

# Introduction et définition d'un fichier

## Définition d'un fichier

Un **fichier** est une collection nommée d'information en relation. Pour l'utilisateur·ice c'est un moyen de conserver des informations.

Le contenu d'un fichier est défini par son type (et donc sa structure), le programme qui l'a créé et les informations placées par l'utilisateur·ice.

Il existe plusieurs médias qui peuvent être utilisés pour stocker des fichiers de manière "définitive". La qualité de ce dernier est importante, par exemple le stockage à long terme sur les disques optique est difficile à garantir.

Le système d'exploitations propose, grâce aux fichiers un moyen d'utiliser l'espace de stockage. Le fichier est une abstraction nécessaire qui permet la représentation informatique de concept existants, et cela, de manière uniforme et simple d'accès.

Un fichier a plusieurs **attributs** tel que, le nom, l'identifiant, le type, les permissions, la localisation, la taille, la date de création et le propriétaire.

## Opérations sur les fichiers

Les opérations sur les fichiers sont,

- La **création** qui alloue de l'espace de stockage et stocke les informations concernant le fichier.
- La **lecture** qui est un appel système qui permet de lire le contenu d'un fichier et de stocker le résultat en mémoire.
- L'**écriture** qui est un autre appel système qui permet d'écrire dans un fichier des informations présente en mémoire.
- Le **positionnement** qui est un déplacement du pointeur de position centrale (pour pouvoir lire ou écrire à un certain endroit).

- La **suppression** qui correspond à supprimer le fichier ou une partie de celui-ci

# Ouverture d'un fichier

L'**ouverture** d'un fichier est l'opération indiquant au système d'exploitation que le programme souhaite accéder à un fichier. Lors de cette ouverture le système mémorise les informations sur le fichier, vérifie si l'accès est autorisé et initialise les structures internes et place le pointeur de position au début.

Une fois que le traitement est fini, le fichier est **fermé**, ce qui va supprimer les structures internes.

Ce système d'ouverture et de fermeture soulage fortement le système d'exploitation, car sans ça il faudrait répéter toutes les procédures d'ouverture et fermeture à chaque opération sur le fichier, tandis qu'ici, on le fait qu'une fois.

Il est aussi bon de noter que plusieurs utilisateurs doivent pouvoir ouvrir les , il faut donc mettre une structure efficace en mémoire pour éviter les informations en double.

# Informations à retenir sur les fichiers utilisés

Les informations à retenir sur le fichier sont, le mode d'ouverture (lecture seule, lecture-écriture, écriture seule, etc), l'emplacement vers le premier bloc du fichier, la position courante, et les permissions d'accès.

# Type de fichier

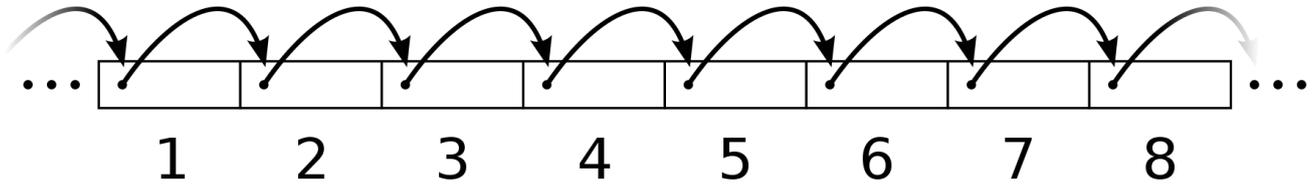
Pour traiter le fichier correctement, il faut connaître sa structure interne, et pour cela, il faut connaître son type.

Le type est identifié par l'extension (sous Windows), par un Uniform Text Identifier (sous Mac OS X), par un "nombre magique" qui est la valeur des premiers octets (sous Unix) et avec des "attributs étendus" sous OS/2.

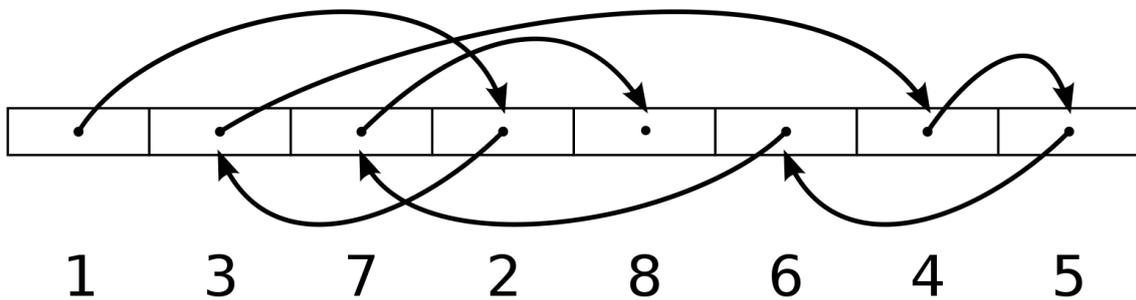
# Méthode d'accès des fichiers

La méthode d'accès aux fichiers dépend du support physique, ainsi une bande magnétique aura un accès **séquentiel** alors qu'une clé USB ou un disque dur aura un accès **aléatoire**.

# Accès séquentiel



# Accès direct



L'accès séquentiel correspond à un enregistrement en séquence, ainsi pour pouvoir lire le 10<sup>e</sup> élément, il faut d'abord passer sur les neuf premiers.

L'autre type d'accès est l'accès **aléatoire** ou **direct** qui correspond à des enregistrements de taille fixe, ainsi, on peut calculer directement l'adresse de l'élément auquel on veut aller sans avoir besoin de lire les précédents.

Il est aussi possible d'avoir un **accès séquentiel indexé** qui va créer un index permettant à la fois un accès séquentiel et un accès direct aux fichiers.

---

Revision #2

Created 4 January 2024 09:50:00 by SnowCode

Updated 6 January 2024 19:13:15 by SnowCode