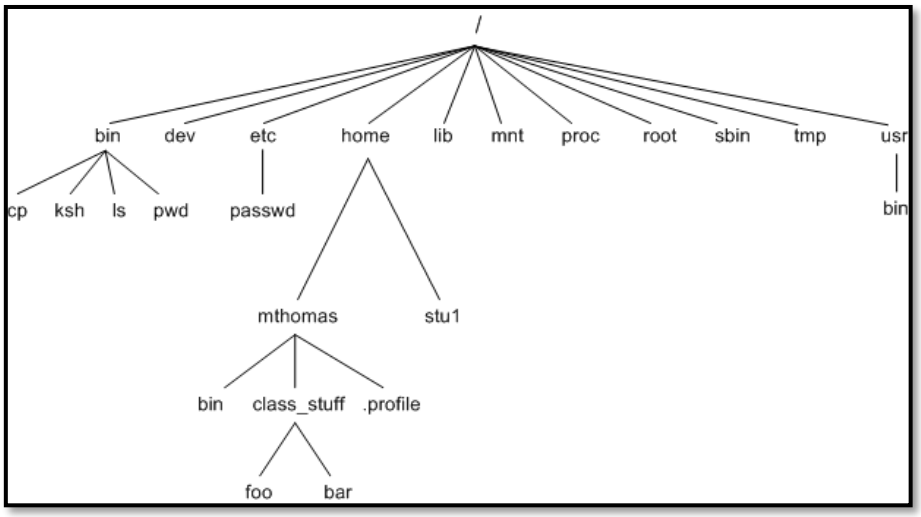
Gestion des fichier UNIX

A l'attention de M. Van Kerm



Henallux - Potier Clément

2022 – 2023

1. Qu'est ce qui est considéré comme un fichier dans Unix ?

Il y a en réalité plusieurs types de fichier, à savoir :

- Les fichiers normaux :
  - Ce sont les fichiers classiques que tout le monde utilise, les .txt, .pdf, etc
- Les répertoires :
  - Qui sont des fichiers contenant d'autres fichiers, plus communément appelés des répertoires. Ceux-ci sont des nœuds dans un arbre.
- Les fichiers de lien symboliques
  - Ce sont des fichiers qui mènent directement vers un autre fichier. Cette mécanique est utilisée notamment pour créer des raccourcis.
- Les fichiers spéciaux :
  - Qui sont les fichiers dédiés aux périphériques.

2. Représentation logique

Dans UNIX, les fichiers sont représentés par :

- Un nom
- Un emplacement
- Les droits d'accès au fichier
- Quelques informations complémentaires comme la date de création, …

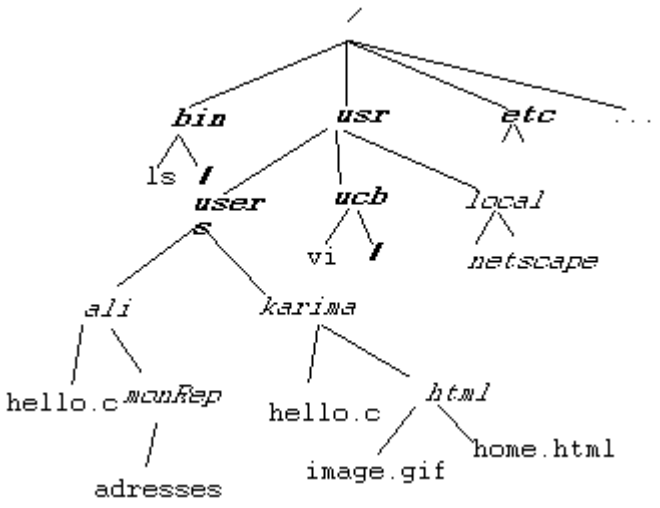
La représentation la plus adaptée est celle d'un arbre à une seule racine.

Chaque répertoire représente un nœud et chaque fichier est une feuille. La particularité c'est que tout descend forcément de la même racine, le « / ».

Par exemple, dans l'image, pour accéder à adresse, il faut suivre le chemin d'accès suivant :

/usr/user/s/ali/monRep/adresses

Si l'utilisateur peut créer les répertoires comme bon lui semble, le système a besoin d'une structure prédéfinie.



Par exemple, « /bin » contient les binaires qui font fonctionner le système, « /dev » gère les fichiers concernant les périphériques, « /lib » quant à lui concerne les librairies nécessaires.

3. Les i-nodes et la représentation physique des fichiers, qu'est ce que c'est, comment ça fonctionne ?

Dans UNIX, les informations des fichiers sont stockées dans une structure associée au fichier (via un numéro unique) qu'on appelle i-node ou nœud d'information.

Pour afficher ces ID, on utilise la commande « ls -i ».

```
pi@raspberrypi:~/Desktop $ ls -i
269768 bot_horaire  388533 discord  270856 OS  269780 test.c
```

Figure 1 - Exemple de la commande "ls -i"

Dans les informations stockées dans l'i-node, on retrouve entre autres :

- La taille du document
- Les informations de création / modification
- Les pointeurs vers les blocs constituant le fichier en question

Chaque i-node est reprit dans la table des i-node. Ceux-ci sont limitées. On peut également connaitre le nombre total d'i-node disponible et utilisés avec la commande « df -i [*disk_path*] »

```
pi@raspberrypi:/dev $ df -i /dev/mmcblk0
Sys. de fichiers Inœuds IUtil. ILibre IUti% Monté sur
devtmpfs          88620    443  88177    1% /dev
```

Figure 2 - Exemple de la commande "df -i [disk]"

Les fichiers sont enregistrés sur le disque sous forme de bloc. Les blocs font soit 512, 1024 ou 2048 octets (bytes). A chaque fichier correspond donc une liste de bloc éparpillés sur le disque. Il y a 2 méthodes pour conserver les fichiers via ces blocs : La liste chainée et la table d'index.

La liste chainée :

Comme son nom l'indique, il s'agit d'une liste chainée. C'est-à-dire que chaque bloc va contenir, en plus du stockage de l'information du fichier, un lien vers le bloc suivant. Cette méthode pose les mêmes problèmes qu'en programmation, c'est-à-dire qu'on doit parcourir tous les blocs pour accéder à une partie précise du fichier.

La table d'index :

C'est la méthode qui utilise les i-node vu précédemment. En effet, en associant la structure d'un i-node à un fichier, on peut également stocker une table d'index indiquant l'emplacement physique des blocs sur le disque. La seule particularité, c'est qu'on ne retient que les 10 premiers blocs. Si le fichier est plus gros et qu'il a recourt à plus de 10 blocs, on utilise des indirections.

Une indirection est l'adressage d'un bloc lui-même composé de bloc.

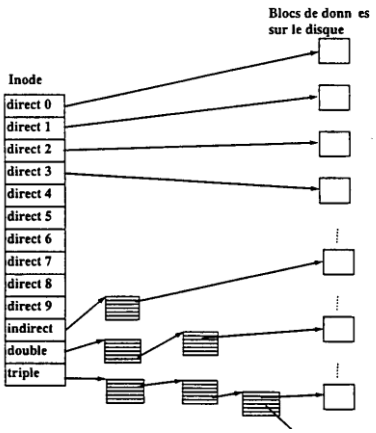


Figure 3 - Description de l'indirection

4. La protection des fichiers et des répertoires

Comme vous pouvez vous en douter, les dossiers concernant le système ne peuvent être accessible à n'importe qui sous peine de rendre le système inutilisable. Il faut donc bloquer l'accès aux personnes non autorisées.

Vous auriez aussi envie de rendre un fichier uniquement lisible et non modifiable pour un autre utilisateur du système.

Tout cela est possible grâce aux protections des fichiers.

Pour savoir ce que l'on protège, il faut d'abord savoir ce qu'on peut autoriser. Prenons connaissance des différents types de permissions :

- Lecture (r)
- Récriture (w)
- Exécutable (x)

On peut définir également un champ d'action. Déterminer les personnes affectées d'une permission particulière. Il y a 3 groupes de « portée » :

- L'utilisateur lui-même
- Le groupe duquel il fait parti
- Le reste des utilisateurs

Pour afficher les permissions, il faut exécuter la commande « ls -l »

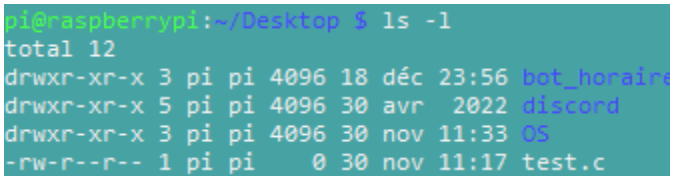


Figure 4 - Commande ls- l

Comment décrypter les sigles de permissions à gauche ?

Commençons par le premier caractère. « d » pour les 3 premier et « - » pour le dernier. Il s'agit en réalité du type de fichier. Voici une liste exhaustive de mise en relation :

-	Fichier classique
d	Répertoire
l	Lien symbolique
c	Périphérique de type caractère
b	Périphérique de type bloc
p	Pipe (fifo)
s	socket

Ensuite, on peut découper les 9 caractères restant en 3 groupes de 3 caractérisant respectivement les 3 groupes de portée (propriétaire – groupe – autre)

Pour chaque groupe, 3 caractères désignant : lecture (r) – écriture(w) – exécution(x).

Prenons l'une de mes lignes en exemple et décomposons là pour en reconnaitre tous les aspects :

drwxr-xr-x.

d	Indique c'est un dossier	
rwx	Indique les permissions pour le propriétaire du groupe	Accès en tout
r-x	Indique les permissions pour les membres du groupe	Accès uniquement en lecture et en exécution
r-x	Indique les permissions pour n'importe qui	Accès uniquement en lecture et en exécution

Heureusement, ces restrictions ne sont pas figées à jamais. Nous devons utiliser la commande « chmod » suivit du groupe visée (la portée), un signe d'action et les permissions visées.

Encore une fois, une image vaut mieux que 1000 mots :

```
pi@raspberrypi:~/Desktop $ chmod o+wx test.c
```

Figure 5 - Commande chmod

La commande chmod demande donc de quel groupe il faut modifier les permissions :

- Le propriétaire : (u)
- Le groupe : (g)
- Les autres : (o)
- Tout le monde : (a)

Ensuite, il faut renseigner un cigle d'addition ou de soustraction en fonction que l'on veuille ajouter ou supprimer des permissions.

Suivi de quoi on donne les permissions qu'on désire ajouter/supprimer. On utilise le même code de caractère qu'au-dessus (r,w,x).

Et on termine en précisant le fichier/répertoire concerné.

Une fois la commande précédente lancée, voici le résultat :

```
pi@raspberrypi:~/Desktop $ ls -l
total 12
drwxr-xr-x 3 pi pi 4096 18 déc 23:56 bot_horaire
drwxr-xr-x 5 pi pi 4096 30 avr  2022 discord
drwxr-xr-x 3 pi pi 4096 30 nov 11:33 OS
-rw-r--rwx 1 pi pi    0 30 nov 11:17 test.c
```

Figure 6 - Permissions modifiées

5. Les appels système Posix et OFT

Il existe un tas d'appels système proposé par la norme Posix. Concernant les fichiers, on aimerait par exemple un créer un, l'ouvrir, le modifier, le fermer, l'effacer, et plus.

Voici une liste non exhaustive des appels système Posix :

open()	Pour ouvrir un fichier
close()	Pour fermer un fichier
read()	Pour lire un fichier
write()	Pour modifier un fichier

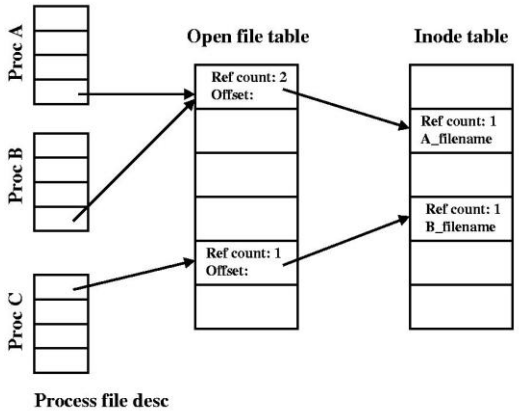
5.1 L'ouverture d'un fichier

Si la fonction d'ouverture de fichier 'open' réussit, elle renvoi un descripteur de fichier. Lequel viendra prendre place dans la Table des descripteur (OFT en anglais).

L'Open File Table liste donc tous les fichiers actuellement ouvert.

A noter qu'un fichier n'est fermé que lorsque c'est demandé explicitement.

Ce système permet notamment à ce que 2 processus fasse appel au même fichier.



5.2 mount et umont

Les appels système mount et umont permettent respectivement de monter et de démonter des systèmes de fichiers.

Pour faire simple, cela signifie que nous pouvons assigner un nouveau nœuds de l'arbre de Linux au répertoire visé.

Exemple :

- ➔ Je branche un disque dur sur mon ordinateur.
- ➔ Il est actuellement repéré dans les fichiers spécifiques aux périphériques mentionné au début de ce document, soit : /dev/sda1 (Pas forcément « sda1 », c'est à titre d'exemple).
- ➔ Je peux créer un nœud dans mon arborescence pour y accéder directement depuis mon /usr. Il suffit l'utiliser la commande mont *<source> <destination>*

➔ `pi@raspberrypi:~/Desktop $ mount /dev/sda1 /usr/disk1`

Figure 7 - Commande mount

Désormais, j'ai accès à mon disque dur via le chemin /usr/disk1 et l'ancien chemin d'accès devient inutilisable jusqu'au démontage.

Comme précisé précédemment, la structure logique dans UNIX est un arbre constitué de nœud et de feuilles. La commande mount ne fait que rajouter un nœud à l'arbre si tant est qu'on ait les droits.

6. Bibliographie

1. **Udacity** (23 février 2015). « Inode Structure », sur le site *Youtube.com*, consulté le 10 décembre 2022. https://www.youtube.com/watch?v=tMVj22EWg6A
2. **Anonyme**, (date inconnu), « Système de gestion de fichiers Windows NT vs Unix », sur le site *igm.univ-mlv.fr*, consulté le 10 décembre 2022. http://igm.univ-mlv.fr/~dr/XPOSE/NTFSvsUFS/
3. **Moths-art** (11 septembre 2022). « Gérer les droits d'accès (propriétés et permissions) des fichiers et répertoires », sur le site *doc.ubuntu-fr.org*, consulté le 11 décembre 2022. https://doc.ubuntu-fr.org/permissions
4. **Anonyme** (19 juin 2013). « La commande chmod », sur le site *dms.umontreal.cs*, consulté le 11 décembre 2022. https://dms.umontreal.ca/wiki/index.php/La_commande_chmod
5. **TechGuiders** (12 août 2019). « Open file Table Tutorial – 3 », sur le site *Youtube.com*, consulté le 14 décembre 2022. https://www.youtube.com/watch?v=rQaIbFHT_8k
6. « Unix system file tables » (7 Janvier 2013), sur le site *stackoverflow.com*, consulté le 14 décembre 2022. https://stackoverflow.com/questions/14189944/unix-system-file-tables
7. **Anonyme** (date inconnue). « Chapitre 11 : Système de fichiers », sur le site *cours.polumtl.ca*, consulté le 10 décembre 2022. https://cours.polymtl.ca/inf2610/documentation/notes/chap11.pdf
8. **Anonyme** (6 juin 2008). « Mount », sur le site *manpagesfr.free.fr*, consulté le 22 décembre 2022. http://manpagesfr.free.fr/man/man8/mount.8.html

Table des matières