

# Données et modélisation

## === Données numériques

Les données numériques sont une des composantes essentielles des programmes informatiques.



- Il s'agit par définition des *informations qui doivent être conservées entre deux exécutions*.
- Avec l'augmentation des capacités de stockage et leur mise en réseau, les quantités de données conservées ont considérablement augmenté au cours des dernières décennies.

La conservation des données repose principalement sur la **mémoire secondaire**,



- définissant la manière dont les données sont physiquement stockées sur le support,
- autrement dit la méthode de **rangement** de la série de mesures :
  - fichiers,
  - répertoires,
  - index,
  - etc...

### Persistence des données



- Contrairement à la mémoire primaire (comme la RAM) rapide et volatile (utilisée pour exécuter les programmes en cours), la mémoire secondaire conserve les informations même en l'absence d'alimentation électrique.
- Elle est généralement plus lente que la mémoire primaire, mais offre une capacité de stockage beaucoup plus importante.
- Les exemples courants de mémoire secondaire incluent les disques durs (HDD), les disques SSD, les clés USB, et les supports optiques comme les CD ou DVD.

## Fichiers et répertoires

La mémoire secondaire n'est pas directement accessible (les programmes n'ont pas la possibilité d'aller écrire directement sur le disque). Le système d'exploitation assure ainsi l'indépendance du programme par rapport à l'emplacement des données, au travers d'instructions d'entrée/sortie spécialisées.

Pour que les programmes puissent écrire sur le disque, on introduit des objets intermédiaires : les fichiers. Un fichier est caractérisé par (nom, emplacement (volume, arborescence), droit d'accès, taille,...). Il s'agit d'une entité logique. Tout programme utilisant un fichier passe par le système

d'exploitation qui, à partir des informations, détermine la localisation des données sur le support.

### A retenir :



- Un fichier est une référence vers un ou plusieurs blocs de données, enregistrés sur un support physique.
- Un fichier est caractérisé par son descripteur, constitué de son nom, son chemin d'accès, ses droits d'accès (lecture/écriture/exécution) selon les utilisateurs, sa position sur le support, sa taille, etc...

- La gestion des fichiers est une des fonctions essentielles des systèmes d'exploitation :
  - possibilité de traiter et conserver des quantités importantes de données
  - possibilité de partager des données entre plusieurs programmes.

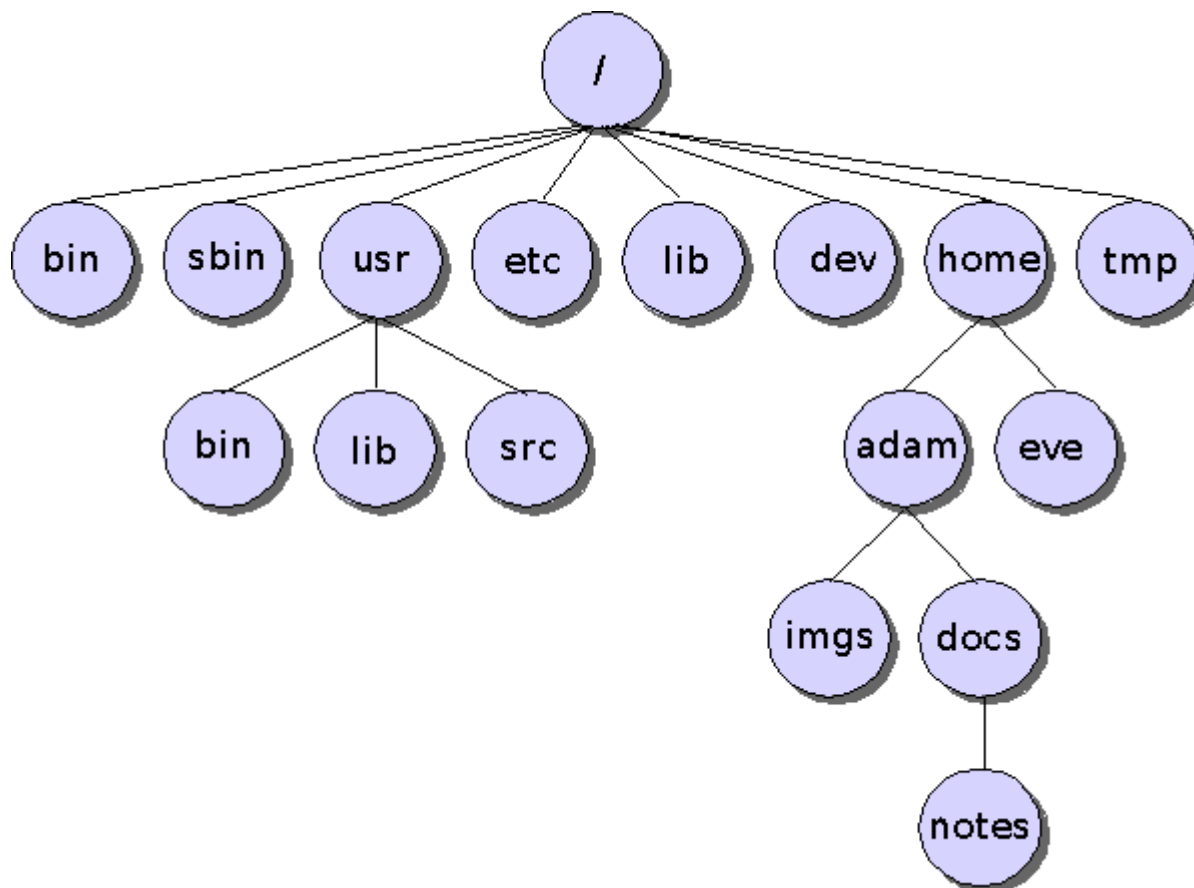
### Opérations de base :



- *Ouverture* : initialisation d'un **flux** en lecture ou en écriture
- *Lecture* : consultation des lignes l'une après l'autre (à partir de la première ligne), dans l'ordre où elles ont été écrites sur le support
- *Ecriture* : ajout de nouvelles données à la suite ou en remplacement des données existantes

### Organisation hiérarchique :

- Lorsque le nombre de fichiers est élevé, les descripteurs de fichiers sont classés dans plusieurs répertoires, organisés sous une forme arborescente.
- Le répertoire de plus bas niveau hiérarchique est appelé racine → chemin d'accès (path)



### Consultation des données : lecture d'un fichier (Read)

Méthode traditionnelle pour le traitement de fichiers de petite taille. La consultation des données nécessite d'ouvrir une "communication" entre le programme et les fichiers. Ce canal de communication permet de recevoir un flux de données.

Pour établir la communication, il faut connaître : le "chemin d'accès" aux données (disque dur local) l'"adresse" des données (lorsqu'il s'agit de données stockées sur un serveur distant)

L'opération d'ouverture de fichier initialise un descripteur de fichier, qui sert à désigner (dans le programme) le fichier sur lequel on travaille, et d'accéder au contenu du flux.

Ouverture simple:

#### Python



```
f = open('/chemin/vers/mon/fichier.dat', 'r')
```

Le deuxième argument représente le mode d'ouverture, ici 'r' représente une ouverture en mode lecture.

## Ouverture avec test :

Il est important de vérifier que cette opération d'ouverture s'effectue correctement avant de poursuivre le programme (nombreuses possibilités d'erreur : fichier effacé, erreur de nom, pas de droits de lecture,...). On utilise une instruction de test spécifique pour vérifier que l'ouverture du fichier s'est correctement effectuée, de type `try...catch...` (essaye ... sinon ...) permettant de prévoir une action de secours lorsqu'une opération "risquée" échoue.

### Python



```
try :  
    f = open('/chemin/vers/mon/fichier.dat', 'r')  
except IOError:  
    print "Erreur d'ouverture!"
```

Lorsque l'opération d'ouverture est réalisée avec succès, le flux de données devient accessible en lecture (les premières lignes du fichier sont chargées en mémoire et une tête de lecture se positionne sur le premier caractère de la première ligne). Il ne reste plus qu'à lire les données.

La consultation des données s'effectue séquentiellement à l'aide de l'opérateur de lecture `readline`. Chaque appel à cet opérateur charge les nouvelles données et déplace la tête de lecture vers les données suivantes. Cette opération peut être effectuée plusieurs fois jusqu'à atteindre la fin de fichier.

From:

<https://wiki.centrale-med.fr/informatique/> - **WiKi informatique**

Permanent link:

[https://wiki.centrale-med.fr/informatique/tc\\_info:2024\\_cm\\_modeles?rev=1734554705](https://wiki.centrale-med.fr/informatique/tc_info:2024_cm_modeles?rev=1734554705)

Last update: **2024/12/18 21:45**

