

Gestion des disques  
Structure et gestion du système de fichiers à l’aide des commandes   
et utilitaires usuels

Formation continue  
Collège de Maisonneuve  
2030, boulevard Pie IX,  
Montréal (Québec) H1X 3M3  
514.251-7131 poste 4800

Document proposé par Louis Savard, M. Ing. TI  
Dernière révision : Mai 2021

# Table des matières

[Table des matières 2](#_Toc73250943)

[Gestion des disques 4](#_Toc73250944)

[Introduction 4](#_Toc73250945)

[BIOS et UEFI 4](#_Toc73250946)

[Différences entre le BIOS et l'UEFI 4](#_Toc73250947)

[Schéma de partition MBR 5](#_Toc73250948)

[Schéma de partition GPT 6](#_Toc73250949)

[Différence entre MBR et GPT 6](#_Toc73250950)

[Noms de périphériques de disque courants 6](#_Toc73250951)

[Partitionnement 8](#_Toc73250952)

[fdisk 9](#_Toc73250953)

[parted 9](#_Toc73250954)

[sfdisk 10](#_Toc73250955)

[Formatage d’une partition ou d’un volume 11](#_Toc73250956)

[Liste des périphériques de type bloc 12](#_Toc73250957)

[df 12](#_Toc73250958)

[lsblk 13](#_Toc73250959)

[cfdisk 14](#_Toc73250960)

[du (disk usage) 14](#_Toc73250961)

[pydf 15](#_Toc73250962)

[fdisk 15](#_Toc73250963)

[parted 15](#_Toc73250964)

[blkid 15](#_Toc73250965)

[Espace d’échange 15](#_Toc73250966)

[Point de montage 16](#_Toc73250967)

[Commande mount 16](#_Toc73250968)

[Synopsis de la commande mount 16](#_Toc73250969)

[Commande umount 17](#_Toc73250970)

[Montage du système de fichiers manuellement 17](#_Toc73250971)

[Montage automatique des systèmes de fichiers 18](#_Toc73250972)

[Drapeau de sauvegarde dump 18](#_Toc73250973)

[Ordre de vérification avec fsck <pass> 19](#_Toc73250974)

[Options de montage 19](#_Toc73250975)

[Point de montage avec systemd 20](#_Toc73250976)

[Système de fichiers Linux 22](#_Toc73250977)

[LVM, une autre manière de partitionner 23](#_Toc73250978)

[Introduction à LVM 23](#_Toc73250979)

[Éléments de base de LVM 24](#_Toc73250980)

[Arborescence de répertoires et systèmes de fichiers 51](#_Toc73250981)

[Droits – Fichiers Linux 53](#_Toc73250982)

[Droits du système de fichiers 53](#_Toc73250983)

[Sticky bit 54](#_Toc73250984)

[Mode numérique 55](#_Toc73250985)

[Droits pour les groupes d’utilisateurs (*group*) 56](#_Toc73250986)

[Liens 57](#_Toc73250987)

[Listes de contrôle 59](#_Toc73250988)

[Exemple pratique 59](#_Toc73250989)

[Mise en place 60](#_Toc73250990)

[Noyau 60](#_Toc73250991)

[Commandes 61](#_Toc73250992)

[Commande setfacl 61](#_Toc73250993)

[Droits par défaut et héritage des droits étendus 62](#_Toc73250994)

[Retirait de permissions 62](#_Toc73250995)

[Masque 62](#_Toc73250996)

[Commande getfacl 64](#_Toc73250997)

[Commandes ls, cp et mv 64](#_Toc73250998)

[Sauvegarde des données 65](#_Toc73250999)

[Syntaxe de setfacl 65](#_Toc73251000)

[Conclusion 65](#_Toc73251001)

[Annexe 01 Autres configurations de disques 72](#_Toc73251002)

[Annexe 02 Utilitaire lshw 73](#_Toc73251003)

[Annexe 03 Unités de mesure 73](#_Toc73251004)

[Annexe 04 Création d'un fichier d'échange 74](#_Toc73251005)

# 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.

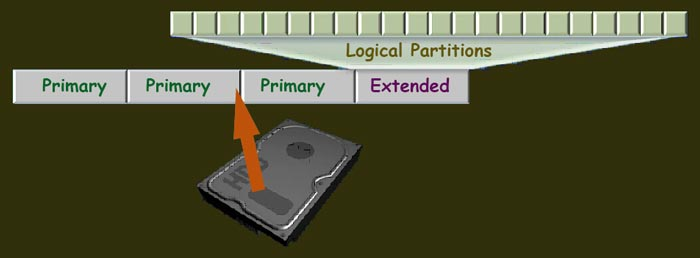
### Schéma de partition MBR

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 a plusieurs disques, la lettre disponible suivante sera utilisée.   
Par exemple sdb représente le deuxième disque SCSI, sdc représente le troisième disque **SCSI** 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 alpha-numériques 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.  
**>>** **sudo** **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**

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

**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.

|  |
| --- |
| Fichier |
| 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.

## 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 …  
**>>** **sudo** **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.

**Command (m for help): m**

**Commande 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.

## 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.

**>>** **sudo** **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 lqa 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.

## 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.

**>>** **sudo** **sfdisk -l**

Toutes ces commandes devraient permettre de voir au moins les volumes logiques, les partitions et les disques durs que 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 la commande **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.btrfs mkfs.ext2 mkfs.ext4 mkfs.xfs  
mkfs.bfs mkfs.cramfs mkfs.ext3 mkfs.minix**

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 **ensem**ble 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 À  
  (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 faire le travail.

## 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.

**>>** **sudo** **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**.

Options les plus fréquentes …

* **-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  
  Similaire à l’option -h** mais en **mégaoctets** (10^6 octets) ;
* **-T**   
   Affiche le **type du système de fichier**.

**Exemples d'utilisation**

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

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

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

## 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.

**>>** **sudo** **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 détectée 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.

## 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 modifier 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.

## du (disk usage)

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

Options les plus fréquentes

* **-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**

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

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

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

## 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.

## fdisk

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

## parted

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

## 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 ...

|  |  |
| --- | --- |
| 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).

**>>** **sudo** **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) …  
**>>** **sudo** **mount -t ext4 /dev/sdb1 /mnt/data01**

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** ;
* **btrfs** ;
* **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 …

**>>** **sudo** **umount /boot**  
**>>** **sudo** **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 …

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

**>>** **sudo** **mount | grep /dev/sda1  
/dev/sda1 on /boot type ext4 (rw)**  
**>>** **sudo** **umount /dev/sda1  
>>** **sudo** **mount -o usrquota,grpquota /dev/sda1 /boot  
>>** **sudo** **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 l’**option -t** …  
**>>** **sudo** **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...  
**>>** **sudo** **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.

### 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 for**ce 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 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: mnt.mount** car le point de montage est /mnt.

**>> less /etc/systemd/system/mnt-data.mount**

[**Unit]**

**Description=Mon système de fichiers**

**[Mount]**

**What=/dev/sdb1**

**Where=/mnt/data**

**Type=ext4**

**Options=defaults**

**[Install]**

**WantedBy=multi-user.target**

Après avoir créé le fichier d'unité, il **devrait être activé avec la commande** …

**>> systemctl start <Point de montage systmd>**

**>> systemctl start mnt-data.mount**

Pour **vérifier le point de montage activé** …

**>> mount | grep data**

**/dev/sdb1 on /data type ext4 (rw,relatime,data=ordered)**

Pour **activer le montage automatique au démarrage** …

**>>** **systemctl enable <Point de montage systemd>**

**>>** **systemctl enable mnt-data.mount**

**Attention** …

Dans cet exemple, le **répertoire local qui sert de point de montage est /data**.

Le **fichier de configuration dans le répertoire /ect/systemd/system doit obligatoirement porter le nom du répertoire local**.

Dans l’exemple précédent, ce fichier doit se nommer … **/etc/systemd/system/data.mount**.

# 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 les systèmes de fichiers jusqu'à 16 To ;   
  Il prend en charge 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 btrfs (b-tree file system)**   
  btrfs est le successeur de ext4 et ext3. Il est un système de fichiers basé sur le Copy-On-Write.  
  btrfs offre les fonctionnalités absentes d'autres systèmes de fichiers tels que les instantanés (snapshots) et les sommes de contrôle. Ces caractéristiques sont importantes pour les systèmes Linux, serveurs comme postes clients, car les tailles de stockage comme les configurations tendent à augmenter et à se complexifier.  
  SUSE (openSUSE) et RHEL propose btrfs par défaut pour la partition racine (/) afin d'assurer d’une sécurité accrue et laisse le choix entre ext4 et XFS (plus rapide) pour /home.
* **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.

# LVM, une autre manière de partitionner

## Introduction à LVM

LVM (*Logical Volume Manager*, ou gestionnaire de volumes logiques en français) permet la création et la gestion de volumes logiques sous Linux.   
La gestion par volumes logiques est à la fois **une méthode et un logiciel de gestion de l'utilisation des espaces de stockage d'un hôte**.  
Il permet de **gérer**, **sécuriser** et **optimiser** de manière souple les **espaces de stockage en ligne** dans les systèmes d'exploitation de type Linux.   
L'utilisation de volumes logiques **remplace en quelque sorte le partitionnement des disques**.

LVM est un système beaucoup plus souple, qui permet par exemple de **diminuer la taille d'un système de fichier pour pouvoir en agrandir un autre**, sans se préoccuper de leur emplacement sur le disque.

**Avantages de LVM**

* Il n'y a **pas de limitations « étranges »** comme avec les partitions   
  (primaire, étendue, ...) ;
* On ne se **préoccupe plus de l'emplacement exact des données** ;
* On peut conserver quelques giga-octets de libres afin de pouvoir les ajouter n'importe où et n'importe quand ;
* Les **opérations de redimensionnement deviennent quasiment sans risques**, contrairement au redimensionnement des partitions.

**Inconvénients de LVM**

* **Étapes supplémentaires dans la configuration** du système, plus compliqué ;
* S'il s'agit d'un double démarrage, **Windows ne prend pas en charge LVM**  
  Il n’est pas possible d’accéder aux partitions LVM à partir de Windows ;
* Si un des **volumes physiques devient hors-service** (HS), alors c'est **l'ensemble des volumes logiques** qui utilisent ce volume physique qui sont perdus.   
  Pour éviter ce désastre, il faudra, par exemple, utiliser **LVM sur des disques raid**.

## Éléments de base de LVM

L'une des **décisions difficiles à prendre** par un administrateur système qui **installe Linux consiste à partitionner les unités** **de disque**. La **nécessité d'estimer la quantité probable d'espace nécessaire** pour les fichiers système et les fichiers utilisateur **rend l'installation plus complexe** que nécessaire.

Une fois que l’administrateur système a évalué la quantité d’espace nécessaire pour les répertoires / (répertoire principal, /home, /var, /tmp, … (ou qu’il a laissé le programme d’installation le faire), il est assez courant que l’une de ces **partitions se remplisse même s’il y a beaucoup d’espace disque sur les autres partitions**.

Les options offertes alors sont assez limitées …

* **Reformater le disque**, **modifier le schéma de partitionnement** et **réinstaller** ;
* Faire l’acquisition d’un nouveau disque et **trouver un nouveau schéma de partitionnement** nécessitant un minimum de mouvement de données ;
* **Configurer une batterie de liens symboliques** sur / pointant sur /home et installer le nouveau logiciel sur /home

Une autre solution consisterait à allouer un **minimum d'espace pour chaque volume logique** (LV) et à laisser une partie du disque non allouée. Ensuite, lorsque les partitions commencent à se remplir, elles **peuvent être redimensionnées si cela est nécessaire**.

Les avantages de la gestion des volumes logiques (LV) sont plus évidents sur les grands systèmes dotés de nombreux lecteurs de disque.

**LVM permet de concaténer des disques** (ou partitions) **au sein d’un groupe de volumes**. On peut alors effectuer un découpage de cet ensemble de volumes, que l’on peut voir comme un pseudo-disque dur, afin de former différents volumes logiques

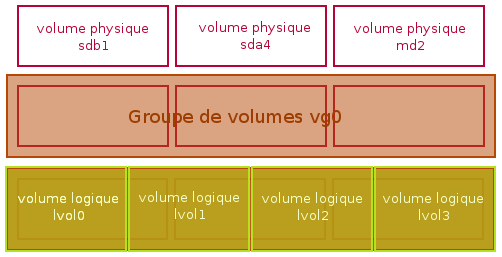
Avec LVM, **le ou les disque(s) entier(s) est/sont alloué(s) à un seul groupe de volumes (VG)** et les **volumes logiques (LV)** **créés pour contenir les systèmes de fichiers** / (répertoire principal, /home, /var, /tmp, ….   
Si, par exemple, le volume logique (LV) /home était rempli ultérieurement mais qu'il restait de l'espace disponible sur /var, il sera alors possible de réduire /var de quelques Mo et de réaffecter cet espace à /home.

La **gestion d'une grande batterie de disques est une tâche fastidieuse**, particulièrement complexe si le système contient de **nombreux disques de tailles différentes**.   
**Équilibrer les exigences de stockage** (souvent conflictuelles) de différents utilisateurs peut être une **tâche complexe**.

Les **groupes d'utilisateurs peuvent être alloués à des groupes de volumes** (VG) et à **des volumes logiques** (LV) et ceux-ci peuvent être développés selon les besoins. Il est possible que l’administrateur système ne « retienne » le stockage sur disque jusqu’à ce qu’il soit nécessaire.   
Il peut ensuite être ajouté au groupe de volumes (utilisateur) ayant le besoin le plus pressant.

Lorsque de **nouveaux lecteurs sont ajoutés au système**, il n'est plus nécessaire de déplacer les fichiers des utilisateurs pour utiliser au mieux le nouveau stockage. Il suffit simplement d’ajouter simplement le nouveau disque dans un ou plusieurs groupes de volumes existants et d’**étendre** **les volumes logiques en fonction des besoins**.

Il est également facile de supprimer les anciens disques en déplaçant les données de ceux-ci sur des disques plus récents. Cette opération peut être effectuée en ligne, sans perturber le service aux utilisateurs.



Les éléments de base LVM sont donc …

* **Volume physique** (*Physical Volume*)  
  Nœud de périphérique de bloc Linux, utilisable pour le stockage par LVM  
  Il héberge un en-tête LVM  
  il peut s’agir …
  + d’un disque dur ;
  + d’une partition MBR ou GPT ;
  + d’un fichier de bouclage  
    ou
  + d’un périphérique de mappeur de périphérique (par exemple, dm-crypt) ;
* **Groupe de volume** (*Volume Group*)  
  Groupe de volume physique (PV) servant de conteneur pour les volumes logiques.   
  Les Étendue physique (PE) sont attribués à partir d'un groupe de volume (VG) pour un Volume logique (LV) ;
* **Volume logique** (*Logical Volume*)  
  Partition virtuelle ou logique qui réside dans un groupe de volume (VG) et est composée d’étendues physiques (PE)  
  Les LV sont des périphériques en bloc Linux analogues aux partitions physiques  
  Ils peuvent, par exemple, être directement formatés avec un système de fichiers ;
* **Étendue physique** (*Physical Extent*)  
  Plus petite étendue contiguë (4 Mo par défaut) du volume physique (PV) pouvant être affectée à un LV  
  Les étendues physiques (PE) comme des parties de volume physique (PV) pouvant être attribuées à n’importe quel volume logique (LV).

**Note** …  
Le **device mapper est un cadritiel (*framework*) fourni par le noyau Linux** afin de faire correspondre les périphériques blocs de niveau physique à des périphériques blocs de niveau virtuel plus élevé.   
Cela constitue la base de LVM2

**Attention** …  
Il ne faut jamais perdre de vue que **seule la partition /boot ne peut faire partie d’un système LVM**, à cause du matériel de démarrage du système qu’il contient.

**Pré-requis**

Un **disque ou une partition libre minimum**. Cet exemple utilise trois disques.  
Ne pas avoir peur de la ligne de commande.

**Installation**

Pour installer le gestionnaire de volume logique …  
**>> sudo apt install lvm2**

**Groupe de volumes (VG – *Volume Group*)**

Un groupe de volumes est, comme son nom l'indique, un **ensemble de volumes physiques**.

On a donc un ou plusieurs volumes physiques dans un groupe de volumes, et pour utiliser LVM, **il faut obligatoirement au moins un groupe de volumes**.

Habituellement, sur les serveurs les plus sollicités, on essaye de regrouper les disques en fonction de leur caractéristiques (capacités, performances, ...).

Une telle configuration est tout de même assez dangereuse en cas de perte d'un disque… De plus, cela n'apporterait aucun gain de performance contrairement à du RAID-0 par exemple. :)

**Volume logique (LV – *Logical Volume*)**

Un volume logique est **l’entité qui est utilisé**.  
Un volume logique est un **espace « quelque part dans un groupe de volume » où l'on peut loger un système de fichiers**.  
C'est, pour LVM, ce qui remplace les partitions.  
On peut donc **utiliser un volume logique pour mettre en place** …

* la **mémoire virtuelle** (*swap disk*) ;
* le **répertoire des usagers** (/home) ;
* le **répertoire principal** (/) ;
* ...

**Astuce**C’est une **bonne pratique d’éviter de situer le répertoire /boot/grub dans un volume logique** car cela peut poser des problèmes pour le démarrage du système.   
En effet, **grub ne sait pas encore** (novembre 2015) **écrire dans un fichier localisé dans un tel volume**.

Or **grub mémorise le résultat du démarrage courant dans le fichier /boot/grub/grubenv** de manière à modifier le démarrage qui suit un démarrage défaillant.

**Création de l’infrastructure de base**

Les commandes LVM sont extrêmement simples à utiliser, et elles intègrent toutes une aide en ligne très bien conçue, claire, courte, mais suffisante.   
De plus, leurs noms se « devinent » assez facilement :

* toutes les **commandes agissant sur les volumes physiques** commencent par **pv** ;
* toutes les **commandes agissant sur les groupes de volumes** commencent par **vg** ;
* toutes les **commandes agissant sur les volumes logiques** commencent par **lv** ;

La **première chose** à faire est de **créer un volume physique**, en attribuant une partition à LVM.

La commande suivante **permet de connaître la liste des commandes disponibles** pour les volumes physiques …

**>> man -k ^pv  
pvchange (8) - change attributes of a physical volume  
pvck (8) - check physical volume metadata  
pvcreate (8) - initialize a disk or partition for use by LVM  
pvdisplay (8) - display attributes of a physical volume  
pvmove (8) - move physical extents  
pvremove (8) - remove a physical volume  
*…***

Parmi toutes les commandes renvoyées, on remarque pvcreate.   
Cette **astuce fonctionne avec toutes les commandes LVM** et permet de les retrouver facilement.

**Groupe de volumes**

Il existe de **nombreuses options lors de la création d'un groupe de volumes**…

Toutefois, le seul paramètre indispensable sera de lui donner un nom.   
Les autres valeurs seront établies par défaut.

Pour connaitre la syntaxe de la commande *vgcreate* (comme pour toutes les autres commandes LVM), il suffit de saisir simplement son nom …  
**>> vgcreate –help**La syntaxe est donc …  
**vgcreate ˂*Nom du Groupe de volume*˃ ˂*Volume physique*˃   
vgcreate ˂*VolumeGroupName*˃ ˂*PhysicalVolume*˃   
[optionnellement d'autres ˂PhysicalVolume˃]**

**>> vgcreate monvg /dev/sdb /dev/sdc  
Physical volume "/dev/sdb" successfully created.  
Physical volume "/dev/sdc" successfully created.  
Volume group "monvg" successfully created**

Le résultat est la création d’un groupe de volumes, contenant un disque physique.

On peut obtenir d'autres informations sur ce groupe de volumes en entrant la commande *vgdisplay* …  
**>>** **vgdisplay  
 --- Volume group ---  
VG Name monvg  
System ID**

**Format lvm2  
Metadata Areas 2  
Metadata Sequence No 1  
VG Access read/write  
VG Status resizable  
MAX LV 0  
Cur LV 0  
Open LV 0  
Max PV 0  
Cur PV 2  
Act PV 2  
VG Size 3,99 Go  
PE Size 4,00 Mo  
Total PE 1022**

**Alloc PE / Size 0 / 0  
Free PE / Size 1022 / 3,99 Go  
VG UUID STKHlD-vjRW-RiRm-dQwX-NXcA-aY0N-mYR2OE**

**Volume logique**

Il faut maintenant créer les espaces que l'on pourra ensuite formater en ext4.

Voici la syntaxe de la commande de base de la commande …  
**>>** **lvcreate**

Les deux options vraiment importantes sont …

* **-n** (pour le nom du volume logique)   
  et
* **-L** (pour la taille du volume logique).

Le **paramètre principal est OriginalLogicalVolume**.

Il s'agit peut-être d'une erreur dans le manuel (man).   
En fait, ce qu'il faut indiquer, c'est bien le groupe de volumes dans lequel nous allons créer le volume logique. Pour l'exemple présent, je fais ici deux volumes, 10 Gio et 50 Gio :

**Lvcreate -n ˂*Nom du Volume logique*˃ -L ˂*Taille du Volume logique*˃   
lvcreate ˂*LogicalVolumeName*˃ ˂*LogicalVolumeSize*˃   
>>** **lvcreate -n vol1 -L 2g monvg  
*Logical volume "vol1" created.***

**>> lvcreate -n vol2 -L 1g monvg  
*Logical volume "vol2" created.***

**Astuce** …

Il est aussi possible de préciser la taille d’un volume logique (LV) en pourcentage d’espace disque libre.  
Pour ce faire, il faudra utiliser, en remplacement de l’option -L), l’option -l suivi du pourcentage.

Voici un exemple … –***l 100%FREE***

On peut maintenant visualiser la création du volume logique …

**>> lvdisplay  
*--- Logical volume ---  
LV Path /dev/monvg/vol1  
LV Name vol1  
VG Name monvg  
LV UUID mVn5dn-1H9E-RCxL-W9Ol-xeYv-FMtw-kthPOH  
LV Write Access read/write  
LV Creation host, time debian9, 2019-04-16 12:08:10 -0400  
LV Status available  
# open 0  
LV Size 2,00 Go  
Current LE 512  
Segments 2  
Allocation inherit  
Read ahead sectors auto  
- currently set to 256  
Block device 254:0  
--- Logical volume ---  
LV Path /dev/monvg/vol2***

***LV Name vol2  
VG Name monvg  
LV UUID MVjL2o-0d0c-mLkg-XeUC-cw7W-sFJa-Ih4XK6  
LV Write Access read/write  
LV Creation host, time debian9, 2019-04-16 12:08:15 -0400***  
***LV Status available  
# open 0  
LV Size 1,00 Go  
Current LE 256  
Segments 2  
Allocation inherit  
Read ahead sectors auto  
- currently set to 256  
Block device 254:1***

**Système de fichiers**

Linux identifie ses partitions avec des étiquettes comme */dev/sda3*.  
Avec **LVM, on utilise aussi des périphériques dans */dev***, mais le **chemin est représenté sous la forme** … **/dev/˂nom\_du\_vg˃/˂nom\_du\_lv˃**.

Les volumes logiques créés précédemment ont comme nom vol1 et vol2, les noms de ces périphériques de ce volume logique sont */dev/monvg/vol1* et */dev/monvg/vol2*.

À partir de maintenant, */dev/monvg/vol****n*** peut être utilisé dans toutes les situations et avec toutes les commandes qui attendent quelque chose de la forme */dev/…* .  
Par exemple …  
**>> mkfs -t ext4 /dev/monvg/vol1  
>> mkfs.ext4 /dev/monvg/vol2  
>> mkdir /travaux  
[>>** **mount /dev/monvg/vol1 /travaux**

**>>** **mkdir /examens  
>>** **mount /dev/monvg/vol2 /examens  
>>**  **df -h  
Sys. de fichiers Taille Utilisé Dispo Uti% Monté sur**

**…**

**/dev/sda1 18G 3,9G 13G 24% /**

**…**

**/dev/sr0 292M 292M 0 100% /media/cdrom0  
/dev/mapper/monvg-vol1 2,0G 6,0M 1,8G 1% /travaux  
/dev/mapper/monvg-vol2 976M 2,6M 907M 1% /examens**

On peut maintenant noter que **/dev/monvg/vol1 est monté sur /travaux** et **/dev/monvg/Vo21** **est monté sur /examen***s*.

**Astuce**Techniquement, pour la commodité, il est **préférable de ne choisir pas des noms longs** (incluant des espaces) pour les **groupes de volumes** (VG) et les **volumes logiques** (LV).  
Ceci **facilite la saisie des différentes commandes LVM**.

**Périphérique *mapper* et LVM**Pourquoi est-il écrit */dev/mapper/monvg-vol1* et non */dev/monvg/vol1* ?

**LVM utilise le périphérique mapper**, ce qui le **rend plus flexible** (comme chiffrer les volumes logiques, ...).   
Ces **deux notations /dev/monvg/vol1 et /dev/mapper/monvg-vol1 sont synonymes**.   
Dans la pratique, il est **conseillé quand même d'utiliser plutôt la forme /dev/monvg/vol1**, certaines commandes ne passeront pas autrement.

**Suppression d’un objet**

Si l’ordre de création d’un volume logique (LV) est clairement défini, il **convient également de respecter l’ordre de suppression**, qui est, en toute logique, **l’ordre inverse** : c’est-à-dire **démontage du système de fichiers** et **retrait** du **volume logique**.

Toutefois, il faut bien comprendre qu’**une** **fois que le volume logique est supprimé**, les **données qui s’y trouvaient ne peuvent pas être récupérées**.

La logique de la suppression d’un objet (VG – LV) suit la même logique …

**>> umount /travaux** **# si le volume vol1 est monté en /travaux  
>> lvremove /dev/monvg/vol1**

**Attention**  
Une fois un **volume logique effacé**, il sera **impossible de récupérer les données qu'il contenait**.

**Redimensionnement des objets**

**Volume physique**

Si un groupe de volume (VG) n'a plus suffisamment d'espace libre, il faut donc lui **rajouter un volume physique** (PV) afin **de d’augmenter l'espace total**.

Il faut, dans un premier temps, **initialiser le nouveau volume physique** (PV) en vue de son utilisation dans LVM …

**>> pvcreate /dev/sdd** **# en supposant un disque supplémentaire disponible  
 # il est aussi possible de le faire avec une partition (/dev/sdd1)**

Par la suite, on **rajoute le volume sdd au groupe de volume monvg** …

**>> vgextend monvg /dev/sdd**

**Volume logique**

La **technique pour augmenter ou de diminuer la taille d'un volume logique est semblable**.   
Il faut toutefois faire attention ; la taille d'un volume logique (LV) n'a pas de lien direct avec la taille de ce qu'il contient (*swap* ou système de fichier).   
Le **volume logique (LV) est une boîte**, le **système de fichier est le contenu de la boîte**.   
**Augmenter la taille de la boîte sans augmenter la taille du contenu** **ne pose pas de problème**, mais l'inverse…

Bien qu'il soit évidemment moins risqué d'agrandir ou de diminuer la taille d'un système de fichiers après l'avoir démonté, la plupart des formats (*ext3*, *reisersfs*, *ext4*…) supportent désormais cette modification à chaud   
(avec des données qui restent donc accessibles en lecture/écriture durant toute l'opération).

**Réduction d’un volume**Pour **agrandir un volume il est nécessaire de démonter le système de fichier** …  
**>> umount /examens**

Pour ajouter 500Mo au volume logique (LV) et agrandir le système de fichier …  
**>> lvresize --resizefs --size +500M /dev/monvg/vol2**

Le **paramètre -resizefs ne fonctionne pas avec tous les systèmes de fichiers**.

**…**

**--- Logical volume ---**

**LV Path /dev/monvg/vol2**

**LV Name vol2**

**VG Name monvg**

**LV UUID MVjL2o-0d0c-mLkg-XeUC-cw7W-sFJa-Ih4XK6**

**LV Write Access read/write**

**LV Creation host, time debian9, 2019-04-16 14:42:24 -0400**

**LV Status available**

**# open 1**

**LV Size 1,49 Go**

**Current LE 381**

**Segments 1**

**Allocation inherit**

**Read ahead sectors auto**

**- currently set to 256**

**Block device  *254:1***

**Remarque** …  
Tous les systèmes de fichiers ne supportent pas d'être redimensionnés.

Afin de vérifier la réussite des modifications voulues …  
**>> df -h | grep monvg****/dev/mapper/monvg-vol1 2,0G 6,0M 1,8G 1% /travaux**

**/dev/mapper/monvg-vol2 1,5G 3,0M 1,4G 1% /examens**

Les valeurs intéressantes à regarder sont la deuxième et la quatrième, à savoir …  
*/dev/mapper/monvg-vol2*

* ***1,5 Go d'espace total****;*
* ***3,0 Go d'espace libre****.*

Pour **diminuer l’espace disque alloué à un volume logique** (LV) …

L'espace disque étant de 1,5 Go, i; est possible de réduire l'espace de 500 Mo.

Dans un premier temps, il faut démonter le volume …  
**>> umount /examens**

Par la suite, faut y **retirer retirer 500 Mo** …  
**>> lvresize --resizefs --size -500M /dev/mapper/monvg   
fsck de util-linux 2.29.2  
/dev/mapper/monvg-vol2 : 11/98304 fichiers (0.0% non contigus), 15140/390144 blocs  
resize2fs 1.43.4 (31-Jan-2017)  
En train de redimensionner le système de fichiers sur /dev/mapper/monvg-vol2 à 262144 (4k) blocs.  
Le système de fichiers sur /dev/mapper/monvg-vol2 a maintenant une taille de 262144 blocs (4k).  
Size of logical volume monvg/vol2 changed from 1,49 GiB (381 extents) to 1,00 GiB (256 extents).  
Logical volume monvg/vol2 successfully resized.**

Pour **vérifier la réussite de la comamnde lvresize** …  
**>> lvdisplay   
--- Logical volume ---**

**LV Path /dev/monvg/vol2**

**LV Name vol2**

**VG Name monvg**

**LV UUID MVjL2o-0d0c-mLkg-XeUC-cw7W-sFJa-Ih4XK6**

**LV Write Access read/write**

**LV Creation host, time debian9, 2019-04-16 14:42:24 -0400**

**LV Status available**

**# open 0**

**LV Size 1,00 Go**

**Current LE 256**

**Segments 1**

**Allocation inherit**

**Read ahead sectors auto**

**- currently set to 256**

**Block device 254:1**

**Opérations de déplacement de données**

**pvmove est la commande permettant de déplacer les segments physiques** alloués d’un disque à un autre.   
Lorsque l’on dispose d’assez d’espace sur d’autres disques du même groupe de volumes, on peut alors exécuter la commande suivante …  
**>> pvmove /dev/sdc**

On peut bien évidemment **déplacer uniquement les extensions d’un volume logique particulier** …  
**>> pvmove -n vol1 /dev/sdc1**

De même, si l’on souhaite **déplacer les données d’un volume physique à un autre**, on peut exécuter en mode arrière-plan la commande suivante …  
**>> pvmove -b /dev/sdc1 /dev/sdd1**

**Attention** …  
Ce **genre de commande peut prendre du temps selon la volumétrie déplacée**.   
Si l’on souhaite **visualiser la progression** (sous forme de pourcentage), d’une telle commande, il faut **utiliser l’option -i** en mentionnant **l’espace d’intervalle entre chaque prise de mesure** …  
**>>** **pvmove -i5 /dev/sdc1 /dev/sdd1**

**Vérification du système de fichier**

Pour démonter la partition …  
**>> umount /travaux   
>> fsck -f -y /dev/monvg/vol1  
fsck de util-linux 2.29.2  
e2fsck 1.43.4 (31-Jan-2017)  
Passe 1 : vérification des i-noeuds, des blocs et des tailles  
Passe 2 : vérification de la structure des répertoires  
Passe 3 : vérification de la connectivité des répertoires  
Passe 4 : vérification des compteurs de référence  
Passe 5 : vérification de l'information du sommaire de groupe  
/dev/mapper/monvg-Vol1 : 11/131072 fichiers (0.0% non contigus), 26156/524288 blocs**

**Interrogations**

Il est possible d’**interroger la liste des volumes physiques** de la manière suivante …  
**>> pvs   
PV VG Fmt Attr PSize PFree**

**/dev/sdb monvg lvm2 a-- 2,00g 0**

**/dev/sdc monvg lvm2 a-- 2,00g 416,00m**

**/dev/sdd monvg lvm2 a-- 2,00g 2,00g**

Cette commande permet de **mettre en relation le nom du volume physique avec le groupe de volumes auquel il appartient**. Elle affiche également sa **taille occupée et ce qu’il reste d’espace libre**.

De la même façon, on peut également interroger la **liste des groupes de volumes existant sur le système** …

**>> vvs   
VG #PV #LV #SN Attr VSize VFree  
monvg 3 3 1 wz--n- 5,99g 2,40g**

On peut noter au passage que comme je l’avais annoncé, le **volume /boot ne fait pas partie du système LVM**.

Toutefois, **l’ensemble des autres partitions sont bien intégrées au groupe de volume vg00** et se répartissent l’espace du volume physique.

Enfin, pour **visualiser la liste des volumes logiques** …  
**>> vvs**

**LV VG Attr LSize Pool Origin Data% Meta% …**

**Vol1 monvg -wi-a----- 2,00g**

**Vol2 monvg owi-aos--- 1,00g**

**lvtest01 monvg swi-a-s--- 600,00m Vol2 0,00**

**Instantanés (*snapshot*)**

Un **instantané ne copie pas l'intégralité du volume logique (LV) original**.   
Au contraire, il ne va **stocker que les différences**.   
C'est pourquoi il est instantané et **commence avec une occupation taille nulle**.   
La commande **lvdisplay permet de voir l'évolution de la taille**.

Avec LVM2, les **instantanés sont par défaut en lecture/écriture**.   
Le fonctionnement est similaire aux instantanés en lecture seule avec la possibilité supplémentaire d'écrire sur l'instantané : le bloc est alors marqué comme utilisé dans la table d'exceptions et ne sera plus récupéré du volume source.   
Par exemple, on peut faire l'instantané d'un volume, le monter et tester un programme expérimental qui modifie des fichiers importants.   
Si le résultat n'est pas satisfaisant, il est alors possible de le démonter, le supprimer et de remonter le système de fichiers originel.

**Création d'un instantané LVM**

La commande suivante va créer un instantané du volume logique (LV) à la taille de 500 Mo.  
**>> lvcreate -L 500M -s -n lvtest01 /dev/monvg/vol2**

**Attention** …  
La **taille d'utilisation de l’instantané évolue avec … l'utilisation**.   
Si cet **instantané se retrouve rempli à 100%,** il devient alors inutilisable (état *INACTIVE*).  
Toutefois, il n'y aura **pas d’impact pour le LV d'origine**.

**Redimensionnement d’un instantané**Si la taille de l’instantané est trop petite et elle arrive à son point de saturation (100%)  
Il faut donc redimensionner l’instantané.

Dans un premier temps, il faut vérifier avec la commande *vgdisplay* que le groupe de volume (VG) dispose encore d'assez d'espace libre (Free PE / Size).

Par la suite, il faut entrer la commande suivante …  
**>> lvresize -L +100M /dev/monvg/lvtest01**

Cette commande va ajouter 100 Mo à l’instantané */dev/monvg/lvtest01* (qui était de 500 Mo à l’origine) …  
**>> lvdisplay /dev/monvg/lvtest01   
--- Logical volume ---**

**LV Path /dev/monvg/lvtest01**

**LV Name lvtest01**

**VG Name monvg**

**LV UUID jq8JwW-JGJG-0HMm-1ZgW-KIGC-QMFz-wZZa2B**

**LV Write Access read/write**

**LV Creation host, time debian9, 2019-04-16 15:53:36 -0400**

**LV snapshot status active destination for Vol2  
LV Status available**

**# open 0**

**LV Size 1,00 Go**

**Current LE 256**

**COW-table size 600,00 MiB**

**COW-table LE 150**

**Allocated to snapshot 0,00%**

**Snapshot chunk size 4,00 KiB**

**Segments 1**

**Allocation inherit**

**Read ahead sectors auto**

**- currently set to 256**

**Block device 254:1**

**Fusion d’un instantané**

La fusion d’un instantané vise à **transférer un instantané modifié vers le volume logique (LV) d'origine**.   
En d’autres termes, les **modifications apportées sur l’instantané va se retrouver sur le volume logique (LV) d'origine**.

Pour **fusionner un instantané** …  
**>> lvconvert --merge /dev/monvg/vol2**

**Attention …**  
Ceci est **possible à partir de la version de noyau >=2.6.33**.

**Notion d'étendue physique** *(Physical Extent)*

Une **étendue physique (PE) est un tout petit morceau d'un groupe de volumes**.   
En fait, au moment de la création d'un groupe de volumes, le ou les disques sont **découpés en morceaux de quelques Mo** (4 Mo par défaut).   
Lors de la création d’un volume logique, **LVM va utiliser autant de PE que nécessaires**.   
La taille **d'un volume logique sera donc toujours un multiple de la taille d'une étendue physique** (PE).

**Petit glossaire**…

|  |  |  |  |
| --- | --- | --- | --- |
| Abréviation | Français | Anglais | Description |
| VG | Groupe de volumes | *Volume Group* |  |
| LV | Volume logique | *Logical Volume* | Partition dans un groupe de volumes |
| PV | Volume physique | *Physical Volume* |  |
| PE | Étendue physique | Physical Extent | Petit morceau d'un groupe de volumes |

**Interface graphique pour LVM**

Il existe une **interface graphique pour LVM**, qui permet de manipuler les volumes logiques : **system-config-lvm**.

**Attention** …  
***system-config-lvm applique directement les changements***.  
Il est essentiel de **bien vérifier les commandes avec system-config-lvm**.  
Il est **risqué de perdre irrémédiablement les données**.

**Commandes LVM**

**Monter une partition**

Pour obtenir la liste des groupes logiques …  
**>> lvm vgscan  
Reading volume groups from cache.  
 Found volume group "monvg" using metadata type lvm2**

Pour obtenir la liste des partitions …  
**>> lvm lvs   
LV VG Attr LSize Pool Origin Data% Meta% …  
vol1 monvg -wi-ao---- 2,00g  
vol2 monvg owi-aos--- 1,00g  
lvtest01 monvg swi-a-s--- 600,00m Vol2 0,00**

Pour rendre la partition disponible …  
**>> lvm lvchange -ay /dev/monvg/vol1**

Pour monter la partition …  
**>> mkdir /exercices   
>> mount /dev/monvg/vol1 ˂/media/user/vol1˃  
>> mount /dev/monvg/vol1 /exercices**

**Miscellaneous**

Les étapes de mise en place de LVM sont les suivantes …

* **Mise en œuvre d’une structure** de type LVM en fournissant tous les éléments d’administration et toutes les opérations réalisables sur une telle architecture ;
* **Activation d’une structure LVM** dès l’installation d’un système ;
* **Mise en place de sauvegardes de structures** avec toutes les possibilités que cela peut offrir, y compris en termes de restauration ;
* **Création et gestion des clichés instantanés** (*snapshots*), d’une ou plusieurs arborescences du système;
* **Gestion, sécurisation et optimisation des espaces de stockage** présentés au sein d’un système d’exploitation Linux.

**Principales commandes**

Au niveau des **groupes de volumes**, on dispose ainsi des commandes suivantes …

* **vgcreate**  
  Création des groupes de volumes ;
* **vgdisplay**  
  Liste du contenu des groupes de volumes ;
* **vgck**Vérification de la cohérence des métadonnées d’un groupe de volumes ;
* **vgremove**Suppression d’un groupe de volumes ;
* **vgscan**  
  Détection des volumes physiques et groupes de volumes.

**Note** …  
La **commande vgscan va également permettre de générer le fichier etc/lvm/.cache** (ainsi que le **fichier lvmtab**), afin de **maintenir une liste cohérente des périphériques LVM courants**.

Le système **LVM exécute automatiquement la commande vgscan lors des phases de démarrage** du système d’exploitation.

Au **niveau des volumes physiques**, il est également **possible d’intervenir et de gérer les différents volumes physiques** …

* **pvcreate**  
  Création d’un volume physique ;
* **pvresize**  
  Modification de la taille d’un volume physique ;
* **pvdisplay**  
  Détails sur le découpage d’un volume physique ;
* **pvremove**  
  Suppression d’un volume physique ;
* **pvmove**  
  Déplacement du contenu d’un volume physique vers un autre ;
* **pvchange**  
  Modification des métadonnées d’un volume physique ;
* **pvscan**  
  Détection des volumes physiques parmi les périphériques du système.

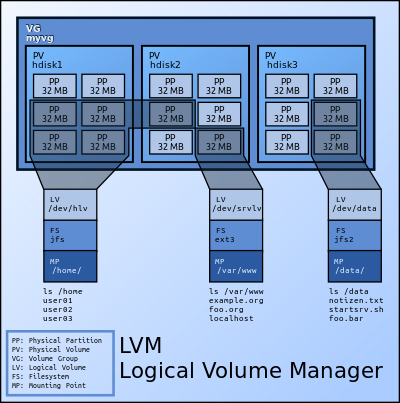
Au niveau des **volumes logiques**, on peut aussi **effectuer des créations et des modifications sur ces éléments** …

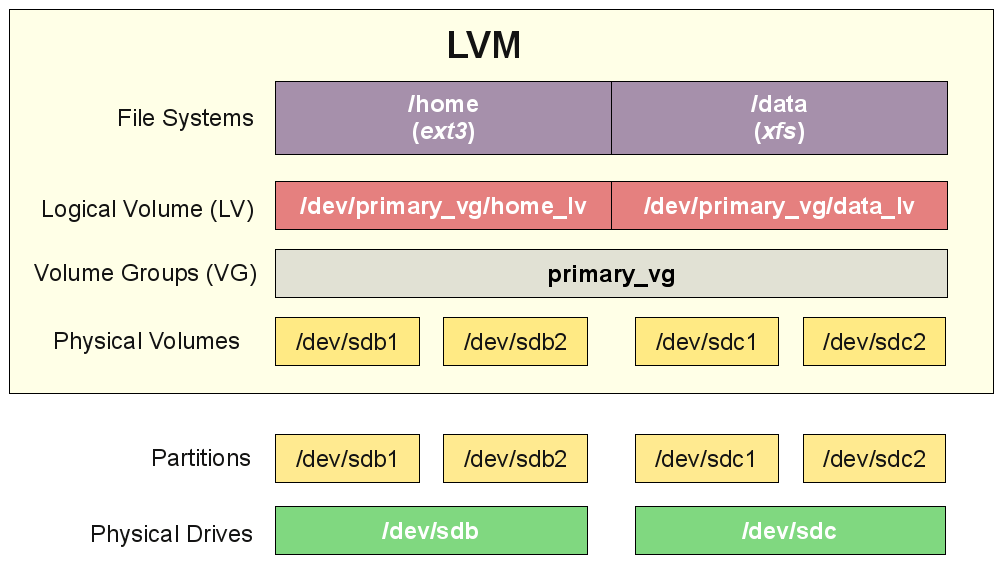
* **lvcreate**  
  Création d’un volume logique ;
* **lvdisplay**  
  Liste des détails d’un volume logique ;
* **lvremove**  
  Suppriession d’un volume logique ;
* **lvextend**  
  Extension de la taille d’un volume logique ;
* **lvmdiskscan**  
  Détection tous les périphériques visibles au système LVM.

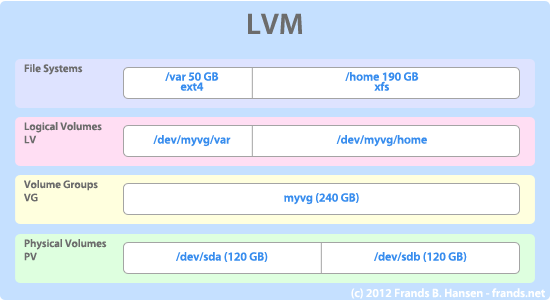
À ces commandes s’ajoutent des **commandes de visualisation sous forme de liste**, **mettant en relation les volumes** **physiques**, les **volumes logiques** et leur(**s) groupe(s) de volumes associés** …

* **pvs**  
  Liste des volumes physiques ;
* **lvs**  
  Liste des volumes logiques ;
* **vgs**  
  Liste des groupes de volumes.

**Schéma des principaux éléments LVM**







Filesystem Hierarchy Standard

Bien que chaque distribution Linux ait sa propre façon de faire les choses, les principaux acteurs du monde Linux reconnaissent la **nécessité d’une normalisation dans la disposition des répertoires**.

Par exemple, les programmes doivent pouvoir **localiser les fichiers de configuration système clés de manière cohérente aux mêmes endroits sur toutes les distributions**.

Si ce n’était pas le cas, les programmes qui reposent sur de telles fonctionnalités deviendraient plus complexes et ne fonctionneraient peut-être pas avec toutes les distributions.

Afin de répondre à ce besoin, FHS (Standard de hiérarchie des systèmes de fichiers) a été créé.

Le FHS fait une distinction importante entre les **fichiers partageables** et les **fichiers non partageables**.

* Les **fichiers partageables peuvent être partagés entre ordinateurs**, tels que les fichiers de données utilisateur et les fichiers binaires de programmes. (Bien sûr, nul n’est besoin de partager de tels fichiers, mais il est toujours possible de le faire.)   
  Si des fichiers sont partagés, ils le sont **généralement à l’aide d’un serveur NFS**.
* Les **fichiers non partageables contiennent des informations spécifiques au système**, telles que les fichiers de configuration.
* Comme exemple, il est **peu probable qu’il soit souhaitable de partager** le fichier de configuration d’un serveur entre hôtes.

Le FHS établit également une **seconde distinction importante** entre les **fichiers/répertoires statiques** et les **fichiers/répertoires variables.**

* Les **fichiers/répertoires statiques ne changent généralement pas**, sauf par intervention directe d’un administrateur du système.   
  La **plupart des exécutables de programme** sont de bons exemples de fichiers statiques.
* **Les utilisateurs**, **les scripts automatisés**, **les serveurs** ou similaires **peuvent modifier les fichiers/répertoires variables**.   
  Par exemple, les répertoires de base et les fichiers de courriels des utilisateurs sont composés de fichiers variables.

|  |  |  |
| --- | --- | --- |
|  | **Statique** | **Dynamique** |
| **Statique** | **/etc**  **/bin**  **/sbin**  **/lib** | **/dev**  **/var**  **/log** |
| **Variable** | **/usr**  **/opt** | **/home**  **/var/mail** |

Il existe aussi une seconde classification …

|  |  |  |
| --- | --- | --- |
|  | **Partageable** | **Non partageable** |
| **Local** | **/usr**  **/opt** | **/etc**  **/boot** |
| **À distance** | **/home**  **/var/mail** | **/var/run**  **/var/lock** |

Le FHS cherche à isoler chaque répertoire dans une cellule d’une matrice 2 × 2   
(partageable / non partageable × statique / variable).

Certains répertoires contiennent des sous-répertoires dans plusieurs cellules, mais dans ce cas, le FHS tente de spécifier le statut de sous-répertoires particuliers.

Par exemple, */var* est variable et il contient des sous-répertoires partageables et des sous-répertoires non partageables,

Le FHS est disponible en versions numérotées.   
La **version 3,0, la dernière version en date**.

**Répertoires importants et leur contenu**

La FHS définit les noms et les objectifs de nombreux répertoires et sous-répertoires sur un système Linux.

Le tableau qui suit récapitule les répertoires les plus importants.

**La plupart d'entre eux sont des répertoires système, les principales exceptions étant /home, /tmp, /mnt et /media**.

En tant qu'**utilisateur** **ordinaire**, la plupart des **fichiers du répertoire personnel**, se retrouve normalement dans un sous-répertoire de */home*. Il est également possible d’accéder à des supports amovibles montés sur */media* (ou parfois */run/media*) et éventuellement aux ressources réseau pouvant être montées ailleurs.

**Répertoires importants selon le FHS**

|  |  |
| --- | --- |
| Répertoire | Objet |
| / | **Répertoire racine**  Tous les fichiers apparaissent dans ce répertoire ou dans des sous-répertoires |
| /etc | Contient les **fichiers de configuration du système** |
| /boot | Contient les **fichiers de démarrage importants**, tels que le **noyau Linux**, le **disque RAM initial** et normalement des **fichiers de configuration du chargeur** |
| /bin | Contient les **fichiers de programme essentiels au fonctionnement normal** et aux fichiers ordinaires que les utilisateurs peuvent utiliser |
| /sbin | Contient les **fichiers de programme critiques pour un fonctionnement normal**  Les utilitaires de ce répertoire sont surtout utiles à l’administrateur système |
| /lib  /lib64 | Contient les **bibliothèques** - le **code utilisé par plusieurs autres programmes** - qui sont critiques pour le fonctionnement de base du système |
| /usr | Contient les **programmes et les données utilisés dans le fonctionnement normal** du système  Ils ne sont toutefois pas critique pour le démarrage du système  Ce répertoire est divisé en sous-répertoires reflétant des parties de l’organisation racine - /usr/bin,  */usr/sbin*, */usr/lib*, … |
| /home | Contient les **répertoires personnels des utilisateurs**  La séparation de ce répertoire dans son propre système de fichiers de bas niveau isole efficacement la plupart des données utilisateur du système d'exploitation, ce qui peut être utile en cas de réinstallation du système d'exploitation sans toutefois perdre les données de l'utilisateur |
| /root | **Répertoire de base de l'utilisateur root** |
| /var | **Contient divers fichiers transitoires**, tels que les **fichiers journaux** et les **fichiers de la queue** **d'impression**  Le sous-répertoire de */var*, */var/tmp*, mérite une attention particulière Comme */tmp*, */var/tmp* contient des fichiers temporaires -- Ces fichiers ne doivent pas être supprimé lorsque l'ordinateur redémarre |
| /tmp | Contient les **fichiers temporaires**, y compris les fichiers temporaires créés par les programmes utilisateur.  De tels fichiers **peuvent théoriquement être supprimés lorsque l'ordinateur redémarre**, bien que dans la pratique de nombreuses distributions ne le fassent pas. |
| /mnt | Le **point de montage traditionnel** pour les supports amovibles  Il est parfois divisé en sous-répertoires pour chaque système de fichiers monté |
| /media | Le **nouveau point de montage pour les supports amovibles**; typiquement séparés dans des sous-répertoires pour chaque système de fichiers monté |
| /dev | Contient les **fichiers de périphérique**, qui fournissent un accès de bas niveau au matériel |
| /run | Contient des **informations sur le système en cours d'exécution** |

Il est aussi possible d’utiliser **/tmp** et certains sous-répertoires de **/var**.   
Toutefois, **la plupart des utilisateurs n’ont pas besoin de connaître explicitement ces emplacements**.

En tant qu'administrateur système, il faudra manipuler des fichiers situés dans l'un de ces répertoires.   
Cependant, pour un **administrateur système, /etc est particulièrement important**, car c’est là que réside la plupart des **fichiers de configuration du système**.

Parmi ces répertoires, plusieurs répertoires individuels ou collections d’entre eux méritent une attention particulière …

* **Répertoire de configuration**Le répertoire **/etc contient la plupart des fichiers de configuration système**.   
  Ce répertoire inclus plusieurs fichiers de ce type, tels que */etc/fstab* (qui définit où les partitions sont montées) et */etc/passwd* (qui est le fichier de définition de compte principal).   
  Beaucoup d'autres existent. Il existe des sous-répertoires dans */etc* pour héberger plusieurs fichiers de configuration pour des sous-systèmes complexes et des serveurs, tels que */etc/X11* (pour le système X Window) et */etc/samba* (pour le serveur de fichiers Samba).
* **Répertoires exécutables**  
  Les **fichiers de programme résident principalement dans /sbin, /bin, /usr/bin et /usr/sbin**.  
  (Des répertoires supplémentaires peuvent héberger des fichiers de programme sur certains systèmes. En particulier, */usr/local/bin et /usr/local/sbin* contiennent des programmes compilés localement.)
* **Répertoires de bibliothèque**Les **bibliothèques sont des ensembles de fonctions de programmation pouvant être utiles à de nombreux programmes**. Ils sont stockés dans des fichiers distincts pour économiser l’espace disque et la RAM lors de l’exécution des programmes.   
  Ils permettent de **mettre à jour facilement les fichiers de bibliothèque** sans réinstaller tous les programmes qui en dépendent.   
  Sous Linux, la plupart des **bibliothèques résident dans /lib, /lib64, /usr/lib et /usr/lib64**, bien que certaines puissent résider ailleurs (comme */usr/local/lib*) sur certains systèmes.

**Remarque** …  
Sous Linux, la plupart des **fichiers clés d’un programme sont susceptibles de résider dans des emplacements standard partagés** avec d’autres programmes et peuvent être dispersés.   
Par exemple, l’exécutable du programme peut se trouver dans */usr/bin*, les bibliothèques associées dans */usr/lib*, les fichiers de configuration dans */etc* ou dans les répertoires de départ des utilisateurs, ... Cela fonctionne bien sous Linux car les systèmes de paquetage de Linux suivent de près les nombreux fichiers d’un paquet, ce qui permet de supprimer ou de mettre à jour facilement un paquet.

**Un peu plus sur les répertoires système**

Un système Linux est **constitué de centaines de milliers de fichiers**. Afin de garder une trace,

il existe certaines **conventions pour la structure de répertoires et les fichiers** comprenant un système Linux, le **FHS (*Filesystem Hierarchy Standard*)**.

**/ – Répertoire racine** (*root*)  
L'arborescence du système de fichiers commence au répertoire racine, représenté par la barre oblique /.  
Il est important de ne pas confondre avec le répertoire racine */root*, le répertoire de base de l'utilisateur *root*).

**/boot – Noyau du système d'exploitation**Le répertoire */boot* **contient le système d’exploitation actuel** : **vmlinuz** est le **noyau Linux**.   
Le répertoire */boot* contient également d'autres **fichiers nécessaires au chargeur de démarrage** (LILO ou GRUB).

**/bin – Utilitaires généraux**Dans /bin, on retrouve les **programmes exécutables les plus importants** (principalement des programmes système) **nécessaires au démarrage du système**. Cela inclut, par exemple, *mount* et *mkdir*.   
Beaucoup de ces programmes sont si essentiels qu'ils sont nécessaires non seulement lors du démarrage du système, mais **également lorsque celui-ci est en cours d'exécution**, comme *ls* et *grep*.   
*/bin* contient également des **programmes nécessaires afin de relancer un système endommagé**. Cela est possible seulement le système de fichiers contenant le répertoire racine est disponible.   
Des **programmes supplémentaires qui ne sont pas nécessaires au démarrage ou pour la réparation** du système sont disponibles dans **/usr/bin**.

**/sbin – Programmes système spéciaux**

Comme */bin*, */sbin* contient les **programmes nécessaires au démarrage ou à la réparation du système**.   
Toutefois, pour la plupart, ce sont des **outils de configuration système** qui ne peuvent vraiment être utilisés que par *root*. Les utilisateurs dits normaux peuvent utiliser certains de ces programmes pour interroger le système, sans toutefois être capable de modifier quoi que ce soit.   
Tout comme avec */bin*, il existe un répertoire appelé */usr/sbin* contenant plus de programmes système.

**/lib – Bibliothèques système**C'est dans ce répertoire que l’on retrouve les **bibliothèques partagées qui sont utilisées par les programmes résidant dans /bin et /sbin**, sous forme de **fichiers et de liens** (symboliques).   
Les bibliothèques partagées sont des **morceaux de code utilisés par divers programmes**. Ces bibliothèques permettent d'**économiser** **beaucoup de ressources**, car de **nombreux processus utilisent les mêmes composants de base. Ces composants de base ne doivent ensuite être chargés qu'une seule fois en mémoire.**En outre, il est plus facile de corriger les bogues dans ces bibliothèques lorsqu'elles ne sont dans le système qu'une seule fois et que tous les programmes récupèrent le code en question à partir d'un fichier central.

On retrouve, sous **/lib/modules**, des **modules de noyau qui sont en réalité le code du noyau qui est utilisé au besoin** – pilotes de périphérique, systèmes de fichiers ou protocoles réseau, …  
Ces modules peuvent être chargés par le noyau quand ils sont nécessaires et, dans de nombreux cas, également supprimés après utilisation.

**/dev – Fichiers de périphérique**Ce répertoire, et ses sous-répertoires, contiennent une **multitude d'entrées pour les fichiers de périphérique**.   
Les fichiers de périphérique **constituent l'interface entre le shell et les pilotes de périphérique du noyau**.   
Ils ne se présentent pas comme les autres fichiers car ils font référence à un pilote du noyau à partir de numéros de périphériques.

Linux fait une distinction entre les périphériques de caractères et les périphériques en mode bloc.

* Un **périphérique de bloc de caractères** de périphérique est, par exemple, **un terminal, une souris** ou **un** **modem**. Ils sont des périphériques qui fournissent ou traitent des caractères uniques.
* Un **périphérique en mode bloc** traite les données en blocs. Ils interfacent avec les **disques durs** où les octets ne peuvent pas être lus séparément mais uniquement par groupes de 512 (ou autres).

Selon leur type, les fichiers de périphérique sont étiquetés à la sortie de *ls -l* avec un c ou un b …  
**crw-rw-rw- 1 root root 10,4 octobre 25 0:00 mouse  
brw-rw ---- 1 root disque 8,1 octobre 25 0:00 sda1  
brw-rw ---- 1 root disque 8,2 octobre 25 0:00 sda2  
crw-rw-rw- 1 root root 1,3 octobre 25 0:00 null**

Au **lieu de la longueur du fichier**, la **liste contient deux nombres**.

* Le premier est le **numéro de périphérique majeur spécifiant le type de périphérique** et **déterminant le pilote du noyau responsable de ce périphérique**.   
  Par exemple, tous les disques durs SCSI ont le numéro de périphérique majeur 8.
* Le deuxième **numéro est le numéro de périphérique mineur**.   
  Ce numéro est utilisé par le pilote pour faire la **distinction entre différents périphériques similaires** ou **apparentés** ou pour **désigner les différentes partitions d'un disque**.

Il existe **plusieurs pseudos périphériques intéressants** ...

* Le **périphérique nul**, **/dev/null**, est comme un **pseudo périphérique poubelle**. Il agit comme une **sortie** de **programme non requise**, mais devant être **dirigé quelque part**.   
  Avec une commande telle que **programme > /dev/null**, la sortie standard du programme, qui sinon serait affichée sur le terminal, est ignorée.   
  Si **/dev/null est lu**, il **se présente comme un fichier vide** et **renvoie immédiatement la fin du fichier**.   
  */***dev/null doit être accessible à tous les utilisateurs** pour la **lecture et l'écriture**.
* Les **périphériques /dev/random** et **/dev/urandom** **renvoient des octets aléatoires de qualité cryptographique** créés **à partir de bruit dans le système**.   
  Comme exemple, on compte parmi les événements utilisés afin de créer ces octets aléatoires les intervalles entre les événements imprévisibles, tels que les pressions sur les touches.

**Remarque** …

Les données **du fichier /dev/random** conviennent à la **création de clés pour des algorithmes cryptographiques** courants. Le **fichier /dev/zero renvoie un nombre illimité d’octets nuls**.  
Il est **possible de les utiliser**, par exemple, afin de **créer ou de remplacer des fichiers avec la commande dd**.

**/etc – Fichiers de configuration**Le **répertoire */etc* est très important**.   
Ce dernier contient les **fichiers de configuration pour la plupart des programmes**.   
Les fichiers / etc / inittab et /etc/init.d/\*, par exemple, contiennent la **plupart des données spécifiques au système nécessaires au démarrage des services système**.

Voici une description plus détaillée des fichiers les plus importants.   
À l'exception de quelques-uns d'entre eux, **seul root a le droit d'écrire**. Toutefois, **tout le monde peut les lire**.

* **/etc/fstab**Ce fichier **décrit tous les systèmes de fichiers montables et leurs propriétés**   
  (type, méthode d'accès, point de montage) ;
* **/etc/hosts**Ce fichier est l’**un des fichiers de configuration du réseau TCP/IP**.   
  Il établit le lien entre les noms des hôtes du réseau avec leurs adresses IP   
  Dans les petits réseaux et sur les hôtes autonomes, ce fichier peut remplacer un serveur de noms ;
* **/etc/inittab**Le fichier */etc/inittab* est le **fichier de configuration du programme init** et donc du démarrage du système  
  ***/etc/init.d/\**** est le répertoire qui **contient les scripts d'initialisation pour divers services système**  
  Ils sont utilisés pour démarrer ou arrêter les services système lorsque le système est démarré ou arrêté  
  Sur les distributions Red Hat et dérivés, ce répertoire s'appelle */etc/rc.d/init.d* ;
* **/etc/issue**Ce fichier contient le **message d'accueil qui est émis avant qu'un utilisateur ne soit invité à se connecter**  
  Après l'installation d'un nouveau système, il contient souvent le nom du fournisseur ;
* **/etc/motd**Ce fichier contient le **message du jour** (***M****essage* ***O****ff* ***T****he* ***D****ay*) qui **apparaît après la connexion d'un utilisateur**  
  L'administrateur système peut utiliser ce fichier pour informer les utilisateurs d'événements et d'événements importants ;
* **/etc/mtab**Ce fichier propose une **liste de tous les systèmes de fichiers montés**, y compris leurs **points de montage**  
  */etc/mtab* diffère de */etc/fstab* en ce sens qu'il contient tous les systèmes de fichiers actuellement montés, alors que */etc/fstab* ne contient que des paramètres et des options pour les systèmes de fichiers pouvant être montés, généralement au démarrage du système, mais aussi ultérieurement.   
  **Remarque** ...  
  Il est préférable de ne pas mettre ce genre d’informations dans un fichier tel que */etc* où les fichiers doivent être statiques.
* **/etc/passwd**Dans */etc/passwd*, on retrouve la **liste de tous les utilisateurs connus du système**, ainsi que diverses **informations spécifiques à ces utilisateurs**  
  **Remarque** …  
  Malgré le nom du fichier, sur les systèmes modernes, les mots de passe ne se retrouve pas dans ce fichier mais dans un autre appelé */etc/shadow*Contrairement à */etc/passwd*, ce **fichier n'est pas lisible par les utilisateurs normaux**.

**/opt – Accessoires**Ce répertoire est **destiné aux logiciels** tiers, c'est-à-dire aux **paquetages complets** qui sont supposés être **installables sans conflit avec les fichiers de distribution** ou les **fichiers installés localement**.   
De tels paquetages logiciels occupent le sous-répertoire */opt/˂Nom u paquetage˃*.   
Techniquement, le répertoire */opt* **devrait être complètement vide après l’installation** d’une système Linux sur un disque vide.

**/usr – Fichier immuables**Dans le répertoire /usr, il existe divers sous-répertoires contenant des **programmes et des fichiers de données qui ne sont pas essentiels**, ou **indispensables**, au **démarrage ou à la réparation du système**.  
Les répertoires les plus importants incluent …

* **/usr/bin**Programmes système qui **ne sont pas essentiels au démarrage** ou qui sont d’une autre importance ;
* **/usr/sbin**Plus de **programmes système habituellement réservé à l’usager root** ;
* **/usr/lib**Autres **bibliothèques** (non utilisées pour les programmes du répertoire */bin* ou */sbin* ;
* **/usr/local**Répertoire des **fichiers installés par l'administrateur système local**  
  Correspond au répertoire ***/opt*** (la distribution ne peut rien mettre ici) ;
* **/usr/shareDonnées indépendantes de l'architecture** ;
* **/usr/share/doc**Ce répertoire contient **de la documentation relative aux « HOWTOs »** ;
* **/usr/share/info**Ce répertoire **contient de la documentation relative aux pages d'information** (commande *info*) ;
* **/usr/share/man**Ce répertoire **contient les pages du manuel** (dans des sous-répertoires) accessibles à l’aide de la commande *man* ;
* **/usr/src**Ce répertoire **rassemble le code source pour le noyau et d'autres programmes** (si disponible).

**Astuce** …  
Le **nom /usr est souvent considéré à tort comme un acronyme de *Unix System Ressources***.  
En fait, à l'origine, ce répertoire provient du temps où les ordinateurs avaient souvent un pe**tit disque dur rapide et un autre plus gros mais plus lent**.

Afin d’**optimiser** les **performances de Lecture/Écritures**, tous les **programmes et fichiers fréquemment utilisés** sont **sauvegardés sur le petit disque**, tandis que le **gros disque** (monté sous /usr) **servait de référentiel pour les fichiers et les programmes moins fréquemment utilisés** ou encore **trop volumineux**.

**/proc – Une fenêtre sur le noyau**C'est l'un des **répertoires les plus intéressants et les plus importants**.   
**/proc** est vraiment un **pseudo système de fichiers**.   
Ses **sous-répertoires et ses fichiers sont créés par le noyau** afin que les utilitaires et programmes puissent y trouver de l’information pertinente sur le système.

On y retrouve, entre autres, de **nombreuses données sur les processus en cours**, ainsi que d’autres **informations dont dispose le noyau sur le matériel** de l’ordinateur.

Par exemple, certains fichiers proposent une **analyse complète du matériel**.

Les fichiers les plus importants incluent …

* **/proc/cpuinfo**Ce fichier contient des **informations sur le type et la fréquence d’horloge du CPU** ;
* **/proc/devices**Ce fichier contient une **liste complète des périphériques pris en charge par le noyau**, y compris leurs principaux numéros  
  Cette **liste est consultée lors de la création des fichiers de périphérique** ;
* **/proc/dma**Ce fichier consiste en une **liste des canaux DMA utilisés**.   
  Sur les systèmes PCI actuels, ces données ne sont ni intéressantes ou importantes ;
* **/proc/interrupts**Ce fichier renferme une **liste de toutes les interruptions matérielles en cours d'utilisation**.   
  Celui-ci contient le numéro d'interruption, le nombre d'interruptions déclenchées et les pilotes gérant cette interruption particulière  
  **Note** …  
  Une interruption survient dans cette liste uniquement si un pilote du noyau le réclame ;
* **/proc/ioports**Ce fichier est similaire à */proc/interrupts* mais pour les **ports d'entrées/sorties** ;
* **/proc/kcore**Ce fichier est remarquable par sa taille.   
  Il met à disposition la **totalité de la RAM de l'ordinateur** et est **requis pour le débogage du noyau**   
  Ce fichier nécessite des privilèges root pour la lecture ;
* **/proc/loadavg**Ce fichier contient **trois chiffres mesurant la charge du processeur au cours des dernières 1, 5 et 15 minutes**. Ces valeurs sont généralement **générées par le programme uptime** ;
* **/proc/meminfo**  
  Ce fichier contient des données relatives à l'**utilisation de la mémoire et du fichier d'échange**  
   Ce fichier est notamment **utilisé par l’utilitaire free** ;
* **/proc/mounts**Ce fichier est une autre **liste de tous les systèmes de fichiers montés actuellement  
  Il est essentiellement identiques à /etc/mtab ;**
* **/proc/scsi**Ce répertoire contient un fichier appelé *scsi* répertoriant les **périphériques SCSI disponibles**  
  Il existe un autre sous-répertoire pour chaque type d'adaptateur hôte SCSI du système contenant un fichier 0 (1, 2, …, pour plusieurs adaptateurs du même type) donnant des informations sur l'adaptateur SCSI ;
* **/proc/version**Ce fichier permet de récupérer les **informations concernant le numéro de version et la date de compilation du noyau actuel**.

**Remarque** ...

À l'époque où le répertoire */proc* n’existait pas, des programmes tels que l'outil d'affichage de l'état des processus, *ps*, qui devait accéder aux informations du noyau, devaient posséder une connaissance considérable des structures de données internes du noyau ainsi que les droits d'accès appropriés pour lire les données en question (depuis le noyau en cours d'exécution.   
Comme ces structures de données changeaient assez rapidement, il était souvent nécessaire d’installer une nouvelle version de ces programmes à chaque nouvelle version du noyau.   
Le système de fichiers */proc* sert de couche d'abstraction entre ces structures de données internes et les utilitaires.

**/sys – Contrôle du matériel**Le noyau Linux utilise ce répertoire **depuis la version 2.6**.   
Comme le répertoire */proc*, */sys* est rendu **disponible à la demande par le noyau lui-même et permet**, dans une hiérarchie étendue de sous-répertoires, une **vue cohérente du matériel disponible**.   
Il prend également en charge les **opérations de gestion sur le matériel via divers fichiers spéciaux**.

**Remarque** ...  
Théoriquement, toutes les entrées dans */proc* qui n'ont rien à voir avec des processus individuels devraient lentement migrer vers */sys*.

**/var – Fichiers changeant dynamiquement**Ce répertoire contient des **fichiers changeant dynamiquement**, répartis sur **différents répertoires**.   
Lors de l'exécution de différents programmes, l'utilisateur crée souvent des données (souvent sans s'en rendre compte).

Par exemple …

* La commande *man* provoque la décompression des sources de pages manuelles compressées, tandis que les pages *man* formatées peuvent être conservées pendant un certain temps au cas où elles seraient bientôt nécessaires.
* De même, lorsqu'un document est imprimé, les données d'impression doivent être stockées avant d'être envoyées à l'imprimante dans */var/spool/cups*.
* Les fichiers dans/var/log enregistre les heures de connexion et de déconnexion et d'autres événements système (fichiers journaux).
* Les fichiers de journalisation */var/spool/cron* contiennent des informations sur les invocations de commandes automatiques standard et le courrier électronique non lu des utilisateurs est conservé dans */var/mail*.

Sous Linux, les fichiers de **journalisation du système** sont généralement **gérés par le service *syslog***.   
**Un démon** (programme) appelé *syslogd* **accepte les messages d’autres programmes** et les trie en fonction de leur origine et de leur priorité. Il peut s’agir de l’aide au débogage, de message d’erreur et d’urgence, …).   
L’ensemble de ces **messages sont** **emmagasinés dans les fichiers situés dans le répertoire */var/log***.  
Outre les fichiers, le service *syslog* peut également écrire ses messages sur un hôte distant qui sert de station de gestion en centralisant tous les messages de journalisation d’un centre de données.

**Remarques** ...

Outre le *syslogd*, certaines distributions Linux contiennent également un service *klogd*.   
La tâche de ce service consiste à accepter les messages du noyau du système d'exploitation et à les transmettre à *syslogd*. Les autres distributions n’ont pas besoin d’un *klogd* séparé, leur *syslogd* pouvant faire ce travail lui-même.

Le noyau Linux émet toutes sortes de messages avant même que le système ne soit démarré suffisamment loin pour pouvoir exécuter *syslogd* (et éventuellement *klogd*) afin de les accepter. Étant donné que les messages peuvent toujours être importants, le noyau Linux les stocke en interne et il est possible d’y accéder à l'aide de la commande *dmesg*.

**/tmp – Fichiers transitoires**De nombreux utilitaires nécessitent un **espace fichier temporaire afin de compléter leur travail**.  
Tous les programmes, peuvent déposer des **données temporaires** dans */tmp*.   
De nombreuses distributions peuvent être **configurées pour nettoyer (vider) le répertoire /tmp au démarrage** du système. C’est un répertoire qui **ne devrait pas contenir de fichiers d’importance durable**.

**Remarque** ...  
Selon la tradition, */tmp* est vidé lors du démarrage du système mais pas le répertoire */var/tmp*.   
Une vérification est préférable de comment gère ce répertoire la distribution installée.

**/srv – Fichiers serveur**

On retrouve ici les **fichiers proposés par différents programmes serveur**, tels que **FTP** ou **HTTP**.

Ce répertoire est une invention relativement nouvelle et il est fort possible qu'il n'existe pas encore sur tous les systèmes.   
Malheureusement, il n'y a pas d'autre endroit évident pour les pages Web, les documents d'un serveur FTP, etc., sur lequel les auteurs FHS pourraient s'accorder (raison réelle de l'introduction de */srv*). Sur un système sans répertoire */srv*, ces fichiers peuvent se retrouver à un endroit complètement différent, e. g., dans des sous-répertoires comme */usr/local* ou */var*.

**/media – Accès aux cédéroms ou à des clés USB**Ce répertoire est **souvent généré automatiquement**.   
Lorsqu'un périphérique amovible (comme une carte mémoire SD ou une clé USB) est inséré dans une station, Linux permet d'y accéder à partir d'un sous-dossier de */media*.

**/mnt – Accès à d'autres supports de stockage**Ce répertoire (également vide) sert de **point de montage pour le montage à court terme de supports de stockage** supplémentaires. Avec certaines distributions, telles que celles de Red Hat, les points de montage de supports pour les cédéroms peuvent apparaître ici plutôt que sous */media*.

**/home – Répertoires personnels des utilisateurs**Ce répertoire contient les **répertoires personnels de tous les utilisateurs** sauf l’usager root dont le répertoire personnel étant situé ailleurs.

**Remarque** ...  
Si on a plus de quelques centaines d'utilisateurs, il est judicieux, pour la protection de la vie privée et l'efficacité, de ne pas conserver tous les répertoires personnels en tant qu'enfants immédiats de */home*.

Il est possible, par exemple, d’utiliser le groupe principal des utilisateurs comme critère de subdivision supplémentaire …

* /home/support/louis
* /home/developement/patrick
* ...

**/root – Répertoire de base de l'administrateur**Le **répertoire personnel de l’administrateur système** se trouve dans */root*.   
Il s’agit d’un répertoire personnel complètement normal, similaire à celui des autres utilisateurs, à la différence près qu’il ne se trouve pas sous */home* mais juste en dessous du répertoire racine (/).

On explique cet isolement car le répertoire des usagers */home* est souvent situé sur un système de fichiers sur une partition ou un disque dur séparé. Toutefois, l’usager *root* doit pouvoir accéder à son propre environnement utilisateur et ce, même si le système de fichiers séparé /home n'est pas accessible pour une raison quelconque.

**/lost+found– Objets trouvés**  
(systèmes de fichiers *ext(*2*.*3 *ou* 4*)* uniquement; non mandaté par FHS)

Ce répertoire est utilisé pour les **fichiers d’une certaine importance** mais qui **ne semblent pas appartenir à aucun répertoire**.

### Arborescence de répertoires et systèmes de fichiers

L’arborescence de répertoires d’un système Linux **s’étend généralement sur plus d’une partition sur un disque**, et les supports amovibles tels que les disques cédéroms et les clés USB, les appareils photo numériques, … doivent être pris en compte.

Sous Linux, toutes les partitions et tous les supports de disque disponibles sont **intégrés dans l’arborescence commençant à la racine** (/).

En général, rien n’empêche d'installer un système Linux complet sur une seule partition de disque dur.

Toutefois, il est **courant de placer au moins le répertoire de partitionnement /home**(où résident les répertoires personnels des utilisateurs) sur sa propre partition. L’avantage de cette approche est que si l’on doit réinstaller complètement le système d’exploitation actuel, cela permet d’isoler les données et à ne pas se soucier de la sécurité de ses propres données. Il suffit simplement de faire attention au moment de la sélection des partitions cibles. Cela simplifie également la création de copies de sauvegarde.

Sur des systèmes de serveur plus importants, il est également assez courant d’**attribuer leurs propres partitions à d’autres répertoires**, généralement des systèmes de serveur (*/tmp*, */var/tmp* ou */var/spool*).

L'objectif de cette isolation est d'**empêcher les utilisateurs de perturber le fonctionnement du système en remplissant complètement** les partitions importantes.

Par exemple, **si /var est saturé**, aucun **message de protocole ne peut être écrit sur le disque**. On veut donc empêcher les utilisateurs de remplir le système de fichiers avec de grandes quantités de courrier non lu, de travaux d'impression non imprimés ou de fichiers géants dans */var/tmp*.   
Toutefois, **l’utilisation de plusieurs partitions ont tendance à encombrer le système**.

# Droits – Fichiers Linux

## Droits du système de fichiers

Les **droits du système de fichiers d’un système basé sur UNIX** sont définis pour **trois catégories d’utilisateurs** …

* l’**utilisateur** **qui possède le fichier** (u) ;
* les **autres utilisateurs du groupe** à qui appartient le fichier (g) ;
* **tous les autres utilisateurs** (o) dont on parle aussi en tant que « monde entier » ou « tout le monde ».

Pour les **fichiers**, chaque droit correspondant permet les actions suivantes :

* le **droit en lecture** (r) permet à son propriétaire de voir le contenu du fichier ;
* le **droit en écriture** (w) permet à son propriétaire de modifier le fichier ;
* le **droit d’exécution** (x) permet à son propriétaire de lancer le fichier comme une commande.

Pour les **répertoires**, chaque droit correspondant permet les actions suivantes :

* le **droit en lecture** (r) permet à son propriétaire d’afficher le contenu du répertoire ;
* le **droit en en écriture** (w) permet à son propriétaire d’ajouter ou supprimer des fichiers de ce répertoires ;
* le **droit d’exécution** (x) permet à son propriétaire d’accéder aux fichiers du répertoire.

Ici, le droit en exécution sur un répertoire ne signifie pas uniquement l’autorisation de lire des fichiers dans ce répertoire mais aussi l’autorisation de voir leurs attributs, tels que leur taille et l’heure de modification.

La commande **ls** est utilisée pour afficher les informations de droits (et davantage) des fichiers et répertoires.   
Lorsque cette commande est passée avec l’option « *-l* », elle affiche les informations suivantes dans l’ordre donné :

* le **type de fichier** (premier caractère) ;
* l’**autorisation** **d’accès au fichier** (neuf caractères, constitués de trois caractères pour l’utilisateur, le groupe et « les autres », dans cet ordre) ;
* le **nombre de liens physiques** vers le fichier ;
* le **nom de l’utilisateur propriétaire** du fichier ;
* le **nom du groupe** à qui appartient le fichier ;
* la **taille du fichier en caractères** (octets) ;
* la **date** **et heure du fichier** (mtime) ;
* le **nom du fichier**.

|  |  |
| --- | --- |
| Caractère | Signification |
| - | fichier normal |
| d | répertoire |
| l | lien symbolique |
| c | nœud de périphérique en mode caractère |
| b | nœud de périphérique en mode bloc |
| F | tube nommé |
| s | socket |

*Liste des premiers caractères de la sortie de ls -l*

La **commande chown** est utilisée depuis le **compte de l’administrateur** pour **modifier le propriétaire d’un fichier**.   
La **commande chgrp** est utilisée **depuis le compte du propriétaire du fichier ou de l’administrateur pour changer le groupe** **du fichier**.

La **commande chmod** est utilisée depuis le **compte du propriétaire du fichier ou de l’administrateur** pour **modifier les** **droits d’acc**ès à un fichier ou un répertoire.

La syntaxe de base pour manipuler le fichier travail est la suivante :

**>> chown <nouvel\_utilisateur> travail  
>> chgrp <nouveau\_groupe> travail  
>> chmod [ugoa][+-=][rwxXst][,...] travail**

On peut, par exemple, faire qu’une arborescence de répertoires soit la propriété de l’utilisateur toto et partagée par le groupe titi en faisant ce qui suit …  
**>> cd <Répertooire>  
>> chown -R tux:gestionnaires .  
>> chmod -R ug+rwX,o=rX .**

### Sticky bit

Il existe un **bit particulier qui donne des droits particuliers** : le **bit collant sticky bit** (t ou T situé à la place du x)   
des autres (o).

**Positionner le sticky bit d’un répertoire** **empêche un fichier de ce répertoire d’être supprimé** par un **usager qui n’est pas le propriétaire du fichier**.

Pour **sécuriser le contenu d’un fichier** dans des **répertoires pouvant être écrits par tout le monde tels que /tmp** ou dans des **répertoires pouvant être écrits par le groupe**, il ne faut pas uniquement supprimer le droit du fichier en écriture mais aussi **positionner le sticky bit sur le répertoire**.

Sinon, le **fichier pourra être supprimé** et **un nouveau fichier créé avec le même nom** par un utilisateur ayant accès en écriture au répertoire.

Voici quelques exemples intéressants de droits de fichiers.

**>> ls -l /etc/passwd /etc/shadow /dev/ppp /usr/sbin/exim4**

**crw------- 1 root root 108, 0 oct. 20 07:12 /dev/ppp**

**-rw-r--r-- 1 root root 2718 oct. 5 22:09 /etc/passwd**

**-rw-r----- 1 root shadow 1741 oct. 5 22:09 /etc/shadow**

**-rwsr-xr-x 1 root root 973824 sept. 23 07:04 /usr/sbin/exim4**

**>> ls -ld /tmp /var/tmp /usr/local /var/mail /usr/src**

**drwxrwxrwt 17 root root 102400 oct. 20 16:42 /tmp**

**drwxrwsr-x 10 root staff 4096 nov. 29 2011 /usr/local**

**drwxr-xr-x 13 root root 4096 sept. 30 11:24 /usr/src**

**drwxrwsr-x 2 root mail 4096 sept. 27 11:07 /var/mail**

**drwxrwxrwt 5 root root 4096 oct. 20 09:14 /var/tmp**

### Mode numérique

Il existe aussi un mode numérique pour décrire les droits des fichiers avec **chmod**.   
Ce mode numérique utilise des nombres en base 8 (radix=8) codés sur 3 ou 4 chiffres.

|  |  |
| --- | --- |
| Chiffre | Signification |
| 1er chiffre (optionnel) | somme de set user ID (=4), set group ID (=2), et sticky bit (=1) |
| 2ème chiffre | somme des droits de lecture (=4), écriture (=2), et exécution (=1) pour l’utilisateur |
| 3ème chiffre | identique pour groupe |
| 4ème chiffre | identique pour autres |

*Mode numérique des droits de fichiers dans les commandes chmod*

Cela peut sembler compliqué mais c’est en fait assez simple. Si vous regardez les quelques premières colonnes (2-10) de la sortie de la commande « ls -l » et que vous lisez en représentation binaire (base 2) les droits des fichiers (le « - » représentant « 0 » et « rwx » représentant « 1 »), les trois derniers chiffres de la valeur numérique du mode devraient vous donner la représentation des droits du fichier en octal (base 8).

Essayez, par exemple, ce qui suit …  
**>> touch tux.texte tux.document  
>> chmod u=rw,go=r tux.texte  
>> chmod 644 tux.document>** **ls -l tux.texte tux.document** **-rw-r--r-- 1 root root 0 oct. 20 16:46 titi  
 -rw-r--r-- 1 root root 0 oct. 20 16:46 toto**

Les droits appliqués à un fichier ou à un répertoire venant d’être créé sont restreintes par la commande interne du *shell* **umask**.

|  |  |  |  |
| --- | --- | --- | --- |
| umask | Droits des fichiers créés | Droits des répertoires créés | Utilisation |
| 0022 | -rw-r--r-- | -rwxr-xr-x | ne peut être écrit que par l’utilisateur |
| 0002 | -rw-rw-r-- | -rwxrwxr-x | peut être écrit par le groupe |

*Exemples de valeurs de umask*

De manière générale, Linux utilise par défaut un schéma de **groupe privé par utilisateur** (UPG).

Un **UPG est créé** chaque fois qu’un **utilisateur est ajouté au système**.

Un UPG a le **même nom que l’utilisateur** pour lequel il a été créé et **cet utilisateur est le seul membre de l’UPG**.

Un principe d’UPG rend sûr le positionnement de umask à 0002 car **chaque utilisateur a son propre groupe privé**. (Sur certaines variantes d’UNIX, il est assez courant de faire appartenir tous les utilisateurs au même groupe users et, dans ce cas, c’est une bonne idée de définir umask à 0022 pour des raisons de sécurité).

**Astuce...**On **active UPG en plaçant umask 002 dans le fichier ~/.bashrc**.

## Droits pour les groupes d’utilisateurs (*group*)

Afin que des droits **attribués à un groupe soient appliquées à un utilisateur particulier**, il faut que cet **utilisateur soit déclaré membre du groupe**. La nouvelle configuration du groupe n’est **effective qu’après une [re]connexion** **de** **l’utilisateur** (**ou l’exécution de exec newgrp**).

Sous Linux, les **périphériques ne sont qu’un autre type de fichier**. Si on a des **problèmes pour accéder à des** **périphériques** tel qu’un cédérom ou une clé USB depuis le compte d’un utilisateur, il faudra **rendre cet utilisateur membre du groupe concerné**.

Certains **groupes importants fournis par le système** permettent à leurs membres l’**accès à des fichiers et des** **périphériques particuliers** sans avoir les privilèges de l’administrateur.

|  |  |
| --- | --- |
| Groupe | Description des fichiers et périphériques accessibles |
| dialout | accès complet et direct aux ports série */dev/ttyS[0-3]* |
| dip | accès limité aux ports série pour une connexion *dialup IP* (réseau commuté) vers des pairs de confiance |
| cdrom | lecteurs et graveurs de CD-ROM, DVD+/-RW |
| audio | périphérique audio |
| video | périphérique vidéo |
| scanner | dispositifs de numérisation (scanners) |
| adm | journaux de surveillance du système |
| staff | quelques répertoires où effectuer du travail d’administration de début : */usr/local*, */home* |

*Liste des groupes importants fournis par le système pour l’accès aux fichiers*

**Astuce**...  
Certains **groupes faisant partie du** système, permettent à leurs **membres d’exécuter des commandes particulières** **sans** **les privilèges de l’administrateur** (root).

Horodatage

Il existe trois types d’horodatage pour un fichier GNU/Linux.

|  |  |
| --- | --- |
| Type | Signification |
| mtime | date de modification du fichier (**ls -l**) |
| ctime | date de changement d’état du fichier (**ls -lc**) |
| atime | date de dernier accès au fichier (**ls -lu**) |

*Liste des types d’horodatage*

**Notes**...

* **Écraser un fichier** va **modifier tous les attributs** mtime, ctime et atime du fichier.
* **Modifier le propriétaire ou les droits d’un fichier va modifier les attributs** ctime et atime du fichier.
* Lire **un fichier va modifier l’attribut** atime du fichier.

## Liens

Il existe deux méthodes pour associer le fichier « toto » avec un nom de fichier différent « titi » :

* **Lien physique** ou *hard link*Nom dupliqué d’un fichier existant  
  **>> ln tux.texte tux.document**
* **Lien symbolique** ou *sym link*  
  Fichier spécial pointant vers un autre fichier par son nom  
  **>> ln -s tux.texte tux.document**

Consultez l’exemple suivant pour des modifications du nombre de liens et les subtiles différences dans le résultat de la commande **rm**.

**>> umask 002  
>> echo "Contenu d'origine" > tux.texte  
>> ls -li tux.texte** **1449840 -rw-rw-r-- 1 tux tux 18 oct. 20 16:50 tux.texte**

**>> ln tux.texte tux.document # lien physique  
>> ln -s tux.texte tux.rapport # lien symbolique  
>> ls -li tux.texte tux.document tux.rapport** **1449840 -rw-rw-r-- 2 tux tux 18 oct. 20 16:50 tux.texte  
 1449840 -rw-rw-r-- 1 tux tux 18 oct. 20 16:50 tux.document  
 1450180 lrwxrwxrwx 2 tux tux 18 oct. 20 16:55 tux.texte -> tux.document**

**>> rm tux.texte  
>> echo "Nouveau contenu" > tux.texte  
>> ls -li tux.texte tux.document tux.rapport** **1450183 -rw-rw-r-- 2 pingouin pingouin 18 oct. 20 16:56 tux.texte  
 1449840 -rw-rw-r-- 1 pingouin pingouin 18 oct. 20 16:50 tux.document  
 1450180 lrwxrwxrwx 2 pingouin pingouin 18 oct. 20 16:55 tux.texte -> tux.rapport**

**>> cat tux.document  
Contenu d'origine**

**>> cat tux.texte  
Nouveau contenu**

Un lien physique peut être mis en place à l’intérieur du même système de fichiers, il partage le même numéro d’inœud, ce que montre l’option *-i* de la commande **ls**.

Le lien symbolique a les permissions d’accès nominales au fichier de « *rwxrwxrwx* » comme il apparaît dans l’exemple ci-dessus, alors que les permissions d’accès effectives sont celles du fichier vers lequel il pointe.

**Attention**...  
En règle générale — à moins d’avoir une très bonne raison pour cela — il faudrait s’abstenir de créer des liens physiques ou des liens symboliques compliqués. Cela peut provoquer des cauchemars lorsque la combinaison logique des liens symboliques crée une boucle dans le système de fichiers.

**Note**…  
Il est généralement préférable d’utiliser des liens symboliques plutôt que des liens physiques à moins que vous n’ayez une bonne raison d’utiliser un lien physique.

Si on vient de passer à Linux depuis Windows, la bonne conception d’un système de fichiers UNIX comparé à l’équivalent le plus proche que sont les « raccourcis Windows » deviendra vite claire. Parce qu’il est implémenté dans le système de fichiers, les applications ne voient pas de différence entre un fichier lié et son original. Dans le cas de liens physiques, il n’y a vraiment aucune différence.

# Listes de contrôle

Une Liste de contrôle d’accès ou ACL (*Access Control List*) est une **liste de permissions** sur un fichier, un répertoire ou une arborescence, **ajoutée aux permissions classiques** (c'est-à-dire, techniquement, les permissions POSIX.1) de ce fichier. Ces permissions concernent des utilisateurs et/ou des groupes définis. La gestion des ACL sous Linux s'inspire de la norme POSIX 1003.1e (projet 17) mais ne la respecte pas entièrement.

Au moyen des ACL, on peut **étendre le nombre d'utilisateurs et de groupes ayant des droits sur un même fichier**. Dans le monde UNIX, chaque fichier ne peut normalement indiquer des permissions que pour un seul utilisateur et un seul groupe, qui s'opposent à une unique catégorie correspondant à tous les autres (ou le reste du monde).   
Avec les ACL, on peut (entre autres) **ajouter à un fichier d'autres utilisateurs et groupes et définir leurs droits séparément**.

Les ACL sont très utiles (voire indispensables) dans des environnements informatiques axés sur le travail collaboratif et mutualisé. De plus, leur utilisation avec SAMBA permet d'en étendre les capacités.

Dans les exemples qui suivent, les commandes peuvent être saisi en tant que root ou en utilisateur normal, selon la portée des modifications : pour modifier les droits d'un fichier qui n'est pas possédé par un utilisateur ou pour ajouter des permissions à un autre utilisateur ou encore pour ajouter aux ACL un groupe dont l'utilisateur ne fait pas partie, il faut être root.

La mise en place des ACL implique trois notions principales ...

* **ACL minimale**  
  Une ACL est composée exclusivement d’éléments de type propriétaire, groupe et autres.  
  L’ACL minimale est une traduction en ACL des droits d’accès traditionnels Unix ;
* **ACL étendue**  
  Une ACL étendue prolonge les droits de l’ACL minimale.   
  Elle contient au moins un élément de type **mask** et peut contenir des éléments de type utilisateur et/ou groupe ;
* **ACL par défaut**  
   Les ACL par défaut ne peuvent être appliquées qu’aux répertoires et définissent quels droits un objet du système de fichiers devra hériter (de son répertoire parent) lors de sa création.

## Exemple pratique

Soit un fichier /var/www/index.php(page d'index d'un site web, par exemple) dont les droits sont les suivants …  
**>>** ls -l /var/www/index.php  
 -rw-r----- 1 root www-data 18 2005-09-11 11:24 /var/www/index.php

Dns cet exemple, root est le propriétaire ; il peut le lire et le modifier ; le fichier est aussi accordé au groupe www-data (celui sous lequel, par exemple, tourne le serveur web), dont les membres peuvent le lire mais pas le modifier. Quant au reste du monde, il ne peut pas y accéder (le fichier contient des informations confidentielles telles qu'un mot de passe à une base de données MySQL).

L’objectif est de rendre le fichier accessible en lecture aux utilisateurs tux et kermit, en lecture et écriture à elmo et cookieMonster. On pourrait à la rigueur faire entrer tux et kermit dans le groupe www-data mais cela introduirait une faille de sécurité car www-data peut aussi accéder à des données qui ne les concernent pas. Il n'est en tout cas rationnellement pas prudent d'ajouter elmo et squiddly au groupe root. On ne peut non plus modifier les permissions (les mettre en lecture et écriture pour tout le monde) ou modifier le propriétaire et le groupe.

Les ACL sont une solution pratique et facile à gérer dans ce cas ; il suffit d'ajouter des permissions à l'ACL du fichier afin de qu'il se présente ainsi …

* root: rw- www-data: r--
* elmo: rw-
* cookiemonster: rw-
* tux: r--
* kermit: r--
* Autres : ---

## Mise en place

Selon la version du noyau utilisée, le **support des ACL peut ou non être déjà compilé**.   
S'il l'est, il faut aussi que la partition contenant les fichiers dont on veut étendre les droits soit montée avec l'option idoine.

### Noyau

Il faut d'abord savoir **si le** **noyau a été compilé avec le support des ACL**.   
Le plus simple est de le vérifier dans le **fichier de configuration du noyau**, fichier normalement situé sous **/boot**.   
Pour ce faire, utiliser la commande grep …  
**>>** grep ACL /boot/config-<Version du noyau>  
**>>** grep ACL /boot/config-4.19.0-6-amd64

CONFIG\_EXT4\_FS\_POSIX\_ACL=y

CONFIG\_REISERFS\_FS\_POSIX\_ACL=y

CONFIG\_JFS\_POSIX\_ACL=y

CONFIG\_XFS\_POSIX\_ACL=y

CONFIG\_BTRFS\_FS\_POSIX\_ACL=y

CONFIG\_F2FS\_FS\_POSIX\_ACL=y

CONFIG\_FS\_POSIX\_ACL=y

CONFIG\_TMPFS\_POSIX\_ACL=y

CONFIG\_JFFS2\_FS\_POSIX\_ACL=y

CONFIG\_NFS\_V3\_ACL=y

CONFIG\_NFSD\_V2\_ACL=y

CONFIG\_NFSD\_V3\_ACL=y

CONFIG\_NFS\_ACL\_SUPPORT=m

CONFIG\_CEPH\_FS\_POSIX\_ACL=y

CONFIG\_CIFS\_ACL=y

CONFIG\_9P\_FS\_POSIX\_ACL=y

Elle doit renvoyer la ligne suivante ...   
**CONFIG\_FS\_POSIX\_ACL=y** pour signaler que le support général des ACL est présent, et plusieurs lignes du type CONFIG\_<Système de Fichiers\_FS\_POSIX\_ACL=y où <Système de Fichiers peut recevoir les valeurs (pour un noyau 4.19.0-6 à la date de rédaction) EXT2, EXT3, JFS et XFS.

On peut aussi utiliser les ACL avec les systèmes de fichiers IBM JFS, ReiserFS, SGI XFS et NFS. Leur implémentation peut nécessiter de modifier (*patch*) le noyau.

**Remarque** …  
Les ACL ne sont pas possibles avec des systèmes de fichiers comme **vfat qui ne gèrent aucun type de permissions**.

## Commandes

Il existe deux commandes essentielles …

* l'une pour **manipuler l'ACL** d'un fichier (**setfacl**)   
  et
* l'autre pour la **consulter** (**getfacl**).

Les commandes traditionnelles **chmod et chown ne peuvent accéder aux ACL**.

Ces deux commandes nécessitent, sous Debian, l'installation du **paquet acl**.   
Pour installer le paquet …  
**>>** sudo apt install acl

Pour les distributions à base de RHEL ou SUSE, il faut installer les paquets acl.\*.rpm et libacl1.\*.rpm   
(leur nom contient leur numéro de version).

### Commande setfacl

Le nom de la commande se comprend comme set file's ACL.  
Elle possède de nombreuses options dont il convient de prendre connaissance en consultant la page de manuel   
(man setfacl).   
La commande fonctionne également de manière récursive (option -R).  
**setfacl -Rm u:tux:rw /var/www/** modifie l'ACL de tous les fichiers situés sous /var/www/ en attribuant une permission de lecture et d'écriture à l'utilisateur Tux.

#### Ajout de permissions

La commande **setfacl -m u:tux:rw /var/www/index.php** modifiera (-m) l'ACL de /var/www/index.php en attribuant à l'utilisateur (préfixe u:) tux les droits rw et en lui refusant le droit d'exécution (qui n'a pas été mentionné dans la commande).

Les principaux paramètres à connaître sont ...

* **Préfixes** …  
  u: (droits pour un utilisateur, nommé ou désigné par son uid) ;  
  g: (droits pour un groupe, nommé ou désigné par son gid) ;  
  o: (droits pour autres (*other*) ;
* **Autorisations**  
  Elles sont codées dans l'ordre r, w et x ou X (ce dernier représentant, comme avec chmod, le droit d'entrée dans les répertoires ou celui d'exécution pour les fichiers qui ont déjà un marqueur).   
  On les remplace par - pour une interdiction explicite.   
  Ne pas mentionner un droit revient aussi à une interdiction …  
  **setfacl -m u:tux:w /var/www/index.php** et **setfacl -m u:tux:-w- /var/www/index.php** reviennent au même.

On peut construire des commandes plus complexes en enchaînant les entrées dans l'ACL …  
**setfacl -m u:tux:rw,g:finances:r--,o:--- /var/www/index.php** définit des permissions dans l'ACL de /var/www/index.php pour l'utilisateur tux, le groupe finances et les autres .

Cette commande permet aussi de modifier les droits classiques (et remplace dans ce cas chmod) : l'utilisateur, le groupe et les autres initiaux du fichier sont simplement désignés par le préfixe (u:, g:, o:) suivi d'un nom vide.  
Si un fichier appartient à elmo:www-data avec les droits r--r-----, pour donner à l'utilisateur et le groupe les droits en lecture et écriture, il suffit d'une commande **setfacl -m u::rw,g::rw /var/www/index.php**.   
Si l'utilisateur et le groupe possèdent déjà un droit qui ne serait pas mentionné dans la commande setfacl, ce droit sera annulé. Par exemple le fichier index.php avec les droits rw-r----- pour elmo:www-data, la commande **setfacl -m u::r,g::x index.php** modifiera les droits à r----x--- pour pour elmo:www-data.

**Remarque** …  
Un fichier dont seules les permissions classiques ont été altérées par setfacl au lieu de chmod ne reçoit pas pour autant une ACL. De fait, il n'est pas référencé par ls -l comme fichier à ACL.

## Droits par défaut et héritage des droits étendus

Les **droits étendus d'un objet parent ne sont pas automatiquement hérités par les objets contenus**.   
Par exemple, si un répertoire (root:www-data, rwxr-x-r-x) possède une ACL u:elmo:rwx, un fichier créé à l'intérieur (ou déjà présent avant l'adjonction de l'ACL ne reçoit pas cette ACL et ses droits sont ceux impliqués par l'umask défini.

On peut modifier ce comportement en ajoutant, aux répertoires seulement, un attribut default, codé d:, qui se transmet à tous les fichiers créés dans le répertoire après l'ajout de l'ACL par défaut.   
Par exemple, **setfacl -m d:u:elmo:rwX /var/www** donne à elmo les droits de lecture et écriture (ainsi qu'exécution quand il s'agit de répertoires) pour tous les fichiers qui seront créés sous /var/www à partir de ce moment, jusqu'à ce que cette ACL par défaut soit annulé ou remplacé.

## Retirait de permissions

Pour annuler tout ou partie d'une ACL, la commande **setfacl -b /var/www/index.php** retire tout le contenu de l'ACL du fichier, tandis que **setfacl -x u:tux,g:finances/var/www/index.php** retire les permissions propres à tux et au groupe finances.

Les permissions ACL par défaut d'un répertoire (d:) s'annulent par **setfacl -k**.

## Masque

Le masque est une synthèse des valeurs les plus permissives que possède un fichier doté d'une ACL.   
Les droits classiques de l'utilisateur ne sont cependant pas pris en compte.   
Le masque est calculé automatiquement …  
**>>** chown elmo:www-data index.php   
**>>** chmod 640 index.php  
**>>** ls -l index.php  
 -rw-r----- 1 elmo www-data 5055 2005-10-16 18:53 index.php  
**>>** getfacl index.php  
 # file: index.php  
 # owner: luce  
 # group: www-data  
 user::rw-  
 group::r--  
 other::---

Ce fichier n'a pas d'ACL donc pas de masque.

**>>** setfacl -m u:kermit:rw,g:web:rw index.php  
**>>** getfacl index.php  
 # file: index.php

# owner: luce

# group: www-data

user::rw-

user:kermit:rw-

group::r--

group:web:rw-

mask::rw-

other::---

Maintenant que le fichier possède une ACL, il a reçu un masque.  
Les permissions les plus élevées (utilisateur exclu) étant rw, c'est aussi la valeur du masque.

L'intérêt du masque est de pouvoir limiter d'un coup toutes les permissions d'un fichier (étendues ou non), sauf celles du propriétaire ; on utilise pour cela le préfixe m: suivi du droit maximal à accorder …  
**>>** getfacl index.php  
 # file: index.php

# owner: tux

# group: www-data

user::rw-

user:cookiemonster:rw-

group::r--

group:web:rw-

mask::rw-

other::---

**>>** setfacl -m m:r index.php   
**>>** getfacl index.php  
 # file: index.php

# owner: tux

# group: www-data

user::rw-

user:cookiemonster:rw- #effective:r--

group::r--

group:web:rw- #effective:r--

mask::r--

other::---

Les valeurs modifiées sont indiquées par le commentaire effective: suivi des permissions effectives après l'application du masque (ici, jean et web n'ont plus que le droit r, la situation reste la même pour www-data).

L'existence même d'un masque renvoie au fonctionnement profond des ACL. Pour en comprendre l'utilité réelle sans se limiter à l'application pragmatique qui en est donnée ici, on se reportera à **POSIX Access Control Lists on Linux**.

### Commande getfacl

Cette commande suivie d'un nom de fichier affiche l'ACL de ce fichier (get file's ACL).   
Par exemple …  
**>>** getfacl /var/www  
 # file: var/www

# owner: root

# group: www-data

user::rwx

user:elmo:rwx

group::rwx

mask::rwx

other::r-x

default:user::rwx

default:user:kermit:rwx

default:group::rwx

default:group:www-data:r-x

default:mask::rwx

default:other::r-x

On voit qu'outre les droits traditionnels attribués à root:www-data (droits indiqués après user:: et group::), sont aussi définis ...

* des droits complets pour elmo (user:elmo:rwx) ;
* une permission ACL par défaut donnant des droits complets à kermit sur tous les nouveaux fichiers créés sous /var/www/ (default:user:kermit:rwx) ;
* une autre permission ACL par défaut donnant des droits de lecture et d'exécution au groupe www-data sur les mêmes fichiers (default:group:www-data:r-x).

Noter que user::, group:: et other:: représentent le triplet utilisateur/groupe/autres du monde des permissions classiques. Appliquer cette commande sur un fichier qui n'a pas d'ACL définie donne les mêmes informations que la commande **ls -l**, dans un format différent.   
La commande **setfacl -b index.php** permet de retirer les ACL pouvant exister.

### Commandes ls, cp et mv

Ces commandes doivent pouvoir lister, copier et déplacer les ACL en même temps que les fichiers.

Pour les deux premières commandes, il faut préciser explicitement que l'on veut afficher/conserver les droits (ce qui est aussi le cas quand on ne travaille que sur les droits classiques) ... **ls -l**, **cp -a**.   
La commande **mv**, quant à elle, préserve toujours les droits.

Lorsque les droits étendus ne peuvent être conservés (déplacement ou copie vers un système de fichier qui n'est pas configuré pour les recevoir ou utilisation d'une version de cp trop ancienne), un message d'avertissement en informe l'utilisateur.   
Par exemple …  
**>>** setfacl -m u:kermit:rw index.php cp -a index.php /mnt/vfat  
 cp: preserving permissions for `/mnt/vfat/index.php': Opération non supportée

Noter qu'un fichier comportant une ACL qu'on veut lister par ls -l n'affiche qu'un + à la suite de ses permissions. Seule la commande getfacl, pour l'instant, permet d'avoir connaissance du détail.

Par exemple …  
**>>** setfacl -m u:khadija:rw /var/www/index.php   
**>>** ls -l /var/www/index.php  
-rw-rw----+ 1 elmo www-data 5055 2005-10-16 18:53 /var/www/index.php

Avec -rw-rw----+, on sait que le fichier possède une ACL (+), sans en connaître les constituants.

## Sauvegarde des données

La sauvegarde des données dotées d'ACL nécessite ...

* l'utilisation d'un système de fichiers pour le stockage qui soit compatible ;  
  et
* l'utilisation d'un logiciel de sauvegarde qui soit tout autant compatible.

À titre indicatif,   
tar et cpio et rsync ne le sont pas (à moins d'être modifiés),   
star et pax le sont.

Pour contourner le problème de sauvegarde, il est possible d'écrire toutes les ACL dans un fichier qui servira de base à une restauration ultérieure. La commande **getfacl --skip-base -R /<Répertoire>/<Répertoire>/ > <Fichier>** récupère les informations récursivement et les inscrit dans un simple fichier.

La restauration se fait au moyen de **setfacl --restore=<Fichier>**.   
Il faut, pour qu'elle fonctionne, se placer à la racine contenant l'arborescence, en raison de la notation relative des chemins (d'où le message **Removing leading '/' from absolute path names** que l'on peut souvent lire en saissisant des commandes avec ces programmes).   
Le chemin d'un répertoire /tmp/test est enregistré comme tmp/test. On doit donc, pour restaurer, lancer la commande depuis la racine de /tmp, c'est-à-dire /.

Par exemple, le répertoire /tmp/test contient trois fichiers à ACL.   
On sauvegarde les ACL avec **getfacl --skip-base -R /tmp/test > acl.acl**.   
Pour restaurer, on se positionne à la racine (cd /) et on lance la commande **setfacl --restore=acl.acl**.   
Si on avait lancé la commande depuis /test, setfacl aurait renvoyé les erreurs …  
**setfacl: tmp/test: Aucun fichier ou répertoire de ce type setfacl: tmp/test/a: Aucun fichier ou répertoire de ce type setfacl: tmp/test/b: Aucun fichier ou répertoire de ce type setfacl: tmp/test/c: Aucun fichier ou répertoire de ce type**.

### Syntaxe de setfacl

Les préfixes abrégés peuvent être développés et les permissions codées en octal (avec préfixe 0 optionnel).   
Ces trois commandes ont donc le même sens …

* setfacl -m d:u:kermit:rw,g:www-data:r,o:- index.php
* setfacl -m default:user:kermit:6,group:www-data:4,other:0 index.php
* setfacl -m default:user:kermit:06,group:www-data:04,other:00 index.php

## Conclusion

Tout au long de ce document, on a parlé de permissions classiques opposées à des permissions étendues qui leur seraient ajoutées. C'est une simplification : dans les faits, un système de fichiers monté avec le support des ACL change son appréhension globale des permissions, qui sont toutes des ACL, qu'elles soient minimales (utilisateur primaire, groupe primaire, autres) ou étendues.   
Dans ce dernier cas, c'est la notion de groupe qui est redéfinie : tout utilisateur ou groupe ajouté à l'utilisateur primaire est enregistré avec ses droits dans une classe groupe étendue.

Cela explique la nécessité d'un masque : c'est la valeur limite réelle de la classe groupe, que les entrées qu'elle contient ne peuvent dépasser.   
La commande ls -l sur un fichier dont le masque ACL a été abaissé à r-- alors que les droits du groupe primaire étaient auparavant rw- donnera : -rw-r-----+ et non -rw-rw----+. Inversement, si l'on modifie par lacommande chmod les permissions du groupe principal, cela revient à changer le masque, donc les permissions de tous les utilisateurs et groupes ajoutés.

Utilisation de quotas de disque

**Introduction**La plupart des administrateurs système ont eu une partition système serveur saturée pour une raison ou une autre, généralement suivie de près par un plantage du système ou une panique du noyau.

L’utilitaire quota permet de **limiter l'utilisation du disque par utilisateur et par système de fichiers**.

**Avertissement** …  
Une limitation importante est que les **quotas se trouvent sur un système de fichiers** et **non sur une arborescence de répertoires**. Si on a deux arborescences de répertoires (/home et /var/www, par exemple) nécessitant des quotas conflictuels ou différents, ces **arborescences de répertoires doivent figurer sur des systèmes de fichiers distincts**, ce qui signifie des partitions séparées.

**Concepts de quotas**Les termes suivants sont utiles pour configurer les quotas …

* **Limite permisive** (*soft limit*)  
  Limite pouvant être dépassée, avec les avertissements qui en résultent jusqu'à la fin du délai de grâce ;
* **Limite stricte** (*hard limit*)Linite qui est généralement supérieure à la limite souple et ne peut pas être dépassée ;
* **Période de grâce** (*grace period*)  
  Limite souple peut être dépassée jusqu'à la limite maximale jusqu'à ce que la valeur de la période de grâce soit atteinte.   
  Ensuite, pour enregistrer davantage de données, la quantité utilisée doit être inférieure à la limite souple.

**Commandes et fichiers de quota**Le système de quotas est conçu pour **limiter les utilisateurs à une certaine quantité d’espace disque par système de fichiers**. La commande quota permet à un utilisateur de voir les quotas en place pour son compte.

Les commandes, paramètres et fichiers utilisés pour les quotas sont les suivants …

|  |  |
| --- | --- |
| Commande | Action |
| quotaon | Active les quotas |
| quotaoff | Désactive les quotas |
| quotacheck | Vérifie que l’utilisation de chaque utilisateur est correcte et met à jour les valeurs stockées si nécessaire |
| edquota | Modifie les quotas pour un utilisateur |
| quota | Affiche (pour utilisateur) les limites/l'espace des quotas |
| aquota.user | Fichier binaire contenant les informations de quota utilisateur |
| aquota.group | Fichier binaire contenant les informations de quota de groupe |
| usrquota | Option du fichier */etc/fstab* pour les quotas d'utilisateurs |
| grpquota | Option du fichier */etc/fstab* pour les quotas de groupe |

**Configuration des quotas**Pour configurer des quotas, il faut, dans un premier temps, **identifier un système de fichiers**.   
Il n'est pas recommandé de définir des quotas sur le système de fichiers racine (/), mais plutôt sur des systèmes de fichiers contenant les ensembles d'utilisateurs et de données les plus actifs.  
**>>****sudo****cat /etc/fstab  
/dev/mapper/turnkeyvm-root / ext4 errors=remount-ro 0 1  
/dev/mapper/turnkeyvm-swap\_1 none swap sw 0 0  
/dev/sdb /data ext4 rw 0 0**

Pour configurer des quotas sur le système */data* …

1. Vérification que la partition */data* est montée en émettant la commande mount et en inspectant la sortie …  
   **>> sudo mount**   
   **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=240188k,nr\_inodes=60047,mode=755)  
   … Résultat tronqué pour l’affichage …  
   /dev/sdb on /data type ext4 (rw,relatime,quota,usrquota,grpquota,data=ordered)**
2. Modification du fichier */etc/fstab* et ajout des paramètres *usrquota* et *grpquota* afin d’activer les quotas d'utilisateurs et de groupes …  
   **/dev/sdb /data ext4 defaults,usrquota,grpquota 0 0**
3. Remontage du système de fichiers /data avec la commande …  
   **>> sudo mount -o remount,rw /data  
   ou  
   >> sudo mount -a**  
   Ces commandes activent les options de quota qui ont été ajoutées précédemment sans mettre le système de fichiers hors ligne.
4. Vérification que le système de fichiers dispose des options correctes (*usrquota* et *grpquota*) …  
   **>> sudo mount | grep usrquota   
   /dev/sdb on /data type ext4 (rw,relatime,quota,usrquota,grpquota,data=ordered)**
5. Ajout d’un utilisateur normal pour tester les quotas avec les prochains éléments …  
   **>> sudo useradd –m quotatux  
   >> sudo passwd quotatux**
6. Mise à jour des fichiers aquota. \*   
   **>> sudo quotacheck -avugc**  
   **quotacheck: Your kernel probably supports journaled quota but you are not using it. Consider switching to journaled quota to avoid running quotacheck after an unclean shutdown.  
   quotacheck: Scanning /dev/sdb [/data] done  
   quotacheck: Cannot stat old user quota file /data/aquota.user: No such file or directory. Usage will not be   
   ubtracted.  
   quotacheck: Cannot stat old group quota file /data/aquota.group: No such file or directory. Usage will not be subtracted.  
   quotacheck: Cannot stat old user quota file /data/aquota.user: No such file or directory. Usage will not be subtracted.  
   quotacheck: Cannot stat old group quota file /data/aquota.group: No such file or directory. Usage will not be subtracted.  
   quotacheck: Checked 2 directories and 0 files  
   quotacheck: Old file not found.  
   quotacheck: Old file not found.**
7. Modification des paramètres de quota pour un utilisateur …  
   **>> sudo edquota –u quotatux  
   Disk quotas for user quotatux (uid 1002):  
   Filesystem blocks soft hard inodes soft hard  
   /dev/sdb 0 0 0 0 0 0**
8. Définition des quotas de bloc de quotatux pour correspondre aux éléments suivants …  
   **Filesystem blocks soft hard inodes soft hard  
   /dev/sdb 0 5000 6000 0 0 0**
9. Enregistrement et sortie de l’éditeur.  
   Le tout sera chargé correctement par le système de quota***.***
10. Connexion en tant *quotatux*  
    (ou encore avec l’utilisation de la commande su - pour devenir quotatux)  
    **>> su – quotatux**
11. Création d’un nouveau fichier dans le répertoire /data en tant que quotatux …  
    **>> touch /data/quotatux.texte  
      
    Attention** …  
    Il faudra au besoin modifier les droits (permissions) sur le dossier data.
12. Vérification du quota répertorié pour l’usager quotatux (en tant qu'utilisateur root) …  
    **>>****sudo****quota quotatux  
    Disk quotas for user quotatux (uid 1002): none**
13. Activation du système de quotas pour le système de fichier */data*.   
    Auparavant, le système suivait les quotas mais ne les appliquait pas.  
    **>> sudo quotaon /data**
14. Connexion en tant quotatux  
    (ou encore avec l’utilisation de la commande su - pour devenir quotatux)  
    **>> su – quotatux**
15. Exécution de la commande quota pour voir l’utilisation actuelle de quotatux  
    **>> quota**  
    **Disk quotas for user quotatux (uid 1002): none**
16. Copie de tous les fichiers ordinaires du répertoire */etc* dans le répertoire */ data* (en tant que quotatux) …  
    **>> cp /etc/\* /data**
17. Vérification du bon fonctionnement des quotas …  
    **>> sudo quota  
    Disk quotas for user quotatux (uid 1002):  
    Filesystem blocks quota limit grâce files quota limit grâce  
    /dev/sdb 452 5000 6000 83 0 0**

**Astuces** …  
La distribution doit veiller à activer les quotas au démarrage.  
Il est possible de le tester en redémarrant et en vérifiant les quotas  
Si tel n’est pas le cas, il faudra ajouter /sbin/quota aux scripts de démarrage.  
**>> sudo systemctl enable quota  
Synchronizing state of quota.service with SysV service script with /lib/systemd/systemd-sysv-install.  
Executing: /lib/systemd/systemd-sysv-install enable quota**

**Limites strictes et permissives**

L’édition du quota utilisateur de l’usager quotatux (*edquota –u quotatux*) implique la connaissance des paramètres suivants …

|  |  |
| --- | --- |
| Paramètre | Description |
| Filesystem | Système de fichiers sur lequel l'utilisateur dispose d'un quota |
| blocks | Nombre de blocs actuellement utilisés par l'utilisateur sur le système de fichiers en question |
| soft | Limite logicielle qui peut être dépassée pendant la période de grâce \* |
| hard | Limite stricte à ne pas dépasser |
| inodes | Nombre d'inodes actuellement utilisés |
| soft | Nombre d'inodes pour lequel on veut définir une limite souple (À utiliser avec prudence) |
| hard | Limite stricte pour les inodes qui, si elles sont respectées, empêchent l'utilisateur de créer davantage de fichiers |

Les champs modifiables sont les paramètres logiciels et matériels pour les blocs et les inodes.   
Modifier l'une des autres valeurs ne fait rien.

**Réglage du délai de grâce**  
L’utilisation de la commande *edquota –t* permet de définir le délai de grâce des utilisateurs.   
Il est configurable en jours, heures, minutes ou secondes …  
**>> sudo edquota -t**  
**Grace period before enforcing soft limits for users:  
Time units may be: days, hours, minutes, or seconds  
Filesystem Block grace period Inode grace period  
/dev/sdb 7days 7days**

Pour modifier une période de grâce, il suffira d’en modifier les paramètres.

**Informations sur les quotas**La commande *quota*, exécutée en tant qu’utilisateur normal. Permet d’obtenir uniquement les informations de quota de cet utilisateur. S'il n'y a pas de quotas pour cet utilisateur, la sortie indique qu'il n'y en a pas.

L'utilisateur root a la capacité d'obtenir des données statistiques de quota sur tous les utilisateurs …  
**>> sudo repquota -a**  
**Grace period before enforcing soft limits for users:**

**\*\*\* Report for user quotas on device /dev/sdb  
Block grace time: 7days; Inode grace time: 7days  
 Block limits File limits  
User used soft hard grace used soft hard grace  
----------------------------------------------------------------------  
root -- 20 0 0 2 0 0  
quotatux -- 452 5000 6000 83 0 0**

**En vrac**

* L’option ***-p*** (***--print-state***) de la commande *quotaon* affiche les systèmes de fichiers pour lesquels les quotas sont activés …  
  **>> sudo quotaon -p /data  
  group quota on /data (/dev/sdb) is on  
  user quota on /data (/dev/sdb) is on**
* La commande *quotacheck* est utilisée pour mettre à jour les fichiers de base de données des quotas.  
  Pour la mise à jour de toutes les bases de données …  
  **>> sudo quotacheck -a  
  Remarque** …  
  Les quotas doivent être désactivés avant de passer la commande …  
  **>> sudo quotaoff -a**
* Afin de modifier les quotas d’un usager de manière non-interactive, on utilisera la commande *setquota*.
* La commande *quotacheck* permet de créer les quotas d’usager et de groupe.  
  Pour ce faire, on utilisera les options -u (usager) et -g (groupe).
* Les fichiers *aquota.user* et *aquota.group* sont les fichiers de base de données créés à l’aide de la commande *quotacheck*.
* La commande ***edquota -p*** permet de copier les limites de quotas d’un usager à un autre.  
  La commande ***edquota -t*** permet de modifier la période de grâce.

# Annexe 01 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 02 Utilitaire lshw

lshw est un petit outil pour extraire des informations détaillées sur la configuration matérielle de l’hôte. Il 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 03 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 04 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 …  
**>>** **sudo** **dd if=/dev/zero of=/swapfile bs=1024 count=65536**

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

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

Pour activer le fichier d'échange immédiatement mais pas automatiquement au démarrage …  
**>>** **sudo** **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) …  
**>>** **sudo** **swapoff -v /swapfile**

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

Comme dernière étape on supprime le fichier sur disque …  
**>>** **sudo** **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.