

Présentation de la mémoire de masse et du système de gestion de fichier UNIX



Christopher MILAZZO

2019 – 2020

Licence STS – informatique



TABLE DES MATIERES

Mémoire de masse -----	page 3
1 - Présentation -----	page 3
2 – Les types de mémoire de masse -----	page 3
3 – Disque dur mécanique -----	page 4
3.1 - La morphologie d'un disque dur mécanique -----	page 4
3.2 - Principe de fonctionnement -----	page 5
3.3 - Les caractéristiques -----	page 5
3.4 - Les performances -----	page 6
4 – La mémoire réinscriptible à oxyde de silicium -----	page 7
Système de gestion de fichier UNIX -----	page 8
1 – Le but -----	page 8
2 – Représentation du système de fichier -----	page 8
3 – Organisation de l'espace disque -----	page 9
4 – Stockage des fichiers -----	page 12
5 – La structure des répertoires -----	page 13
6 – Les fichiers partagés -----	page 14
7 – La fiabilité du système de fichier -----	page 15
8 – La performance du système de fichier -----	page 16
Références -----	page 18

Mémoire de masse

1 - Présentation

La mémoire de masse est la partie de l'ordinateur ou le support physique qui permet de sauvegarder des données de manière persistante même lorsque l'ordinateur est éteint. Elle est dite "non volatile".

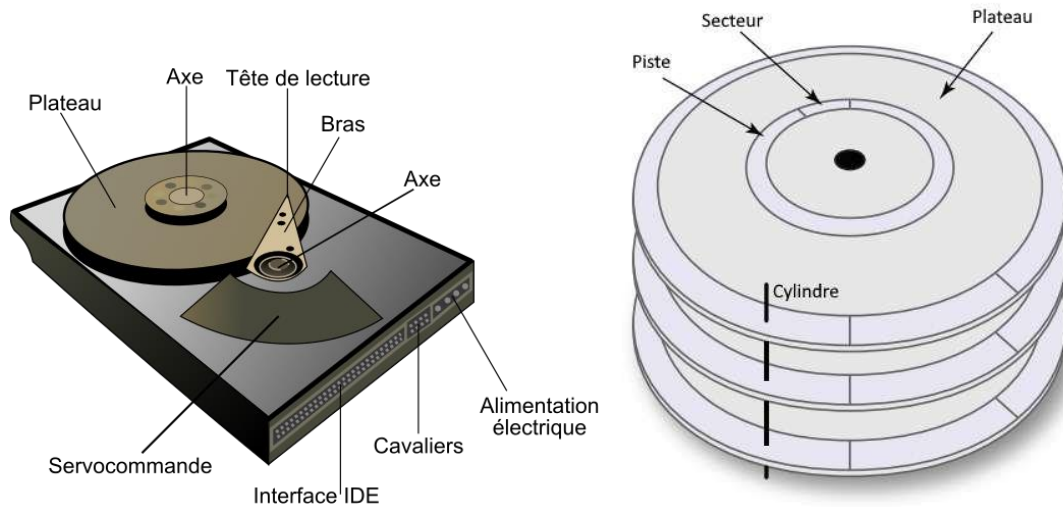
2 - Les types de mémoire de masse

Technologies désuètes
<ul style="list-style-type: none">• Mémoire à tores magnétiques• Carte perforée¹• Ruban perforé• Cassette audio• Tambour magnétique• Disquette.
Technologies en usage
<ul style="list-style-type: none">• Bande magnétique• Disque dure• SSD• Disque optique (CD, DVD, Blu-ray)• Disque magnéto-optique• Mémoire flash.
Technologie expérimentale
<ul style="list-style-type: none">• Mémoire holographique• Memristor

Dans ce rapport, je parlerai du disque dur mécanique, puis j'aborderai succinctement la technologie à l'oxyde de silicium.

3- Disque dur mécanique

3.1 - La morphologie d'un disque dur mécanique



Le plateau : C'est le support sur lequel les données sont inscrites. Un disque dur est constitué de plusieurs plateaux placés les uns en dessous des autres.

Les têtes de lecture : Les têtes de lectures permettent d'écrire ou de lire de la donnée sur le disque dur. Un disque dur est constitué de plusieurs têtes de lecture (2 par plateaux). Ainsi chaque plateau contiendra des données sur le recto puis sur le verso.

Piste : Les données sont organisées en cercles concentriques appelés « pistes ». Les têtes commencent à inscrire des données à la périphérie du disque (piste 0), puis avancent vers le centre.

Secteur : Ces pistes sont découpées en blocs de taille fixe qu'on appelle des secteurs. Ces secteurs sont numérotés. Le début de chaque secteur est identifié par un préambule, qui permet de le délimiter sur une piste, suivi des données du secteur proprement dit, puis de bits de correction d'erreur.

Préambule	Données	ECC
-----------	---------	-----

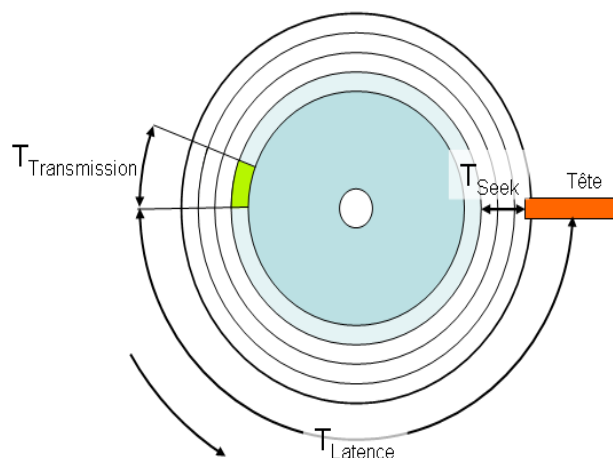
Contenu d'un secteur

Cylindre : C'est l'ensemble des données qui se situent sur des plateaux différents mais qui sont sur la même piste et sont alignées comme indiqué par le trait noir sur le schéma au-dessus.

3.2 - Principe de fonctionnement

Dans un disque dur mécanique, les têtes de lecture-écriture sont dites « inductives » ; C'est à dire qu'elles sont capables de changer la polarité sur la surface des plateaux. Lors de la lecture, les pistes d'un disque vont avoir 2 états possibles sur de très petites zones : chargé négativement ou positivement. Ce qui se traduira par 0 ou 1 (ou vice-versa), ce qui correspondra à un bit qui est l'unité élémentaire en informatique.

3.3 - Les caractéristique



Un disque dur à une certaine vitesse de rotation, ce qui induit 3 facteurs principaux :

- Le temps de transmission, qui représente le temps que va prendre la tête de lecture pour lire ou écrire sur un secteur.
- Le temps de latence (ou délai rotationnel), qui représente le temps que va prendre la tête à trouver les données après avoir trouvé la piste
- Le temps de positionnement (ou seek

time), qui représente le temps que va mettre la tête de lecture pour se déplacer entre les deux pistes les plus éloignées (full-stroke).

La vitesse de rotation, les temps de transmission, de latence et de positionnement ne sont pas les seules caractéristiques d'un disque dur, toutefois, elles sont celles qui vont le plus influencer les performances.

3.4 - Les performances

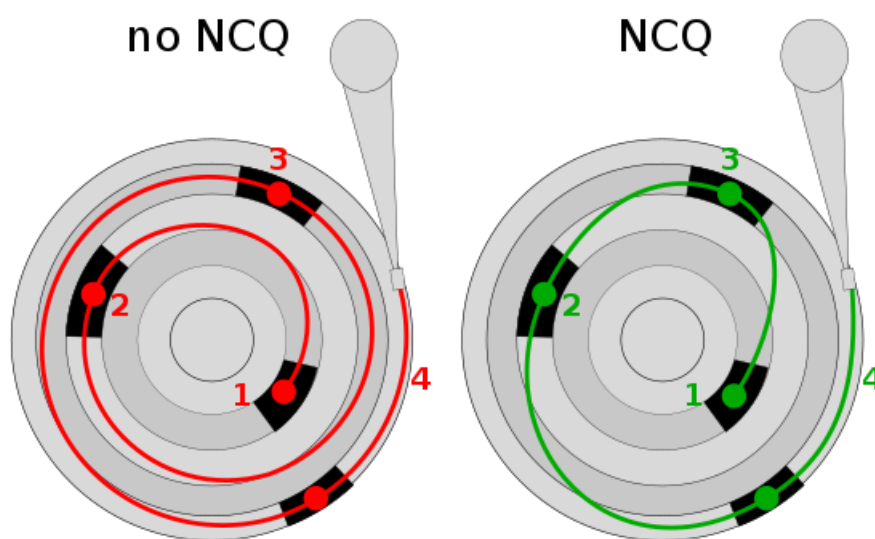
Lorsqu'un disque dur écrit des données alors, la requête suivante devra attendre que le disque dur finisse avant de pouvoir être exécutée. De la même façon que lorsque le disque dur lit de la donnée, il devra parfois attendre que le processeur soit de nouveau disponible pour recevoir ces données.

Pour améliorer les performances, certains disques-durs sont équipés de mémoire cache permettant de stocker la donnée à écrire ou la donnée lue afin de pouvoir continuer à répondre aux requêtes suivantes sans délai. Cette mémoire cache fonctionne comme une liste ordonnancée suivant la politique FIFO.

Une autre avancée technologique en matière de disque-dur est le "native Command and Queuing" (NCQ). Ces disques-durs sont également connus sous le nom de "disque-dur S-ATA".

Le principe de fonctionnement :

Comme vu plus haut, dans les caractéristiques d'un disque-dur, il existe des délais liés aux contraintes mécaniques (rotation des plateaux, positionnement des têtes de lecture, etc.) qui sont très chronophages. La technique du NCQ consiste à éviter les allées retours du bras causé par le fait de devoir lire les données dans un ordre séquentiel précis.



Sur le schéma ci-dessus, on voit la méthode de lecture classique à gauche, contrainte de lire les blocs de données dans un certain ordre. Alors que à droite la méthode NCQ permet d'économiser une rotation.

4- La mémoire réinscriptible à oxyde de silicium

La technologie est basée sur un canal qui se crée dans la couche d'oxyde de silicium lorsqu'une tension est appliquée à ses bornes. Le courant qui passe dans cette couche « écarte » en effet les molécules d'oxygène et crée ainsi un canal large de quelques 5 micromètres (μm), soit 5 millièmes de millimètre. Toutefois, la puce réalisée embarque des points mémoires disposant de canaux larges de 10 à 100 μm afin d'obtenir différents motifs de test.

Suivant la tension appliquée, ce canal peut être « fermé » ou bien recréé, ce qui équivaut à un bit pouvant prendre les valeurs « 1 » ou « 0 ».

Le dispositif s'avère robuste avec un rapport marche arrêt de 10 000 pour 1, ce qui devrait lui permettre de fonctionner durant 10 ans de manière fiable.

Système de gestion de fichier d'UNIX

1 - Le but

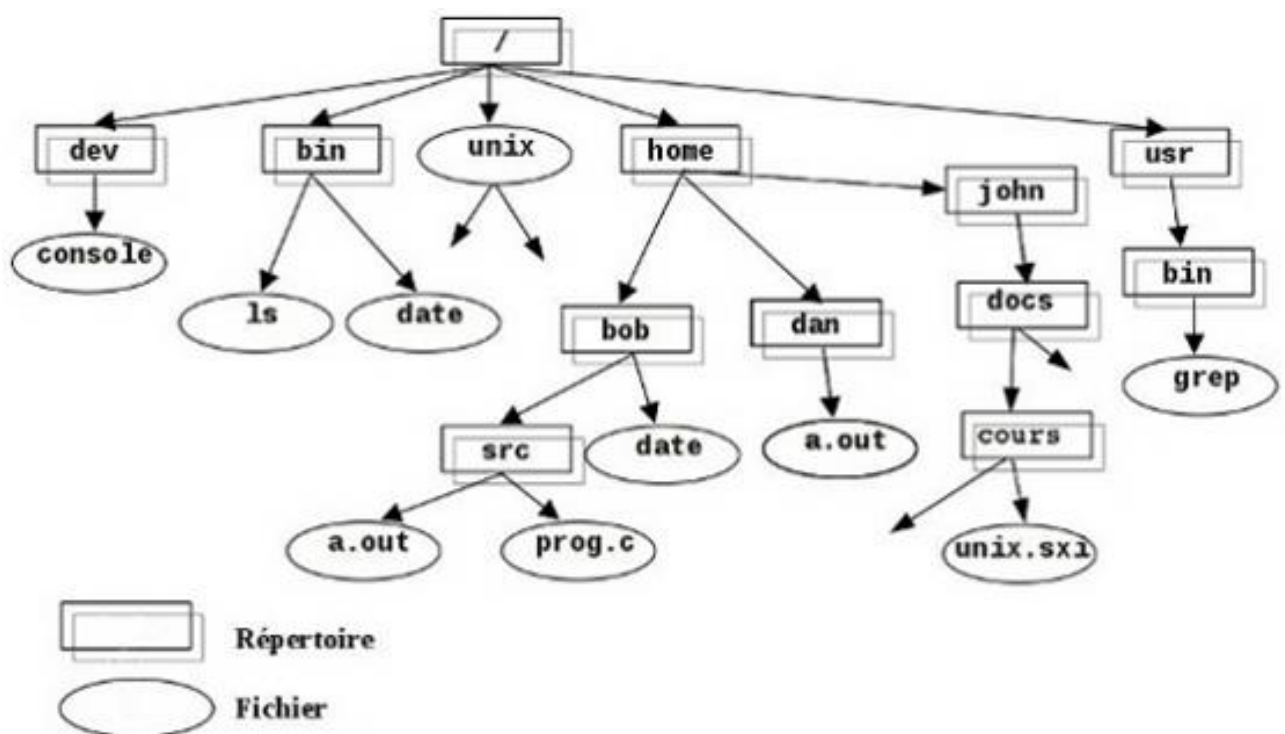
Le système de gestion de fichier (SGF) est la partie visible d'un système d'exploitation. Il permet de structurer la donnée et de l'organiser de manière logique pour les utilisateurs. C'est aussi lui qui va gérer les points d'accès aux différents périphériques branché à un ordinateur.

D'un point de vue du système, le SGF assure également la sécurité et la fiabilité des données, et tend à améliorer les performances en matière d'écriture et de lecture.

Dans la suite de ce document, je traite plus particulièrement du système de gestion de fichier d'UNIX.

2 - Représentation du système de fichier

Le système de fichier d'UNIX est représenté et organisé sous la forme d'une arborescence constituée de répertoires et de fichiers. Les répertoires contiennent plusieurs répertoires et/ou fichiers.



Les périphériques sont accessibles par le biais de fichiers spéciaux.

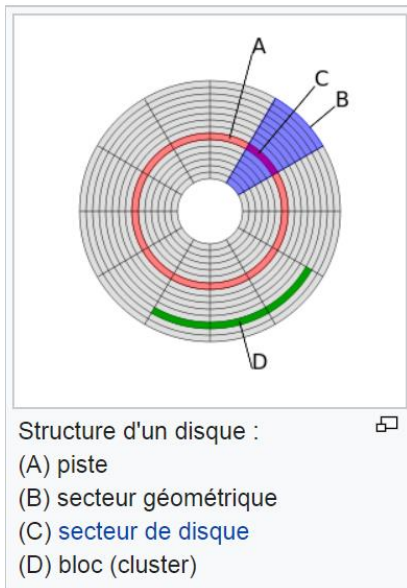
Les fichiers spéciaux de type blocs permettent d'accéder à un dispositif de stockage par blocs adressables (comme les disques-dur mécaniques par exemple).

Les fichiers spéciaux de type caractère sont pour les périphériques utilisant les flux de chaîne de caractère (comme les terminaux par exemple).

3 - Organisation de l'espace disque

On l'a vu plus haut, les disques-durs mécaniques ont des pistes qui sont sectorisées. Chaque secteur est une suite de bit de quantité constante.

Le SGF va lui aussi découper les données en plusieurs blocs. Un bloc est également appelé **clusters**. La taille d'un cluster est forcément un multiple de la taille d'un secteur. La taille d'un bloc ne peut pas être plus petit qu'un secteur. Un bloc est l'unité atomique la plus petite du SGF.



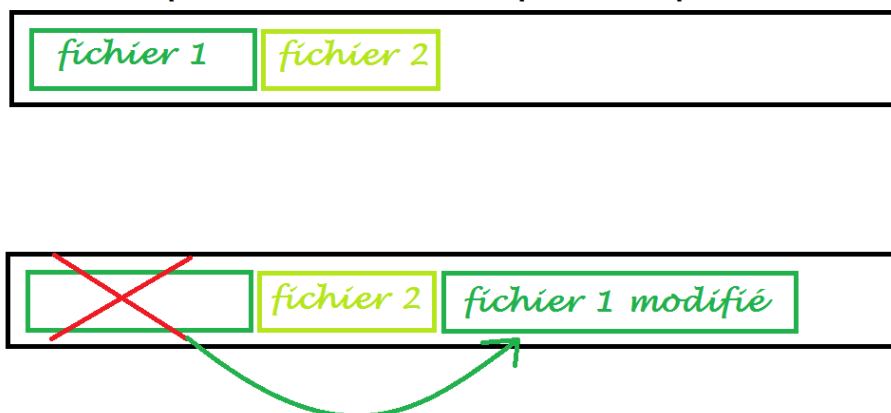
Ici le bloc représenté en vert fait la taille de 3 secteurs.

Par exemple : si un secteur fait 512 octets et un bloc fait 1536 octets alors, un bloc de données sera écrit sur 3 secteurs consécutifs et sera considéré comme un espace unique et non divisible.

La raison pour laquelle on ne se contente pas d'écrire les données d'un fichier tout d'un seul bloc en remplissant le disque dur petit à petit en partant du début et en suivant les pistes dans l'ordre, c'est que ce serait beaucoup trop contraignant.

En effet, les fichiers sont amenés à être modifiés très régulièrement, donc à modifier de taille. Supposons que je crée un fichier 1 puis un fichier 2, ainsi écrit l'un après l'autre sur le disque dur. Si je venais à modifier le fichier 1 en y ajoutant des données, il faudrait alors déplacer le bloc de données de celui-ci puisque qu'il n'aurait plus la place à cause du fichier 2. Cette manipulation deviendrait de plus en plus compliquée et chronophage au fur et à mesure que le disque se remplirait.

représentation de l'espace disque



Une question se pose alors : Quelle serait la taille idéale d'un bloc ?

Si les blocs sont trop petits, cela risque de demander beaucoup d'aller retours au bras du disque dur pour les fichiers un peu gros.

A l'inverse, si les blocs sont trop grands, on retombe sur la problématique citée plus haut. De plus on perdrait beaucoup d'espace de stockage car pour tous les petits fichiers qui n'utiliseraient que 1% du bloc, on perdrait 99% de potentiel de stockage par bloc.

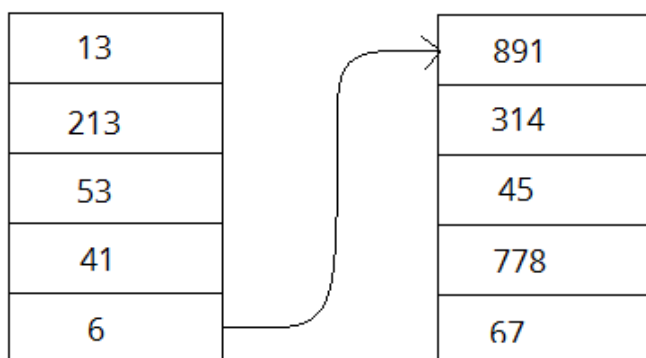
Ainsi la taille d'un bloc doit être calculé intelligemment afin de garantir le moins de perte de place et de performance que possible.

Une fois la granularité des blocs choisie, il faut trouver le moyen de référencé les blocs libres (en d'autres termes, les blocs qui peuvent être alloués)

Pour cela deux techniques existent :

Dans un ou plusieurs blocs eux même libres, on garde le numéro de tous les blocs libres. Ces blocs forment une liste chaînée.

Dans un ou plusieurs blocs, on tient une sorte de matrice de bit où les bit à 1 représenteraient un bloc libre.



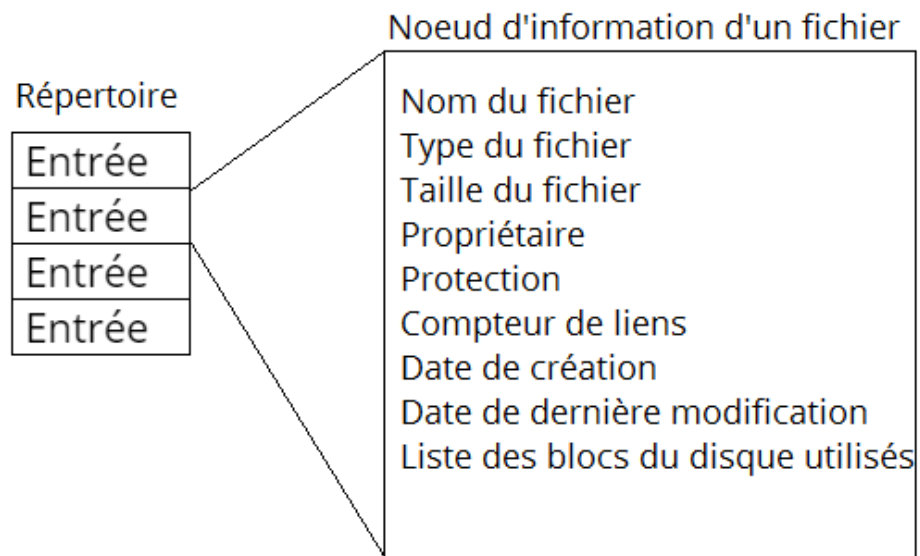
Un bloc libre contenant une liste de numéros de blocs libres

Le bloc n°6 qui est libre et qui contient une liste de numéros de blocs libres

0011011101001110
1010111001110011
0100110001101011
1101110001011010
0111010001111101

La table de bit de bloc libre. Un bit informe si le bloc est libre ou non

Les répertoires contiennent des entrées qui contiennent les numéros de blocs où sont stockés les nœuds d'information (i-node) des fichiers et leurs noms :



Le seul point d'entrée de l'arborescence est la racine (/). La racine n'étant pas référencée, son i-node est situé à un endroit fixe et connu du système d'exploitation (en général vers le début) du disque où est installé le SGF.

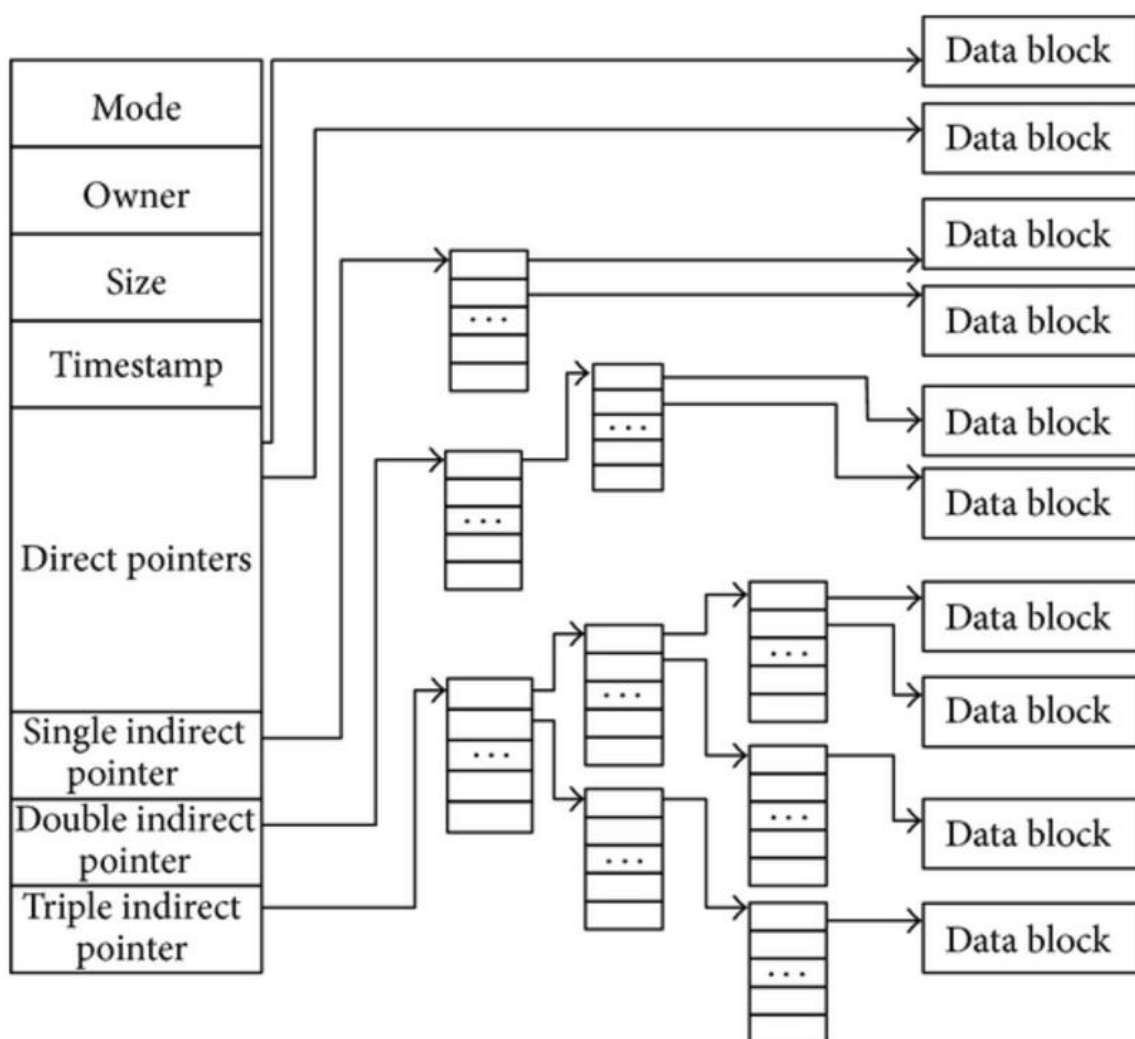
4 - Stockage des fichiers

Les fichiers sont stockés dans multiple blocs disséminés de manière éparses sur tout le disque. Afin de connaître les blocs appartenant au fichier, le SGF maintient à jour une petite liste (généralement 10) de numéro de blocs. Cette liste est située dans l'i-node du fichier.

Si le fichier est trop gros pour tenir dans 10 blocs, alors on utilise un pointeur qui pointerait sur une liste de blocs, appelé le **pointeur à simple indirection**.

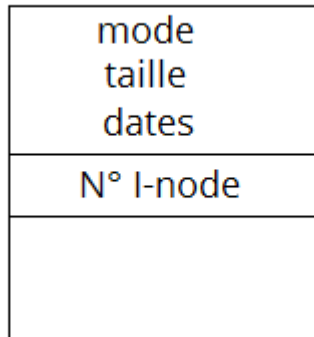
Si ce pointeur supplémentaire n'est toujours pas suffisant, il existe le **pointeur à double indirection** et le **pointeur à triple indirection**.

Par conséquent, un fichier ne pourra pas dépasser un certain nombre de blocs, puisqu'il n'existe pas d'autre pointeur permettant de référencer les blocs appartenant à ce fichier.

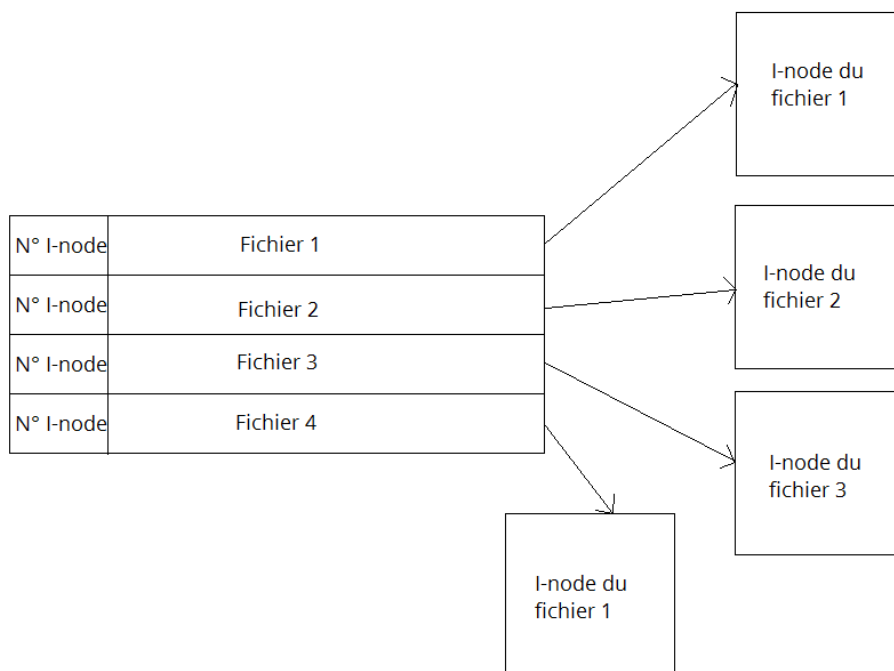


5 - La structure des répertoires

Tout comme les fichiers, les répertoires ont eux aussi un nœud d'information (i-node) décrit comme suit :



L'i-node du répertoire pointe sur un bloc contenant les entrées illustré ci-dessous :

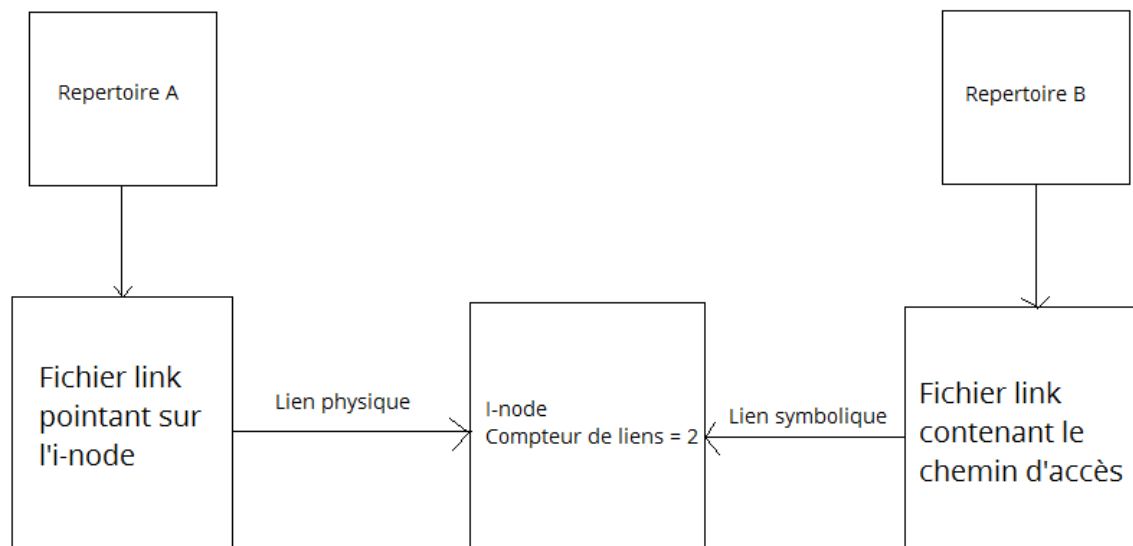


6 - Les fichiers partagés

Il peut être utile de pouvoir travailler sur des fichiers partagés avec d'autres utilisateurs. Ces fichiers ont besoin d'apparaître à plusieurs endroits différents de l'arborescence de fichier. Pour cela le SGF d'UNIX propose les fichiers de type LINK. Ces fichiers pointent directement sur l'i-node, ainsi le fichier est accessible depuis plusieurs endroits, il a plusieurs points d'entrée.

Lorsque l'on crée un lien, l'attribut "Nombre de liens" de l'i-node est incrémenté de 1. À l'inverse, lorsque l'on supprime l'un des points d'entrée, l'attribut est décrémenté de 1. Si à ce moment-là il tombe à 0 alors l'i-node est supprimé puis lui et tous les blocs qui lui sont associés sont réinscrits à la liste des blocs libres.

Il existe un deuxième type de lien appelé lien symbolique. C'est un simple fichier qui contient l'adresse vers le fichier sur lequel il pointe. L'inconvénient c'est que si le fichier pointé est supprimé, le lien lui reste en place mais pointerait sur un fichier qui n'existe plus.



7 - La fiabilité du système de fichier

La fiabilité et la cohérence du système de gestion de fichier est extrêmement importante car sans ça la perte de données importante pourrait subvenir.

Des blocs endommagés (problème à la sortie d'usine, ou suite à une usure ou une chute) peuvent être à l'origine de perte de données. Pour prévenir cela, il existe la solution de sauvegarde (backup) des données avec par exemple deux disques-durs qui se synchronisent l'un l'autre régulièrement.

Le système d'exploitation mais également en place une solution préventive qui consiste à détecter au démarrage les blocs endommagés et les retire de la liste (ou de la table) des blocs libres.

Il y a également des utilitaires qui peuvent vérifier la cohérence du système de fichier en parcourant la liste des blocs libres. Chaque fois qu'un bloc se trouve dans cette liste, on incrémente son compteur. On fait une seconde liste en comptant combien de fois un bloc est référencé par un pointeur.

Une fois ces deux listes obtenues on a 4 cas possibles :

N° de bloc	0	1	2	3	4	Le système de fichier est cohérent, car un bloc est soit libre soit utilisé.
Blocs libres	1	0	0	0	1	
Blocs utilisés	0	1	1	1	0	

N° de bloc	0	1	2	3	4	Le système de fichier n'est pas cohérent, il y a le bloc n° 1 qui est manquant. Ce dernier n'est ni utilisé, ni considéré comme libre. Cette anomalie n'est pas grave car la seule incidence est la perte de place en mémoire. Pour y remédier il suffit de remettre ce bloc dans la liste des blocs libres.
Blocs libres	1	0	0	0	1	
Blocs utilisés	0	0	1	1	0	

N° de bloc	0	1	2	3	4	Le système de fichier n'est pas cohérent car le bloc n°1 est dupliqué dans la liste des blocs libres. Encore une fois cette anomalie n'est pas grave car il suffit de reconstituer cette dernière faisant apparaître le bloc 1 qu'une seule fois.
Blocs libres	1	1	0	0	1	
Blocs utilisés	0	0	1	1	0	

N° de bloc	0	1	2	3	4	Le système de fichier n'est pas cohérent. C'est le pire des scénarios, ici le bloc 1 est utilisé par 2 fichiers. On ne peut pas connaître à quel fichier il appartient. La seule solution est de faire en sorte est de faire remonter l'information à l'utilisateur pour qu'il définisse lequel des fichiers doit garder la possession de ce bloc.
Blocs libres	1	0	0	0	1	
Blocs utilisés	0	1	1	1	0	

8 - La performance du système de fichier

L'accès à la mémoire centrale est bien plus rapide que l'accès au disque-dur (environ 100 000 fois plus rapide).

Afin d'éviter les accès au disque-dur, le système d'exploitation utilise une antémémoire (mémoire cache) qui est stockée en mémoire centrale. Ainsi un fichier actuellement ouvert en écriture ou en lecture se trouve chargé en cache et ainsi on met de côté la problématique liée aux performances du disque-dur.

Néanmoins il se pose le problème de la limite de la mémoire centrale. Quand celle-ci est pleine et que le fichier ne peut plus être chargé entièrement. Il est alors nécessaire de vider l'antémémoire en retirant un bloc tout en le sauvegardant sur le disque-dur si celui-ci a été modifié depuis sa mise en cache. Cela ressemble à la pagination et il est tout à fait possible d'utiliser les algorithmes de pagination tels que le FIFO ou le LRU.

Toutefois l'algorithme doit prendre en compte deux questionnement concernant les blocs : Ce bloc va-t-il être réutilisé rapidement ? Ce bloc est-il critique pour la cohérence du système de fichier ?

Malgré ces mesures, il peut arriver que certains blocs restent dans l'antémémoire relativement longtemps. En effet, quelqu'un qui travail sur l'ordinateur plusieurs heures, il va sans doute sauvegarder son travail régulièrement au cas où. Sauf qu'il y a de fortes chances que la plupart des données soit enregistrées uniquement dans l'antémémoire. Si une panne du système survient, la majorité de son travail, qui aurait dû être sauvegardé, serait perdu !

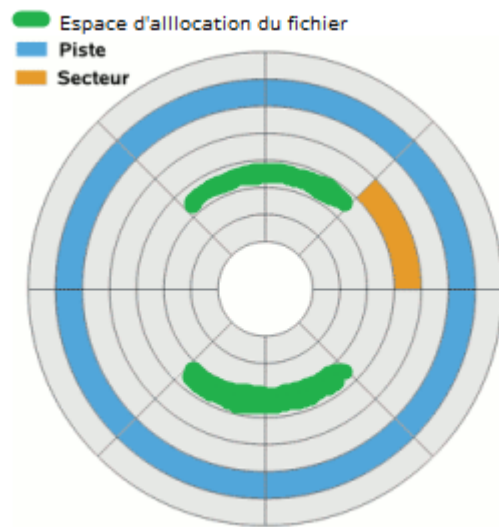
Afin de limiter cela, le système d'exploitation effectuera régulièrement une sauvegarde complète des blocs de l'antémémoire en effectuant l'appel système SYNC toutes les 30 secondes, ce qui limitera les pertes.

L'antémémoire n'est pas la seule façon d'améliorer les performances. Une autre technique consiste à limiter le déplacement du bras en regroupant les blocs susceptibles d'être adressés en séquence dans les mêmes cylindres.

On peut également prendre en compte la vitesse de rotation du disque et le temps que va mettre le système à réclamer le bloc suivant. Ainsi, les groupes de cylindre peuvent s'entrecroiser de manière à toujours être le plus proche possible de la tête de lecture lorsque le système souhaitera y accéder.

Même pour les fichiers de petite taille le système doit au minimum effectuer deux accès au disque. Le premier pour l'i-node et le second pour le bloc de données. Afin de réduire le temps de repositionnement moyen du bras, le système va grouper deux cylindres consécutifs afin de former un groupe de cylindres. Chaque groupe contiendra un l-node et le premier bloc de données ou 2 blocs de données (de préférence adressé en séquence).

Voici un petit schéma représentatif de l'adressage d'un fichier sur le disque :



Références

Disque dur magnétique :

<https://www.futura-sciences.com/tech/dossiers/informatique-stockage-donnees-informatiques-105/page/3/>

https://fr.wikibooks.org/wiki/Fonctionnement_d%27un_ordinateur/Les_disques_durs

Liste des types de mémoire de masse existantes :

https://fr.wikipedia.org/wiki/M%C3%A9moire_de_masse

La mémoire à l'oxyde de silicium :

<https://www.silicon.fr/la-memoire-re-inscriptible-a-oxyde-de-silicium-se-concretise-87794.html>

Le système gestion de fichier d'UNIX :

Le livre "Les systèmes d'exploitation" de Andrew TANENBAUM