot  
>>** **mount | grep /dev/sda1  
/dev/sda1 on /boot type ext4 (rw)**

En regardant de plus près, on constatera une différence entre la manière dont le périphérique est monté à l'origine et la manière dont il a été monté une seconde fois.   
Les **options usrquota** et **grpquota** sont manquantes dans la sortie de la deuxième commande de montage.   
La raison en est que le périphérique /dev/sda1 a été monté à l'origine avec ces options activées à la suite d'une entrée dans le fichier **/etc/fstab**.

Les autres options de montages sont explicitées plus loin.

Pour activer les options de montage manuellement …

**>>** **mount | grep /dev/sda1  
/dev/sda1 on /boot type ext4 (rw)**  
**>>** **umount /dev/sda1  
>>** **mount -o usrquota,grpquota /dev/sda1 /boot  
>>** **mount | grep /dev/sda1  
/dev/sda1 on /boot type ext4 (rw,usrquota,grpquota)**

**Remarque** …  
Une autre option de montage utile est l'option ro, option qui **permet de monter un système de fichiers en lecture seule**. Si le système de fichiers est actuellement monté en lecture-écriture, on peut le modifier en lecture seule en exécutant la commande **mount -o remount, ro /dev/<Nom du périphérique>**.

Comme mentionné précédemment, il existe de nombreuses options de montage et ces options peuvent s'appliquer à plusieurs types de systèmes de fichiers ou être spécifiques à un type de système de fichiers individuel.

Une autre option de ligne de commande de la commande **mount** est l'option **-t**, option qui **permet de spécifier le type de système de fichiers**.   
Dans la plupart des cas, la commande **mount** est suffisamment intelligente pour sélectionner le type de système de fichiers approprié en sondant le système de fichiers avant de le monter.

Toutefois, si on doit **spécifier un type de système de fichiers** on utilisera le **commutateur -t** …  
**>>** **mount -t ext4 /dev/sda1/boot**

## Montage automatique des systèmes de fichiers

Plusieurs systèmes de fichiers sont montés automatiquement au démarrage.   
Les systèmes de fichiers montés dépendent des paramètres du fichier /etc/fstab.

Le fichier **/etc/fstab** est une liste des systèmes de fichiers qui seront montés automatiquement à l'initialisation du systèm**e** sauf en cas d'utilisation de l'**option noauto**.

Il est possible d’éditer le fichier /etc/fstab pour ajouter ses propres volumes et partitions et choisir dans quel répertoire leur contenu apparaîtra.

Une ligne contient, dans l'ordre …

* le **périphérique hébergeant le système de fichiers** ;
* le **point de montage** ;
* le **type du système de fichiers** ;
* les **options de montage** ;
* le **drapeau de sauvegarde par l'utilitaire** dump ;
* l'**ordre** **de la vérification par fsck** (*FileSystem ChecK*, vérification des systèmes de fichiers).

### Drapeau de sauvegarde dump

La cinquième colonne prend la valeur 1 pour les partitions visées par une sauvegarde au moyen de l'utilitaire dump ou 0 pour les autres.   
Le drapeau de sauvegarde dump est utilisé par l'utilitaire dump pour décider du moment où sont effectuées les sauvegardes.   
Quand il est installé, dump vérifie le chiffre inscrit et décide si le système de fichiers doit être sauvegardé.   
Les valeurs possibles sont 0 et 1 …

* **0**, dump va ignorer le système de fichier ;
* **1**, dump fera une sauvegarde.

Puisque la plupart des systèmes n'auront pas l’utilitaire dump installé, ce champ aura la valeur 0.

### Ordre de vérification avec fsck <pass>

La sixième colonne concerne l'ordre de vérification des fichiers par l'outil fsck au démarrage du système (*boot*).

Elle prend la valeur …

* **1** pour la partition racine (/) ;
* **2** pour les autres partitions   
  ou
* **0** si la partition ne doit pas être vérifiée par fsck (le SWAP et les partitions Windows par exemple).

Dans les faits, **/etc/fstab** (*File System Table*) permet à la commande mount de monter les systèmes de fichiers utilisés par le système au cours de son initialisation.

La commande **mount** sait ce qui est actuellement monté car ces informations sont stockées dans le fichier /etc/mtab...  
**>>** **less /etc/mtab**(certaines lignes ont été omises)  
**sysfs /sys sysfs rw,nosuid,nodev,noexec,relatime 0 0  
proc /proc proc rw,nosuid,nodev,noexec,relatime 0 0  
udev /dev devtmpfs rw,nosuid,relatime,size=1004720k,nr\_inodes=251180,mode=755 0 0  
/dev/sda1 / ext4 rw,relatime,errors=remount-ro,data=ordered 0 0**

Bien que ces deux fichiers contiennent des informations sur les systèmes de fichiers actuellement montés, il existe certaines différences qu’il est intéressant de connaître …

* Le fichier /etc/mtab est géré par les commandes mount et umount ;  
  La commande **mount** possède l’**option -n** qui signifie … ne pas mettre à jour le fichier /etc/mtab.  
  Il est donc possible que ce fichier ne soit pas précis ;
* Le fichier /proc/mounts est géré par le noyau et a plus de chance d'être précis ;
* Le **fichier /proc/mounts** contient généralement **plus d'informations**.

### Descripteurs de périphérique

Si on ajoute un nouveau système de fichiers, on devra examiner avec soin le descripteur de périphérique qui est utilisé pour monter ce système de fichiers.   
Par exemple, on a ajouté un nouveau disque dur au système, créé une partition portant le nom de périphérique **/dev/sdb1** et placé un système de fichiers ext4 sur cette partition.   
il faudra ensuite ajouter une ligne dans le fichier **/etc/fstab** …  
**/dev/sdb1/dataext4default 0 1**

Cela fonctionnera très probablement très bien à partir de maintenant, mais il existe un problème potentiel : les noms de périphérique peuvent changer.   
Si on ajoute ou retire un ou des disques durs de ou vers le système, le programme du micrologiciel (le BIOS, par exemple) peut penser que le nouveau disque dur est le premier disque, ce qui entraîne la modification du nom de périphérique des disques existants.   
Une suppression de partitions sur un périphérique peut également entraîner la **renumérotation des partitions existantes**.

Ainsi, au lieu d'utiliser des noms de périphérique pour les partitions ordinaires, on préférera utiliser des étiquettes (*labels*). Une étiquette est un nom que l’on attribue à une partition. Si l’ordre des périphériques change, l'étiquette reste la même pour cette partition, ce qui en fait une méthode plus stable pour un montage.

Pour créer une étiquette, on utilise la commande **e2label** …  
**>>** **e2label /dev/sdb1 data**

**Remarque** …  
Des étiquettes peuvent également être attribuées au moment de la création du système de fichiers.

On utilisera alors la syntaxe suivante dans le fichier **/etc/fstab** …  
**LABEL="data" /data ext4 defaults 0 1**

Il est possible d’afficher les étiquettes à l'aide de la commande e2label ou de la commande blkid …  
**>>** **e2label /dev/sda2** data  
**>>** **blkid | grep sda2**  
**/dev/sda2: UUID="974e2644-aaac-4a48-9bc7-c86d296e97f9" TYPE="ext4" LABEL="data"**

Bien que l’utilisation d’étiquettes pour monter des périphériques soit préférable à l’utilisation de noms de périphériques, cela n’est toujours pas idéal. Si on déplace maintenant ce disque du système actuel et l'ajoute à un autre système, il est possible qu'il y ait un conflit d'étiquettes. Si le nouveau système comporte également une partition portant le libellé data, des problèmes se posent.

La solution la plus stable consiste à utiliser des numéros UUID.   
L'identificateur unique universel est un numéro attribué au périphérique lors de sa création.   
En raison de la taille de ce nombre (128 bits) et de la manière dont il est généré, il est pratiquement impossible que deux périphériques aient le même numéro.

La commande blkid affiche l’UUID du périphérique …  
**>>** **blkid | grep sda1**  
ou  
**>>** **blkid /dev/sda1  
/dev/sda1: UUID="235b8772-0645-4c7c-a88a-18c287199fbd" TYPE="ext4" PARTUUID="5530d6aa-01"**

Enfin, il est important de noter que tous les périphériques ne nécessitent pas l'utilisation d'UUID ou d'étiquettes. Voici une ligne type du fichier **/etc/fstab** …  
**/dev/mapper/vg\_livecd-lv\_swap swap swap defaults 0 0**

Le périphérique **/dev/mapper/vg\_livecd-lv\_swap** est un volume logique.   
Les **noms** d’unités de volume logique ne changent que si l’administrateur modifie le nom. D’autres périphériques, tels que les lecteurs réseau et les périphériques logiciels RAID, n’ont également pas besoin d’UUID ni d’étiquettes.

### Options de montage

Le comportement d'un système de fichiers dépend en partie des options de montage fournies.   
Des dizaines de différentes options de montage sont disponibles. Les options les plus importantes à connaître sont celles liées aux mots-clés par défaut. Lorsque l’on spécifie les valeurs par défaut comme option de montage dans le fichier /etc/fstab, on spécifie réellement les options **rw**, suid, dev, exec, auto, nouser, async et relatime.

Le tableau suivant décrit ces options plus en détails …

|  |  |
| --- | --- |
| Option | Description |
| defaults | Correspond à … rw, **suid**, **dev**, **exec**, **auto**, **nouser**, **async** |
| rw/ro | Montage en lecture/écriture (par défaut) ou lecture seule |
| suid/nosuid | Les bits SUID et SGID sont pris en compte (ou non)  Relatif aux droits donnés aux exécutables sur la partition |
| dev/nodev | Interprète ou non les fichiers spéciaux de périphériques présents sur le système (par défaut) |
| exec/noexec | Autorise l'exécution des programmes (par défaut) |
| auto/noauto | Montage automatique (ou non) lors d'un appel **mount -a** (par défaut) |
| nouser | Seul le compte root peut monter/démonter le système de fichier (par défaut) |
| netdev | Le système de fichiers est sur un hôte qui nécessite un accès réseau Cela indique au système d'attendre que la configuration réseau soit active avant de procéder au montage |
| async | Montage asynchrone (par défaut) |
| atime/noatime | Inscrit (ou non) la date d'accès (préférez **noatime** pour les SSD) |
| Sw | Spécifique à l'activation des partitions swap |
| discard | Active le TRIM1 sur les partitions SSD (à rajouter manuellement) |

**Remarque** …  
À propos de l’option **sync**, pour les distributions Linux modernes, cette option est rarement exécutée.   
L'exécution de la commande sync force l'écriture de toutes les données du système de fichiers stockées en mémoire sur le disque dur.   
Cela n’est vraiment plus nécessaire (et ce depuis longtemps), car la commande **umount** synchronise correctement le système de fichiers avant de le démonter.

### Point de montage avec systemd

Pour lancer un point de montage au démarrage avec systemd, on doit créer un fichier unité dans **/etc/systemd/system**.   
Le **fichier d'unité doit être nommé d'après le point de montage**.   
Dans cet exemple général, le fichier s'appelle: **data.mount** car le point de montage est /data.

**>>** less /etc/systemd/system/data.mount  
  
 [Unit]  
 Description=Mon système de fichiers  
  
 [Mount]  
 What=/dev/sdb1  
 Where=/data  
 Type=ext4  
 Options=defaults  
  
 [Install]  
 WantedBy=multi-user.target

**Attention** …  
Le nom du fichier de configuration systemd (**/etc/systemd/system/data.mount**) est en fonction du répertoire local (**/data**).  
Si le répertoire local est /mnt/data, le nom du fichier doit être **/etc/systemd/system/mnt-data.mount**.

Ce nom de fichier systemd doit **obligatoirement** refléter le nom du répertoire.

Il faut par la suite **actualiser le démon systemd** …

**>> sudo systemctl daemon-reload**

Comme dernière étape, il faut **démarrer le point de montage** …  
**>> sudo systemctl start samba-linux.mount**

Pour que le **point de montage soit activé à chaque démarrage** de la station …

**>> sudo systemctl enable samba-linux.mount**

**Created symlink /etc/systemd/system/multi-user.target.wants/data.mount → /etc/systemd/system/data.mount.**

**Point de montage .automount**

Il est également **possible de monter un partage uniquement à la demande**.

On définit l’unité de montage comme ci-dessus mais on **ne l'active pas à l’aide de systemctl**.

On doit **créer une unité de montage automatique** /etc/systemd/system/**data.automount**.   
L'unité de montage automatique démarre l'unité de montage (data.mount) à la demande.

On active l'unité de montage automatique. Cela va créer une entrée autofs …  
**>> sudo systemctl enable data.automount**

Pour vérifier le bon fonctionnement …  
**>> mount | grep systemd**

# Annexe 01 Système de fichiers Linux

Il **n'est pas possible de créer des fichiers et des répertoires directement dans une partition**.   
Avant d'utiliser une partition pour le stockage de données, on doit **créer un système de fichiers**.

Le système de fichiers est un **conteneur logique** utilisé pour **stocker les fichiers et les répertoires**.

On peut créer un **type de système de fichiers distinct dans chaque partition** ou encore utiliser le même type de système de fichiers dans toutes les partitions.

Toute partition doit être formatée avec un système de fichiers avant de pouvoir être utilisée pour le stockage de données.

Linux prend en charge plusieurs types de systèmes de fichiers.

* **ext** (Extended File system ou système de fichiers étendu)   
  ext est le premier système de fichiers Linux. Il était utilisé dans les premières versions de Linux.
* **Système de fichiers ext2**ext2 est la seconde génération du système de fichiers ext.   
  Il fournit des fonctionnalités très basiques du système de fichiers. Il a été développé en 1980.
* **Système de fichiers ext3**ext3 est la troisième génération du système de fichiers ext.   
  Il comprend plusieurs fonctionnalités améliorées …   
  Il prend en charge des partitions jusqu'à 16 To et des fichiers jusqu'à la taille de 2 téraoctets ;   
  Il prend en charge jusqu'à 32 000 sous-répertoires ;   
  Il s'agit du premier système de fichiers de la série ext qui prend en charge le mécanisme de journalisation.
* **Système de fichiers ext4**ext4 est la quatrième génération du système de fichiers ext.   
  Il prend en charge les systèmes de fichiers jusqu'à 1 exaoctets l   
  Il prend en charge les fichiers jusqu'à 16 téraoctets ;   
  Il prend en charge des répertoires illimités ;   
  Il utilise une série de blocs physiques contigus sur le disque dur appelés extensions. Les extensions sont utilisées pour améliorer les performances des fichiers très volumineux.
* **Système de fichiers XFS** (*X-File System*)   
  Ce système de fichiers a été développé par Silicon Graphics pour leur version d'UNIX. Il a été adopté par la suite par la plupart des distributions Linux.   
  XFS est basé sur une étendue 64 bits ;   
  Il utilise la journalisation pour les opérations de métadonnées ;   
  Il prend en charge les systèmes de fichiers et les fichiers de tailles allant jusqu'à 8 exaoctets.   
  Le seul inconvénient de ce système est qu'il ne prend pas en charge le rétrécissement comme EXT3 et EXT4.
* **Système de fichiers VFAT** (*Virtual File Allocation Table*)   
  Il s'agit de la version améliorée du système de fichiers FAT. Il a été développé en 1995 pour Windows 95. Linux n'utilise ce système de fichiers pour aucune opération mais si besoin il peut lire et écrire des fichiers écrits dans ce format. Cette prise en charge a été ajoutée sous Linux afin qu'un utilisateur puisse échanger des données entre Linux et Windows à l'aide d'un périphérique externe tel qu'une clé USB ou un disque compact.
* **Système de fichiers NTFS** (*New Technology File System*)   
  Il s'agit de la version améliorée de VFAT. Ce système de fichiers est le système de fichiers par défaut du système Windows. Tout comme VFAT, Linux peut lire et écrire des fichiers écrits dans ce format mais il n'utilise pas ce format à ses propres fins.
* **Btrfs** (souvent prononcé *ButterFS*) est un système de fichiers sous licence GPL.   
  Il apporte certaines fonctionnalités inédites comme une meilleure gestion de l'espace occupé par les petits fichiers, la possibilité de créer des instantanés (snapshots) en lecture seule ou en lecture-écriture et la possibilité de créer des sous-volumes.

# Annexe 02 Autres configurations de disques

**Espace d’échange** (swap)   
L'espace d'échange est une portion spéciale d’un disque dur utilisée comme mémoire temporaire. Cet espace peut être alloué en tant que partition de swap séparée, partition LVM ou en tant que fichier (le fichier est utilisé uniquement pour étendre l'espace de swap disponible).   
L'espace d'échange n'est utilisé qu'en cas de manque de mémoire physique.   
En situation de pénurie, le système déplace les données récemment inutilisées de la mémoire vers l'espace d'échange. Lorsque cela est nécessaire, le système ramène ces données du swap à la mémoire.   
C'est le moyen pratique d'améliorer l'utilisation de la mémoire du noyau.

**LVM** (Logical Volume Manager)   
Le schéma de partition classique est de nature fixe. Cela signifie qu'une fois la taille de la partition créée, elle ne peut pas être modifiée ultérieurement. Il n’est pas possible d’ajouter de l'espace supplémentaire dans une partition qui est remplie de données.   
De la même manière, il n'est pas possible de réduire une partition qui a beaucoup d'espace libre inutilisé.   
LVM résout non seulement adresse ces lacunes, mais offre également plusieurs autres avantages par rapport à l'approche classique. LVM est de nature flexible. Il est possible de réduire ou d’agrandir une partition selon les besoins.

**RAID**RAID est un mécanisme permettant de combiner plusieurs disques durs en une baie (ou agrégat) pour obtenir des performances et une redondance. La matrice de disques apparaîtra comme une seule unité de stockage logique ou lecteur pour l'ordinateur. RAID améliore la vitesse d'accès aux données et augmente la capacité de stockage des données à l'aide d'un seul disque virtuel.   
Il existe deux types de RAID : RAID logiciel et RAID matériel.   
Le RAID matériel est construit à partir de disques durs indépendants et d'un dispositif de contrôleur RAID. Il n'utilise aucune ressource système.   
Le RAID logiciel est construit à partir de disques durs connectés. Il utilise toutes les ressources du système.

# Annexe 03 Utilitaire lshw

lshw est un petit outil pour **extraire des informations détaillées sur la configuration matérielle de l’hôte**.

lshw peut indiquer la **configuration exacte de la mémoire**, la **version du micrologiciel**, la **configuration de la carte mère**, la **version et la vitesse du processeur**, la **configuration du cache**, la **vitesse du bus**, ...

**Remarque** …  
Il est possible que l’utilitaire ne soit pas intallé sur le système  
**>>** **sudo** **apt install lshw**

On peut l’utiliser de la manière suivante …  
**>> sudo** **lshw -c disk**ou   
**>> sudo** **lshw -short -c disk**

# Annexe 04 Unités de mesure

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Valeur | Unité | Symbole | Valeur | Unité | Symbole |
| 1024 (1) | Kibibyte | KiB | 10001 | Kilobyte | KB |
| 1024 (2) | Mebibyte | MiB | 10002 | Megabyte | MB |
| 1024 (3) | Gibibyte | GiB | 10003 | Gigabyte | GB |
| 1024 (4) | Tebibyte | TiB | 10004 | Terabyte | TB |
| 1024 (5) | Pebibyte | PiB | 10005 | Petabyte | PB |
| 1024 (6) | Exbibyte | EiB | 10006 | Exabyte | EB |
| 1024 (7) | Zebibyte | ZiB | 10007 | Zettabyte | ZB |
| 1024 (8) | Yobibyte | YiB | 10008 | Yottabyte | YB |

# Annexe 05 Création d'un fichier d'échange

**Ajout d’un fichier d'échange**

Dans un premier temps, il faut déterminer la taille du nouveau fichier d'échange en mégaoctets et on multiplie par 1024 pour déterminer le nombre de blocs.   
Par exemple, la taille de bloc d'un fichier d'échange de 64 Mo est 65536.

À l'invite du shell en tant que root, on entre la commande suivante avec un nombre égal à la taille de bloc souhaitée …  
**>>** **dd if=/dev/zero of=/swapfile bs=1024 count=65536**

On modifie par la suite les autorisations du fichier nouvellement créé …  
**>>** **chmod 0600 /swapfile**

Comme troisième étape, on configure le fichier d'échange …  
**>>** **mkswap /swapfile**

Pour activer le fichier d'échange immédiatement mais pas automatiquement au démarrage …  
**>>** **swapon /swapfile**

Pour l'activer au démarrage, on modifie le fichier /etc/fstab pour inclure l'entrée suivante …  
**/swapfile swap swap defaults 0 0**

Au prochain démarrage du système, le nouveau fichier d'échange sera automatiquement activé.

Après avoir ajouté le nouveau fichier d'échange et l'avoir activé, on vérifie qu'il est activé en affichant la sortie de la commande cat /proc/swaps ou de la commande free.

**Suppression d'un fichier d'échange**

Pour **supprimer un fichier d'échange**, à l'invite du shell en tant que root, on exécute la commande suivante pour **désactiver le fichier d'échange** (où /swapfile est le fichier d'échange) …  
**>>** **swapoff -v /swapfile**

Il faut par la suite **supprimer son entrée du fichier /etc/fstab**.

Comme dernière étape on supprime le fichier sur disque …  
**>>** **rm /swapfile**

**Déplacement de l’espace d’échange**

Pour déplacer l'espace d’échange d'un emplacement à un autre, il faut supprimer l'espace d’change existant, pour, par la suite, en ajouter un nouveau.