

CHAPITRE 5 : GESTION DES PERIPHERIQUES

5.1 Les périphériques :

Un système de fichiers est une façon d'organiser et de stocker une arborescence sur un support (disque, disquette, cd ...). Chaque OS propriétaire a développé sa propre organisation.

On peut faire cohabiter plusieurs systèmes dans des partitions d'un même disque.

Linux possède son système appelé **ext2** mais peut en gérer d'autres. La liste en est donnée dans `/proc/filesystems`. L'utilisateur peut donc accéder sous Linux à d'autres systèmes de fichiers, comme DOS, Vfat, ... provenant d'un périphérique ou importé par le réseau.

Comme pour l'utilisateur *tout est fichier*, tous les systèmes de fichiers quels que soient leur emplacement physique doivent être intégrés dans l'UNIQUE arborescence logique du système Linux. Cette arborescence peut donc être construite (et évoluer) à partir de diverses partitions qui peuvent être situées sur plusieurs disques. Cela réalise une intégration et une abstraction plus poussée que dans le monde Windows où les partitions et lecteurs auxquels sont affectées les lettres A: C: D: ... demeurent des entités séparées. Naturellement la partition sur laquelle est situé le répertoire racine joue un rôle particulier.

On appelle unités logiques, les unités de stockage (qui peuvent être locales ou distantes) : disques durs, lecteurs de disquettes, CDROM ... Ces unités contiennent leur propre arborescence de fichiers.

Sous Linux, ces unités logiques sont accessibles à partir de l'arborescence générale du système à condition d'être rattaché à cette arborescence. C'est ce qu'on appelle le "**montage**". Chaque unité peut être placée n'importe où dans l'arborescence (dans un répertoire existant ou à créer) et le "**montage**" peut être permanent ou temporaire (cas des périphériques amovibles : CDROM, disquette).

La commande "**mount**" permet d'effectuer ce montage. Elle est associée à la commande "**umount**" qui permet, le "**démontage**". Les unités logiques "**montables**" doivent être décrites dans `/etc/fstab`.

5.1.1 Nommage des périphériques sous linux :

Linux utilise un système de nommage des périphériques très différent de ce que l'on peut trouver sous MS Windows ®. Sous Windows ®, chaque partition et lecteur est associé à une lettre A:,C: ... Z: et quand il y'en a plus de 26 ?

Dans les systèmes UNIX, les périphériques, les disques durs et les partitions sont gérées sous forme de fichiers contenus dans le répertoire `/dev` : chaque fichier du répertoire `/dev` correspond à une sorte de driver.

On distingue deux types de disques : les IDE et les SCSI. Ils sont nommés respectivement "hd" et "sd" suivit d'une lettre et d'un chiffre. La lettre représente la position du disque sur le bus et le chiffre représente l'une des partitions de ce disque. Ainsi `/dev/hda` correspond au premier disque dur IDE, `/dev/hda1` à la première partition du disque dur IDE, `/dev/hda2` à la seconde partition, `/dev/hdb` au second disque dur IDE, `/dev/hdb1` à la première partition du second disque dur, etc..

De même /dev/sda correspond au premier disque dur SCSI, /dev/sda1 à la première partition du disque dur SCSI, etc..

/dev/cdrom correspond souvent au lecteur de cd-rom et /dev/fd0 au premier lecteur de disquette.

hda1, fait référence à /dev/hda1 :

hda1 : désigne la première partition primaire du disque maître sur l'IDE 0

hdd : désigne le disque esclave sur l'IDE 1

hdc5 : désigne la première partition logique du disque maître sur l'IDE 1

sda : désigne le premier disque dur scsi

sdb2 : désigne la deuxième partition du deuxième disque dur scsi

sdd6 : désigne la deuxième partition logique du quatrième disque dur scsi

Notez là encore, que les partitions logiques commencent à 5 et ce même s'il n'y a pas 4 partitions primaires.

scd0 ou sr0 : désigne le premier lecteur de CDROM/DVDROM SCSI

fd0 : désigne le premier lecteur de disquette

Le processus de montage, avec sa commande mount, décrite plus loin, est le moyen de faire correspondre parties de l'arborescence et partitions physiques de disque. Il permet de plus d'affecter tout système extérieur (disquette, cdrom, dk zip, rép. réseau ...) à un répertoire créé pour cela dans l'arborescence. Il suffira ensuite de se déplacer à ce répertoire, appelé point de montage, en fait un répertoire "d'accrochage", pour accéder à ses fichiers (bien sûr, conformément aux permissions que possède l'utilisateur)

5.2. Systèmes de fichiers et points de montage

Pour connaître aisément le fonctionnement de ces systèmes, prenons un cas pratique. Supposons que vous veniez juste d'acheter un disque dur flambant neuf, encore vierge de toute partition. Votre partition consacrée à **Mandrake Linux** est pleine à ras bord, et plutôt que de tout refaire à partir de zéro, vous décidez de déplacer toute une partie de l'arborescence sur votre nouveau disque dur. Votre nouveau disque dur étant très gros, vous décidez d'y déplacer votre répertoire le plus encombrant : /usr. Avant d'expliquer la solution, faisons un pas en arrière pour comprendre les principes de base.

5.2.1. Principes

Chaque disque dur est divisé en plusieurs partitions et chacune de ces partitions contient un système de fichiers. Tandis que windows associe une lettre à chacun de ces systèmes de fichiers, Linux possède une arborescence unique, et chacun des systèmes de fichiers est **monté** à un endroit de l'arborescence.

De même que windows a besoin d'un « lecteur C: », Linux a besoin de pouvoir monter la racine de son arborescence (/) quelque part, en l'occurrence sur une partition qui contient le **système de fichiers racine**. Une fois la racine montée, on peut monter d'autres systèmes de fichiers sur différents **points de montage** qui existent dans l'arborescence. N'importe quel répertoire en dessous de la racine peut faire office de point de montage, et vous pouvez monter le même système de fichiers plusieurs fois et ce sur différents points de montage. Cela autorise une grande souplesse dans la configuration. Dans le cas d'un serveur Web, par exemple, il est courant de consacrer une partition entière au répertoire hébergeant les données du serveur. Le répertoire qui les contient est en règle générale /var/www ; ce dernier fera donc office de point de montage pour la partition.

Vous pouvez voir dans les figures 5-1 et 5-2 la situation du système avant et après le montage des fichiers.

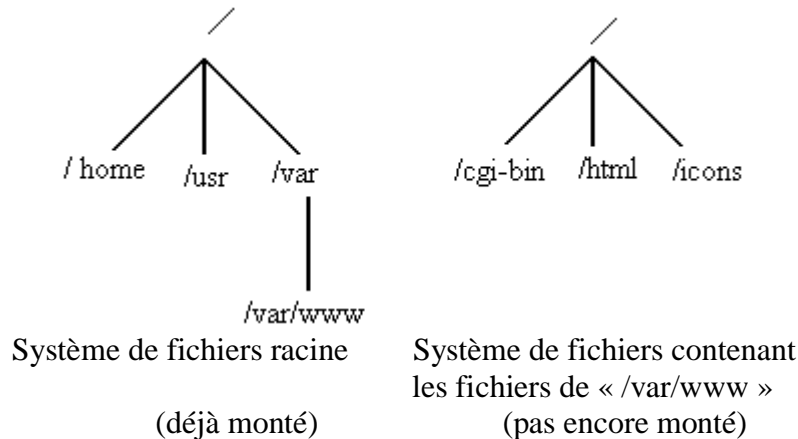


Figure 5-1. Avant le montage du système de fichiers

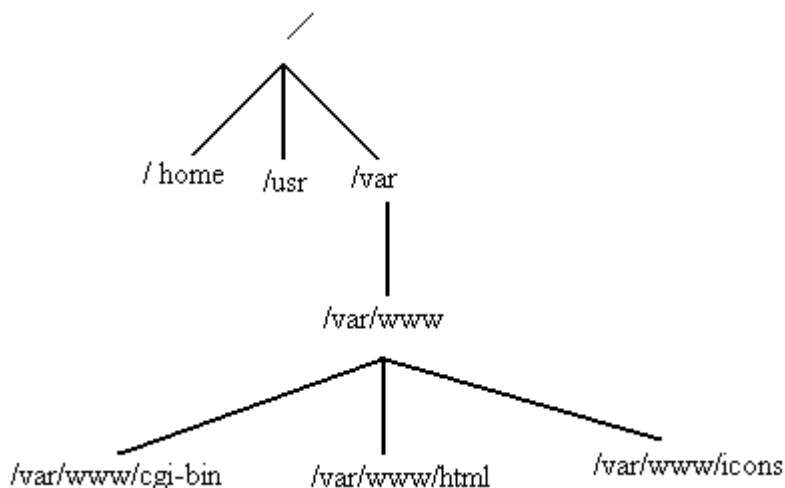


Figure 5-2. Après le montage du système de fichiers

Comme vous pouvez l'imaginer, cela présente de nombreux avantages : l'arborescence sera toujours la même, qu'elle s'étende sur un seul système de fichiers ou plusieurs dizaines, et il est toujours possible de déplacer physiquement une partie encombrante de l'arborescence sur une autre partition quand la place vient à manquer, ce que nous allons faire ici. Il faut savoir deux choses sur les points de montage :

1. le répertoire faisant office de point de montage doit exister;
2. ce même répertoire **devrait être vide** : si un répertoire choisi comme point de montage contient déjà des fichiers et sous-répertoires, ceux-ci seront tout simplement « cachés » par le système de fichiers nouvellement monté, mais ils ne seront pas effacés. Ils seront tout simplement inaccessibles jusqu'à ce que vous libériez ce point de montage.

En fait, il est possible d'accéder aux données cachées par le système de fichiers nouvellement monté. Vous n'avez qu'à monter le répertoire caché avec l'option `--bind`. Par exemple, si vous venez tout juste de monter le répertoire `/cache/repertoire/` et que vous voulez accéder à son contenu original dans le répertoire nouveau `/nouveau/repertoire/`, vous devrez taper la commande :

```
mount --bind /cache/repertoire/ /nouveau/repertoire
```

5.3. Partitionner un disque dur, formater une partition

À la lecture de cette section, gardez à l'esprit deux choses : un disque dur est divisé en partitions, et chacune de ces partitions héberge un système de fichiers. Votre disque dur tout

neuf ne possède ni l'une ni l'autre, donc nous commencerons par le partitionnement. Afin de procéder, vous devez être root. Premièrement, vous devez connaître le « nom » de votre disque dur (par exemple ; quel fichier le désigne).

Supposons que le nouveau disque dur est réglé en tant qu'esclave sur votre interface IDE principale. Dans ce cas, son nom sera /dev/hdb.2.

Remarque : Linux peut gérer jusqu'à 64 systèmes de fichiers montés en même temps.

5.3.1 Partitions

Un disque ne peut contenir que 4 partitions primaires. Mais il est possible de remplacer une partition primaire par une partition étendue. Cette partition étendue peut contenir maximum 12 partitions logiques. Notre disque pourra donc, au maximum, contenir 15 partitions utilisables soit 3 primaires plus les 12 partitions logiques. Cette méthode est la plus utilisée, indépendamment du système d'exploitation.

Sous Linux, contrairement à Windows, chaque partition n'est pas représentée sous forme de lecteur. Nous allons pouvoir monter, c'est à dire ajouter cet espace disponible à notre système de dossier hiérarchique.

Par exemple si notre machine fait office de serveur FTP, nous pourrons créer une partition dédiée qui sera montée dans "/var/ftp/". Ainsi le serveur FTP n'utilisera pas la place réservée aux utilisateurs ou aux fichiers de logs.

Il faudra aussi prévoir une partition réservée au swap.

5.3.2 Partitionner un disque dur

Il existe plusieurs outils pour partitionner un disque dur sous linux. Les plus communs sont :

- **fdisk** : utilitaire en mode texte avec menu interactif
- **cfdisk** : utilitaire en mode curses (pseudo graphique) avec une liste des partitions.

Ces deux utilitaires se lancent de la même façon :

commande /dev/disk

Il est possible de modifier les partitions d'un disque dur en cours d'utilisation. Mais vous devrez redémarrer pour que les changements soient pris en compte.

Voici un exemple avec fdisk. Nous allons créer 1 partition primaire contenant le swap (512Mo) et une partition logique contenant "/" (5Go)

```
root@localhost
# fdisk /dev/hda
Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-2434, default 1):
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-2434, default 2434): +512M
Command (m for help): t
Selected partition 1
Hex code (type L to list codes): 82
```

```

Changed system type of partition 1 to 82 (Linux swap)
Command (m for help): n
Command action
  e  extended
  p  primary partition (1-4)
e
Partition number (1-4): 2
First cylinder (64-2434, default 64):
Using default value 64
Last cylinder or +size or +sizeM or +sizeK (64-2434, default 2434):
Using default value 2434
Command (m for help): n
Command action
  l  logical (5 or over)
  p  primary partition (1-4)
l
First cylinder (64-2434, default 64):
Using default value 64
Last cylinder or +size or +sizeM or +sizeK (64-2434, default 2434): +5000M
Command (m for help): p
Disk /dev/hda: 20.0 GB, 20020396032 bytes
255 heads, 63 sectors/track, 2434 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
   Device Boot   Start    End  Blocks    Id  System
/dev/hda1       1       63   506016    82  Linux swap
/dev/hda2       64    2434  19045057+   5  Extended
/dev/hda5       64    672   4891761   83   Linux
Command (m for help): w
Writing table partition

```

m	Aide
p	Affiche l'état du disque
n	Crée une nouvelle partition
d	Supprime une partition
t	Change le type de partition (82 : Linux Swap, 83 : Linux Native)
w	Sauver et quitter
q	Quitter sans sauver

5.4.3 Formater une partition

Une fois nos partitions créées, il faut les formater afin de pouvoir créer un système de fichier dessus. Le système de fichier indique, entre autres, comment organiser les données sur le disque dur.

Dans la majorité des cas, nous utiliserons la commande mkfs. Certains systèmes de fichiers utilisent leur propre commande, nous les verrons ci-dessous.

Il existe deux moyens d'utiliser la commande mkfs :

mkfs -t type /dev/partition

mkfs.type /dev/partition

Voici un tableau des systèmes de fichiers utilisables sous linux avec la commande à utiliser.

swap	Mkswap
ext2	mke2fs, mkfs -t ext2, mkfs.ext2
ext3	mke2fs -j, mkfs -t ext3, mkfs.ext3
reiserfs	Mkreiserfs
xfs	mkfs -t xfs, mkfs.xfs
jfs	mkfs -t jfs, mkfs.jfs
vfat	mkfs -t vfat, mkfs.vfat

5.4 Access aux périphériques de stockage

Une fois le disque dur partitionné et formaté, nous allons pouvoir l'utiliser. Comme dit précédemment, pour accéder à ce disque nous allons devoir le monter. Le terme monter signifie que nous allons ajouter un disque ou un partage réseau à notre arborescence.

Concrètement, nous allons définir un répertoire, appelé point de montage, qui va pointer vers la ressource voulue. Par convention, les points de montage sont regroupés dans le dossier `"/mnt"`.

5.5 commande mount / umount

- La commande mount

Par soucis de cohérence, il est préférable de monter toutes vos partitions (et CDROM, disquettes ...) externes au système dans le répertoire `/mnt` qui est prévu pour cela. Comme vous avez potentiellement plusieurs périphériques à monter, il est préférable de créer un répertoire pour chacun (avec un nom explicite) dans `/mnt/`. Quelques exemples :

`/mnt/floppy` : sert généralement pour la disquette

`/mnt/cdrom` : sert généralement pour le cdrom

Voici en quoi consiste la syntaxe de mount, qui permet le montage de périphériques :

mount -t [système de fichiers] -o [options] [emplacement du périphérique] [dossier de montage]

Les systèmes de fichiers courant :

vfat : Partition FAT32/FAT16

ntfs : Partition NTFS(lecture seule).

ext2 : Partition ext2

ext3 : Partition ext3

reiserfs : Partition reiserfs

iso9660 : CDRom

auto : s'utilise pour les disquettes et cdroms permet au système de déterminer lui même le système de fichiers.

Les options :

ro : pour read only (lecture seule)

rw : pour read write (accès total)

Un exemple :

`mount -t vfat /dev/hda1 /mnt/win98` : me permet de monter ma partition fat32 de Windows 98 ® (partition 1 du disque maître de l'IDE 0) dans le répertoire `/mnt/win98`. Notez que je n'ai pas utilisé de **-o rw** car par défaut, les partitions sont montées, en read write. Ceci implique que dans le cas d'une partition NTFS il faudra taper :

`mount -t ntfs -o ro /dev/hda1 /mnt/winxp`

Car le driver linux pour le NTFS n'est utilisable qu'en lecture seule.

La commande mount va permettre de monter un périphérique. Si l'on utilise la commande mount sans paramètres, on obtiendra la liste des périphériques déjà montés.

Syntaxe :

mount -t type /dev/partition /mnt/dir

L'option -t renseigne le système de fichier utilisé. On peut mettre auto pour que mount essaye de trouver le système de fichier lui-même.

Remarque : Pour connaître tous les systèmes de fichier supportés par votre noyau, consultez le fichier "/proc/filesystems"

Il est possible de passer plus d'options grâce à l'argument -o (voir tableau ci-dessous)

- **La commande umount :**

La commande umount va, au contraire, démonter un périphérique.

Syntaxe :

umount /mnt/dir

C'est la commande inverse, qui sert à démonter un périphérique. Elle est indispensable, avant d'éjecter un périphérique vous devez d'abord le démonter.

Pour un CDROM, vous y serez contraint car le bouton eject ne fonctionnera pas, pour une disquette si vous ne le faites pas, il y a des chances que les données inscrites sur votre disquette soient incomplètes.

Pour reprendre notre exemple plus haut, si je souhaite démonter Windows 98 ®, il me suffit de taper l'une ou l'autre de ces commandes :

umount /dev/hda1 ou umount /mnt/win98

5.6 fichier /etc/fstab

Tout cela peut devenir très vite fastidieux, c'est pourquoi, il existe /etc/fstab. Fstab (file system table) est un fichier qui permet de définir les périphériques montables. Il est possible d'y définir de les monter automatique (à l'amorçage) ou quand vous le voulez.

Ce fichier va nous être utile pour deux choses. Premièrement, on va pouvoir y entrer toutes les partitions à monter automatiquement au démarrage. Deuxièmement, faciliter l'utilisation de la commande mount, en entrant toutes les options à utiliser pour chaque périphérique.

Dans le fichier "/etc/fstab" chaque ligne représente un périphérique à monter. Chaque ligne est composée de six parties :

- le chemin du périphérique
- le point de montage
- le système de fichier
- les options (argument -o de mount)
- dump : activer ou non la sauvegarde du disque avec l'utilitaire dump (1 ou 0)
- fsck : activer ou non la vérification du système de fichier avec l'utilitaire fsck (0, 1, 2 ou 3)

Voici un exemple de fstab :

```
#####SWAP#####
```

```
/dev/hda6 swap swap defaults 0 0
/dev/hdb5 swap swap defaults 0 0
#####LINUX#####
/dev/hdb4 / reiserfs defaults,notail 0 1
/dev/hdb2 /mnt/woody ext3 defaults 0 0
/dev/hda4 /mnt/gentoo ext3 defaults 0 0
/dev/hda9 /mnt/softs ext2 defaults 1 2
#####PERIPHS#####
/dev/fd0 /mnt/floppy auto defaults,user,noauto 0 0
/dev/hdc /mnt/cdrom auto defaults,ro,user,noauto 0 0
/dev/hdd /mnt/dvdrom auto defaults,ro,user,noauto 0 0
#####WINDOWS#####
/dev/hda1 /mnt/shared vfat user,auto,rw,exec,umask=0 0 0
/dev/hda5 /mnt/win98 vfat user,auto,rw,exec,umask=0 0 0
/dev/hda8 /mnt/multimedia vfat user,auto,rw,exec,umask=0 0 0
/dev/hda3 /mnt/win2k ntfs user,noauto,ro,exec,umask=0 0 0
```

Chaque ligne suit la syntaxe suivante :

[emplacement du périphérique][dossier de montage][système de fichiers][options][options annexes]

Voyons à quoi correspondent les options :

defaults : permet de monter le système de fichier avec les options standard : rw, suid, dev, exec, auto, nouser et async.

notail : Est une option qu'il faut rajouter à toute partition / ou (/boot) au format reiserfs, cela permet à lilo d'amorcer le système d'exploitation qui s'y trouve. En principe les versions récentes de Lilo n'ont plus ce problème.

user : Autorise les utilisateurs (autres que root) à monter la partition.

nouser : Restreint le montage de la partition à root.

auto : Autorise la partition à être montée à l'amorçage du système.

noauto : Empêche la partition d'être montée à l'amorçage du système. Elle pourra l'être plus tard.

ro : La partition sera montée en read only

rw : La partition sera montée en read write

exec : Autorise l'exécution de binaire

noexec : Empêche l'exécution de binaire

async : Les entrées/sorties seront gérées de manière asynchrone.

noasync : Les entrées/sorties seront gérées de manière synchrone.

suid : Autorise l'utilisation des uid,gid sur la partition.

nosuid : Empêche l'utilisation des uid,gid sur la partition.

dev : Autorise l'usage des fichiers spéciaux (contenu du repertoire /dev)

nodev : Empêche l'usage des fichiers spéciaux (contenu du repertoire /dev)

umask=valeur : permet de fixer les droits sur les fichiers qui ne sont pas gérés par le système de fichiers, nativement. Comme dans le cas des partitions windows. La valeur est donnée en octal. 0 (ou 000) correspond à un accès total pour tous (chmod 777), 222 correspond à un accès en lecture et exécution pour tous (chmod 555). Vous l'aurez compris le chmod correspondant = 777 - umask.

gid=valeur : fixe le groupe auquel appartiendra la partition, s'utilise surtout pour /proc. Dans l'exemple ci-dessus, la partition /proc appartiendra au groupe de gid=5 (tty).

uid=valeur : fixe l'utilisateur auquel appartiendra la partition

Finissons avec les options annexes :

Le premier chiffre correspond au nombre de jours entre deux traitements du fichier par la commande dump. Ne fonctionne qu'avec l'ext2.

Ordre de tests des partitions (fsck). Si vous mettez 0, aucune vérification automatique n'est effectuée lors du démarrage. Les partitions situées sur un même disque seront testées d'une manière séquentielle mais si elles sont situées sur deux disques différents, c'est fait en parallèle. / doit toujours avoir le n°1.

Exemple de fichier /etc/fstab

```
/dev/hda1      none          swap          sw              0 0
/dev/hda5      /              ext3          defaults        0 0
/dev/cdrom     /mnt/cdrom    iso9660       noauto,users,ro 0 0
```

Grace à ce fichier, maintenant les deux commandes suivantes sont équivalentes.

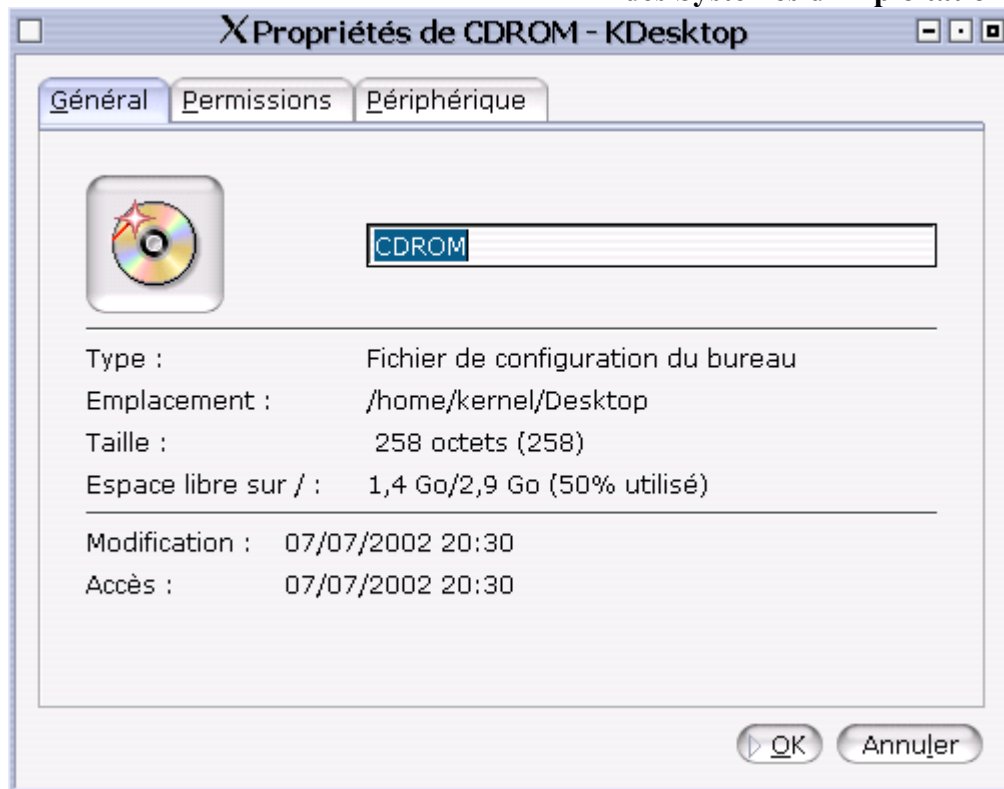
mount -t iso9660 -o noauto,users,ro /dev/cdrom /mnt/cdrom
mount /mnt/cdrom

no/auto	Monte automatiquement le périphérique
defaults	Active rw, suid, exec, auto, nouser
no/exec	Autorise l'exécution de binaires
no/suid	Autorise l'utilisation du suid et guid
ro	Lecture seule
rw	Lecture écriture
<i>nom d'utilisateur</i>	Autorise un utilisateur à monter et démonter un périphérique
no/users	Autorise tous les utilisateurs à monter ou démonter un périphérique

m	Aide
p	Affiche l'état du disque
n	Crée une nouvelle partition
d	Supprime une partition
t	Change le type de partition (82 : Linux Swap, 83 : Linux Native)
w	Sauver et quitter
q	Quitter sans sauver

Désormais pour monter votre partition, win98 par : mount /mnt/win98

Mount va lire fstab et monter win98 avec les options qui y sont spécifiées. Mais il est possible de faire tout ça sans passer par le terminal, depuis kde par exemple. Pour cela faites : **clic-droit -> Nouveau -> lecteur de CDROM:**



Général : permet de fixer le nom du raccourci, choisissez des noms explicite comme, graveur, dvdrom ...

Permissions : laissez par défaut

Périphériques : Dans la liste déroulante, choisissez /dev/cdrom pour un cdrom et cochez si nécessaire lecture seule.

Désormais plus besoin de terminal pour monter vos lecteurs amovibles, il suffit de **cliquer sur l'icône sur le bureau**. Pour démonter, toujours aussi simple : **clic-droit (sur l'icône) -> démonter**.

5.8 Gestion des quotas

5.8.1 Introduction

Avant de rentrer dans le vif du sujet, il m'apparaît essentiel de faire quelques rappels sur les systèmes de fichiers.

a- Définitions :

- **Système de fichiers** : organisation logique des données dans un disque ou une partition. C'est l'existence du système de fichiers qui vous permet de ranger vos fichiers dans des répertoires et sous-répertoires.
- **Formatage** : c'est l'opération qui consiste à créer un nouveau système de fichiers dans un disque ou une partition. Sous Linux, on utilise la commande générique `mkfs`, par exemple.
- **Racine d'un système de fichiers** : la racine correspond au répertoire d'attache du système de fichiers. Par exemple, /home est la racine du système de fichiers contenant les homes directories des utilisateurs. À quoi reconnaît-on qu'il s'agit de la racine ? À la présence du

répertoire lost+found. Il est créé lors de l'opération de formatage et est utilisé lors des réparations éventuelles du système de fichiers.

- **Bloc** : c'est l'unité d'espace du système de fichiers. Il peut varier d'un système à l'autre. Pour connaître la taille en octets du bloc :

```
$dumpe2fs /dev/hdc11 | grep -i 'block size'
dumpe2fs 1.27 (8-Mar-2002)
Block size:          40
```

où dumpe2fs est la commande qui permet d'afficher les informations relatives à un système de fichiers (attention, la quantité d'infos peut être énorme !) et /dev/hdc11 désigne le fichier spécial de la partition contenant le système de fichiers.

- **Inode** : chaque fichier dispose de ce qu'on appelle une inode. Il s'agit de la carte d'identité du fichier. Elle contient notamment sa taille, sa date de dernière modification, les emplacements du disque dur sur lesquels sont situés les données... et un numéro. Et oui nos fichiers ne sont en fait que des numéros. Chaque fichier a donc un numéro unique dans chaque système de fichiers. C'est ce numéro qui est utilisé par le système lors d'opération sur les fichiers. Le nom que vous attribuez à un fichier n'est utilisé que par vous. Pour obtenir ces numéros : ls -li

b- Qu'est-ce qu'un quota :

L'attribution de quotas dans un système de fichiers est un outil qui permet de maîtriser l'utilisation de l'espace disque. Les quotas consistent à fixer une limite d'espace pour un utilisateur ou un groupe d'utilisateurs. Pour la création de ces quotas, on définit 2 types de limites :

La limite douce (ou *soft limit*) : indique la quantité maximale d'espace qu'un utilisateur peut occuper sur le système de fichiers. Si cette limite est atteinte, l'utilisateur reçoit des messages d'avertissement quant au dépassement du quota qui lui a été attribué. Si son utilisation est combinée avec les *délais* (ou *grace period*), lorsque l'utilisateur continue à dépasser la soft limite après que se soit écoulé le délai de grâce, alors il se retrouve dans le même cas que dans l'atteinte d'une limite dure.

La limite dure (ou *hard limit*) définit une limite absolue pour l'utilisation de l'espace. L'utilisateur ne peut pas dépasser cette limite. Passée cette limite, l'écriture sur ce système de fichiers lui est interdite.

De plus ces limites sont exprimées en blocs et en inodes. On a vu que le bloc étant une unité d'espace. Les quotas exprimés en nombre de blocs représentent donc une limite d'espace à ne pas dépasser. En ce qui concerne les quotas exprimés en nombre d'inodes, ils représentent le nombre maximum de fichiers et répertoires que l'utilisateur pourra créer.

Pour mémoire, les *délais* (ou *grace period*) fixent une période de temps avant que la limite douce ne se transforme en limite dure. Elle est fixée dans les unités suivantes : second, minute, hour, day, week.

c- Ce qui est nécessaire pour activer les quotas

Généralement dans la plupart des distributions, les quotas sont utilisables d'office. Vous devez vérifier 2 choses pour pouvoir utiliser les quotas :

vous devez disposer des **outils de gestion des quotas** :

```
$rpm -qa | grep quota  
quota-3.06-5
```

la gestion des quotas doit être activée **dans le noyau** :

```
$grep -i quota /boot/config-2.4.18-14  
CONFIG_QUOTA=y
```

5.8.2 Configuration de la gestion des quotas

a. Configuration de /etc/fstab

Les quotas sont activés au démarrage grâce à la commande `quotaon` lancée par le script `/etc/rc.d/rc.sysinit`. Les quotas sont désactivés à l'arrêt du système par la commande `quotaoff`. Pour fixer les quotas sur un système de fichiers, il faut mettre à jour le fichier `/etc/fstab`. On va pour cela ajouter les options de montage pour le ou les systèmes de fichiers concernés. Deux options peuvent être utilisées (et combinées bien sûr) :

usrquota : active les quotas utilisateurs

grpquota : active les quotas groupes

Exemple :

```
/dev/hdc1 /home ext3 defaults,usrquota 1 1  
/dev/hdc2 /tmp ext3 defaults,grpquota 1 1
```

b. Création des structures nécessaires au fonctionnement des quotas

Un ou deux fichiers doivent être créés pour l'utilisation des quotas : `aquota.user` et `aquota.group`. C'est dans ces fichiers que l'on configurera les quotas attribués aux utilisateurs et/ou aux groupes. Ces fichiers doivent être créés à la racine des systèmes de fichiers qui comportent ces quotas.

Exemple :

```
$touch /home/aquota.user  
$touch /tmp/aquota.group
```

Attention : ne pas oublier de **modifier les droits** sur ces fichiers ! Ils doivent comporter les droits en écriture et lecture pour root uniquement.

Exemple :

```
$chmod 600 /home/aquota.user  
$chmod 600 /tmp/aquota.group
```

Remonter le ou les systèmes de fichiers concernés pour prendre en compte l'utilisation de quotas pour ce système de fichiers.

```
$mount -o remount /home  
$mount -o remount /tmp
```

Après création de ces fichiers, il faut initialiser la base des quotas en exécutant la commande suivante :

```
[#verif quotacheck -auvg].
```

Dans le cas contraire, la sanction est immédiate.

```
edquota: Quota file not found or has wrong format.  
No filesystems with quota detected.
```

Activer les quotas :

```
$quotaon -a
```

5.8.3 Attribution et vérification des quotas

a. Fixer des quotas

L'attribution des quotas se fait grâce à la commande `edquota`, utilisable quelque soit le type de quota (utilisateur ou groupe). La commande ouvre un éditeur (`vi` ou `emacs` selon le contenu de votre variable `EDITOR`), qui vous permet de modifier directement les fichiers `aquota.user` ou `aquota.group`.

Syntaxe : `edquota [-u user] [-g group] [-t]`

- u user définit les quotas pour un ou plusieurs utilisateurs
- g group définit les quotas pour un ou plusieurs groupes
- t définit les délais

Exemple :

```
$edquota -u ali
Disk quotas for user ali (uid 500):
Filesystem      blocks      soft      hard      inodes      soft      hard
/dev/hdc1         0        9000     10000         0       90000     10000
```

Le fichier se compose de 7 colonnes :

- Filesystem : système de fichiers concerné par les quotas
- blocks : nombre de blocs occupés par l'utilisateur dans le système de fichiers. Ici aucun fichier n'a encore été créé.
- soft : limite soft en nombre de blocs. Ici elle est fixée à 9 000 blocs soit environ 9 Mo
- hard : limite hard en nombre de blocs (environ 10 Mo)
- inodes : nombre d'inodes occupées par l'utilisateur dans le système de fichiers
- soft : limite soft en nombre d'inodes
- hard : limite hard en nombre d'inodes

On procédera de la même façon pour l'attribution de quotas à un groupe. (Ne tentez pas d'éditer directement ces fichiers; ils ne sont pas en format texte.)

b. Fixer un délai

On a vu également qu'on pouvait moduler le délai fixé entre le moment où l'utilisateur atteint la limite soft et celui où on va lui interdire toute occupation supplémentaire dans le système de fichiers. On va donc fixer la durée de ce délai. Elle sera la même quelque soit l'utilisateur et/ou le groupe.

Syntaxe : `edquota -t`

Exemple :

```
$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/hdc1	7days	7days

il suffit donc de remplacer les valeurs par vos valeurs dans l'unité qui vous convient : second, minute, hour, day, week.

5.8.4 Dépassement de quotas : que se passe-t-il ?

Une fois n'est pas coutume, on se place du côté utilisateur. Nous allons décrire les principaux cas de figure de dépassement de quotas et les messages envoyés à l'utilisateur.

exemple : un utilisateur dispose de 9Mo en limite douce et 10 Mo en limite dure. Son délai de grâce est de 7 minutes. Ci-dessous le contenu du système de fichiers faisant l'objet de ces quotas :

```
$ ls -l /home/ali
total 1842
-rw----- 1 root    root      7168 fév 28 23:50 aquota.user
-rw-r--r-- 1 ali     ali      1857516 mar  1 12:19 fic1
drwx----- 2 root    root      12288 nov 28 12:59 lost+found
```

Nous sommes largement en-dessous des quotas. Nous allons maintenant copier 4 fois le fichier fic1. Les 3 premières copies se passent bien et nous avons fic2, fic3 et fic4. Ci-dessous, la dernière copie

```
$ cp fic1 fic5
idel(22,10): warning, user block quota exceeded.
```

```
$ ls -l
total 9134
-rw----- 1 root    root      7168 fév 28 23:50 aquota.user
-rw-r--r-- 1 ali     ali      1857516 mar  1 12:19 fic1
-rw-r--r-- 1 ali     ali      1857516 mar  1 13:18 fic2
-rw-r--r-- 1 ali     ali      1857516 mar  1 13:18 fic3
-rw-r--r-- 1 ali     ali      1857516 mar  1 13:18 fic4
-rw-r--r-- 1 ali     ali      1857516 mar  1 13:18 fic5
drwx----- 2 root    root      12288 nov 28 12:59 lost+found
```

La limite douce est dépassée. L'utilisateur reçoit un message mais l'écriture est réalisée car nous n'avons pas dépassé la limite dure.

2 cas de figures peuvent alors se présenter si l'utilisateur ne contacte pas l'administrateur ou s'il ne libère pas de l'espace pour repasser en-dessous de la limite douce.

1er cas : l'utilisateur tente d'écrire dans le système de fichiers ce qui l'amène à dépasser la limite dure.

```
$ cp fic1 fic6
idel(22,10): write failed, user block limit reached.
cp: écriture de `fic6': Débordement du quota d'espace disque L'opération échoue.
Une partie du fichier seulement a été copiée. L'utilisateur ne pourra plus écrire dans le
système de fichiers.
```

2ème cas : l'utilisateur laisse s'écouler le délai de grâce de 7 minutes fixé par l'administrateur. Il tente alors de copier le contenu du fichier /etc/passwd par exemple. Le total de l'espace occupé reste toutefois inférieur à la limite dure. La sanction sera identique que dans le 1er cas. L'opération échoue.

```
$ cp /etc/passwd .  
ide1(22,10): write failed, user block quota exceeded too long.  
cp: écriture de `./passwd': Débordement du quota d'espace disque.  
L'opération a échoué comme en témoigne le listage ci-dessous :  
$ ls -l passwd  
-rw-r--r--  1 ali  ali      0 mar 1 14:48 passwd
```

De même si vous essayez d'écrire dans le fichier passwd, vous obtiendrez le message suivant dans votre éditeur au moment de l'enregistrement :

"passwd" erreur d'écriture (système de fichiers plein ?)

Appuyez sur ENTRÉE ou tapez une commande pour continuer
Il vous est impossible d'écrire.

5.8.5 Vérification et affichage des quotas

Les commandes suivantes vont vous permettre d'une part de vérifier les quotas affectés à chaque groupe et/ou utilisateur et éventuellement de synchroniser les informations nécessaires au système pour le suivi de ces quotas.

a. Edition des informations relatives aux quotas

La commande repquota permet d'afficher un résumé de l'utilisation des quotas et délais de grâce.

Syntaxe : repquota [-vug] -a | filesystem

- v : mode verbeux, affiche des infos supplémentaires
- u : affiche des informations sur les quotas utilisateurs
- g : affiche des informations sur les quotas groupes
- a : affiche des informations sur tous les systèmes de fichiers disposant de quotas
- filesystem** : affiche des informations sur les quotas du système de fichiers spécifié

Pour l'exemple, j'ai ajouté un utilisateur redha.

```
# repquota -avug  
*** Report for user quotas on device /dev/hdc10  
Block grace time: 00:07; Inode grace time: 00:07  


|       |    | Block limits |      |       |       | File limits |      |       |       |
|-------|----|--------------|------|-------|-------|-------------|------|-------|-------|
| User  |    | used         | soft | hard  | grace | used        | soft | hard  | grace |
| root  | -- | 19           | 0    | 0     |       | 2           | 0    | 0     |       |
| ali   | -- | 7293         | 9000 | 10000 |       | 5           | 9000 | 10000 |       |
| redha | +- | 9000         | 8000 | 9000  | 00:07 | 5           | 8000 | 9000  |       |
| +     | -- | 19           | 0    | 0     |       | 2           | 0    | 0     |       |

  
Statistics:  
Total blocks: 7  
Data blocks: 1  
Entries: 3  
Used average: 3,000000
```

On trouve ici les informations relatives au quota imposé aux utilisateurs. On trouvera autant de lignes que d'utilisateurs, groupes et systèmes de fichiers concernés.

Sont rappelés les quotas fixés en nombre de blocs et d'inodes. On trouve également le nombre de blocs et le nombre d'inodes utilisés. Quand un horodatage apparaît dans la colonne `grace`, comme par exemple pour `redha`, cela signifie que l'utilisateur (ou le groupe) a dépassé la limite douce. Le délai de grâce est donc décompté.

Vous pouvez également utiliser la commande `quota` suivie du nom d'un utilisateur ou d'un groupe. Là encore vous obtiendrez toutes les informations relatives aux quotas et à l'utilisation de l'espace attribué.

Exemple : pour obtenir les informations liées aux quotas concernant Ali :

```
$quota aali
Disk quotas for user ali (uid 500):
Filesystem blocks quota limit grace files quota limit grace
/dev/hdc10 7293 9000 10000 5 9000 10000
```

b. Vérifications et synchronisation des fichiers de quotas

Il peut arriver que les fichiers de quotas deviennent incohérents. La gestion de ceux-ci devient alors impossible. D'autre part, lorsque vous ajoutez un nouvel utilisateur ou un nouveau groupe à l'aide de la commande `edquota`, il faut là encore synchroniser les fichiers pour la prise en compte de ces nouvelles informations.

Syntaxe : `quotacheck [-vug] -a | filesystem`

-v : mode verbeux, affiche des infos supplémentaires

-u : vérifie uniquement les fichiers de quotas utilisateurs

-g : vérifie uniquement les fichiers de quotas groupes

-a : vérifie les fichiers de quotas de tous les systèmes de fichiers en disposant

filesystem : vérifie les fichiers de quotas du système de fichiers spécifié

Exemple : vérifier tous les fichiers de quotas, quelque soit le système de fichiers concerné

```
$quotaoff -a
$quotacheck -auvg
quotacheck: Scanning /dev/hdc10 [/home/anne/quota] done
quotacheck: Checked 2 directories and 10 files
```