

Système de Gestion de Fichiers

Problématique : Comment sont organisées les données sur un disque ? (Les systèmes de fichiers) et comment manipuler un fichier sous linux ? (Utilisation du Bash)

Public visé : Classe de 1^{ère} NSI.

Un système d'exploitation doit être capable de gérer et organiser les données sous forme de fichiers et de répertoires afin de les rendre accessible à l'utilisateur.

L'accès aux fichiers se fait soit via une interface graphique (par exemple windows) ou texte via une console de commande.

Voyons l'exemple ci-dessous :

Nom	Modifié le	Type	Taille
dossier1	25/08/2019 16:29	Dossier de fichiers	
dossier2	25/08/2019 16:29	Dossier de fichiers	
fichierA.txt	25/08/2019 16:30	Document texte	1 Ko
FichierB.txt	25/08/2019 16:31	Document texte	1 Ko
memo_bash.pdf	09/07/2019 21:41	Adobe Acrobat D...	191 Ko

Nous avons deux dossiers et 3 fichiers, des informations diverses sur la date de création, la taille, la nature du fichier.

Nous pouvons avoir l'équivalent en ligne de commande :

```
C:\Users\Users\DU\COURS SYSTEME FICHIER\test\dossier1>cd ..
C:\Users\Users\DU\COURS SYSTEME FICHIER\test>dir
Le volume dans le lecteur C n'a pas de nom.
Le numéro de série du volume est 823C-18AC

Répertoire de C:\Users\Users\DU\COURS SYSTEME FICHIER\test

25/08/2019  16:38    <DIR>          .
25/08/2019  16:38    <DIR>          ..
25/08/2019  16:29    <DIR>          dossier1
25/08/2019  16:29    <DIR>          dossier2
25/08/2019  16:30             16 fichierA.txt
25/08/2019  16:31             71 FichierB.txt
09/07/2019  21:41        195 370 memo_bash.pdf
               3 fichier(s)          195 457 octets
               4 Rép(s)  1 676 651 081 728 octets libres
```

Sous linux nous pouvons avoir quelque chose du genre :

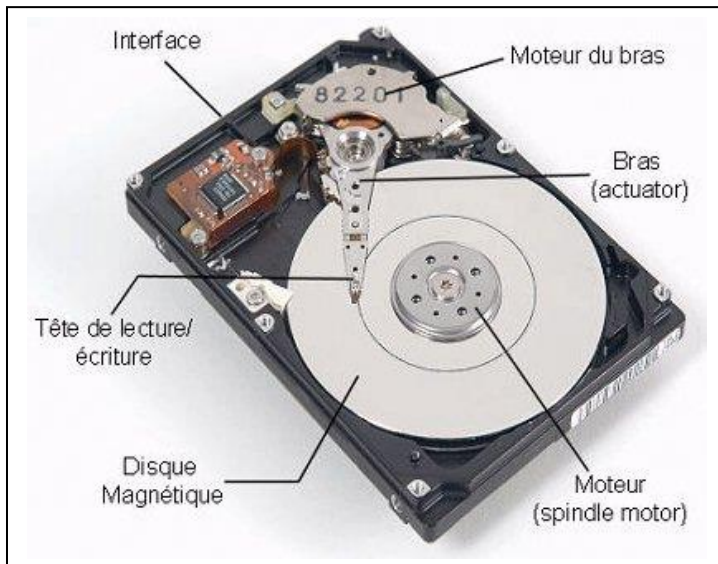
```
[root@localhost ~]# ls -l
total 28
-rw-----  1 1000    root      71 Aug 25 17:12 FichierB.txt
drwxr-xr-x  3 root    root     163 Aug 21  2011 dos
drwxr-xr-x  2 root    root      37 Aug 25 17:14 dossier1
drwxr-xr-x  2 root    root      37 Aug 25 17:14 dossier2
-rw-----  1 1000    root     16 Aug 25 16:46 fichierA.txt
-rw-r--r--  1 root    root    242 Jul 15  2017 hello.c
drwxr-xr-x  2 root    root      37 Aug 25 16:44 toto
[root@localhost ~]#
```

Note : outil pour simuler bash sous linux en ligne :

<https://bellard.org/jslinux/>

<https://bellard.org/jslinux/vm.html?url=https://bellard.org/jslinux/buildroot-x86.cfg>

Ce qu'il faut comprendre c'est que ce que nous voyons n'est qu'une représentation des données sur le disque. En effet, sur le disque dur nous avons qu'une suite de 0 et de 1 (données binaires) et c'est le système d'exploitation (linux, windows, macos ...) qui va devoir gérer et afficher ces données de façon compréhensible pour l'utilisateur, et fournir les commandes de base:



Disque dur mécanique



Disque SSD

Formatage et Partitionnement

Partitionnement : consiste à « cloisonner » le disque. Il permet la cohabitation de plusieurs systèmes d'exploitation sur le même disque. L'information est stockée sur le premier secteur du disque, le MBR (Master Boot Record) .

Formatage : Avant qu'un système de fichiers puisse créer et gérer des fichiers sur une unité de stockage, son unité doit être **formatée** selon les spécificités du système de fichiers. Le formatage inspecte les secteurs, efface les données et crée le répertoire racine du système de fichiers

Le concept de fichiers

Un fichier est l'unité de stockage logique mise à la disposition des utilisateurs pour l'enregistrement de leurs données : c'est l'unité d'allocation. Le Système d'Exploitation établit la correspondance entre le fichier et le système binaire utilisé lors du stockage de manière transparente pour les utilisateurs. Dans un fichier on peut écrire du texte, des images, des calculs, des programmes...

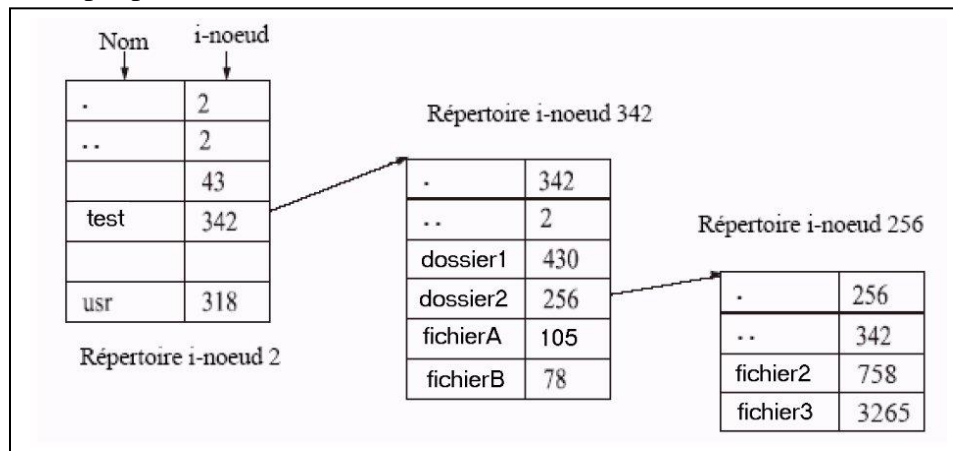
Afin de différencier les fichiers entre eux, chaque fichier a un ensemble d'attributs qui le décrit. Parmi ceux-ci on retrouve : le nom, l'extension, la date et l'heure de sa création ou de sa dernière modification, la taille, la protection. Certains de ces attributs sont indiqués par l'utilisateur, d'autres sont complétés par le système d'exploitation.

Le concept de répertoire

Un répertoire est une entité créée pour l'organisation des fichiers.

Du point de vue du système de gestion de fichier, *un répertoire est un fichier* qui dispose d'une structure logique : il est considéré comme un tableau qui contient une entrée par fichier.

Exemple pour linux :



Un répertoire est donc une table d'adressage qui va pointer vers d'autres répertoires ou des fichiers.

Les différentes catégories de fichiers sous linux :

- **fichiers normaux (-)** : fichiers normaux : * *texte* : courrier, sources des programmes, scripts, configuration ; * *exécutables* : programmes en code binaire
- **fichiers répertoires (d)** : ce sont des fichiers conteneurs qui contiennent des références à d'autres fichiers. ils permettent d'organiser les fichiers par catégories

```
[root@localhost ~]# ls -l
total 8
drwxr-xr-x  3 root    root    163 Aug 21  2011 dos
-rw-r--r--  1 root    root    242 Jul 15  2017 hello.c
[root@localhost ~]#
```

• **fichiers spéciaux** : situés dans /dev, ce sont les points d'accès préparés par le système aux périphériques. Le montage va réaliser une correspondance de ces fichiers spéciaux vers leur répertoire "point de montage". Par exemple, le fichier /dev/hda permet l'accès et le chargement du 1er disque IDE

• **fichiers liens symboliques (l)** : ce sont des fichiers qui ne contiennent qu'une référence (un pointeur) à un autre fichier. Cela permet d'utiliser un même fichier sous plusieurs noms sans avoir à le dupliquer sur le disque.

La commande `ls -l` pour un lien donne :

`lrwxrwxrwx 1 root root 145 Aug 21 01:58 Mail -> ../../bin/mail*`

Sous un système UNIX, un fichier quel que soit son type est identifié par un numéro appelé numéro d'inode (i-noeud). Un répertoire n'est qu'un fichier, identifié aussi par un inode, contenant une liste d'inode représentant chacun un fichier.

Pour connaître le numéro d'inode d'un fichier, on utilise la commande :
ls -li mon-fichier

```
[root@localhost ~]# ls -li hello.c
669 hello.c
[root@localhost ~]# ls -li dos
650 asm-1.9          648 debug.com      646 hpigot.s
647 asm.com          649 hpigot.com
[root@localhost ~]#
```

inode

Commandes pour parcourir et lister les répertoires

ls → commande générale d'accès aux infos des fichiers du répertoire courant (ls, ls -l, ls -a)
cd [chemin] → le chemin peut être absolu ou relatif
cd .. → remonter un niveau (vers le répertoire parent)
pwd → donne le nom complet du répertoire courant
mkdir rep → pour créer un sous-répertoire du répertoire courant
rmdir rep → pour supprimer un sous-répertoire vide
mv répertoire répertoire-d'accueil/ → déplacement d'un répertoire
mv répertoire nouveau-nom → Changement de nom d'un répertoire

Commandes pour gérer les fichiers :

touch mon-fichier → création d'un fichier vide,
more mon-fichier → visualisation d'un fichier page à page,
rm mon-fichier → suppression d'un fichier,

mv mon-fichier répertoire_accueil → déplacement d'un fichier,
mv mon-fichier nouveau-nom → changement de nom d'un fichier,
cp nom-fichier répertoire-accueil/autre-nom → copie de fichier,
ln -s rapport.txt /home/etudiant/rapport-lien-s.txt
ln rapport.txt /home/etudiant/rapport-lien-dur.txt

Droits d'accès au fichier : La commande chmod

La commande
"chmod valeur fichier"
permet de modifier les
droits d'accès .

Valeur	Valeur numérique	Explication
rw	600	Le propriétaire peut lire et écrire. Le groupe et les autres ne peuvent rien faire avec le fichier.
-rw-r-r-	644	Le propriétaire peut lire et écrire, le groupe et les autres peuvent lire.
-rw-rw-rw-	666	Le propriétaire, le groupe et les autres peuvent lire et écrire.
-rwx-	700	Le propriétaire peut lire, écrire et exécuter. Le groupe et les autres ne peuvent rien faire avec le fichier.
-rwx-x-x	711	Le propriétaire peut lire, écrire et exécuter. Le groupe et les autres peuvent exécuter.
-rwxr-xr-x	755	Le propriétaire peut lire, écrire et exécuter. Le groupe et les autres peuvent lire et exécuter.
-rwxrwxrwx	777	Le propriétaire, le groupe et autres peuvent lire, écrire et exécuter.

-rwxrw-r--

Read, write, and execute permissions for all other users
Read, write and execute permissions for members of the group owning the file.
Read, write and execute permissions for the owner of the file.
File type. "-" indicates a regular file. A "d" indicates a directory.

TD Manipulation des commandes bash relatives au système de fichier :

Exécuter le simulateur bash linux suivant :

<https://bellard.org/jslinux/vm.html?url=https://bellard.org/jslinux/buildroot-x86.cfg>

Répondez aux questions ci-dessous en indiquant systématiquement la commande utilisée Pour chacune des commandes et quand cela est nécessaire, vous vérifierez leur action en tapant `ls -l`. **Attention à ne pas quitter la page** du simulateur afin de ne pas perdre vos données en cours.

- Créez deux dossiers appelés dossier1 et dossier2
- Renommez dossier2 en dossier3
- Importer le fichier memo_bash.pdf et copiez le dans dossier3
- Copier le fichier hello.c dans dossier3
- afficher les inodes du dossier3
- Aller dans dossier2 et créez un fichier vide : toto.txt
- Appliquer une autorisation d'écriture, lecture et exécution pour le propriétaire et le groupe et uniquement lecture pour tous les utilisateurs
- Placez vous dans dossier2 et **déplacer** le fichier toto.txt dans le dossier3
- Supprimer le dossier2
- créer un fichier toto2.txt
- Appliquer à toto2.txt la commande `chmod 640` et indiquer à quoi elle correspond
- Donner le chemin complet du répertoire courant
- Donner les numéros d'inode du dossier 3 et du fichier toto2.txt