

# Les bases de l'Informatique

## Système de fichiers

**Thierry Vaira**

BTS SN

v1.0 - 11 juillet 2017



# Qu'est-ce qu'un système de fichiers ?

- Un **système de fichiers** (*file system*) définit l'**organisation d'une partition d'un support de stockage**.
- C'est une **structure de données** permettant de stocker les informations et de les organiser dans des fichiers sur ce que l'on appelle des mémoires secondaires (disque dur, CD-ROM, etc.).
- Un système de fichiers offre à l'utilisateur une vue abstraite sur ses données (**fichiers**) et permet de les localiser à partir d'un **chemin d'accès** (*path*) dans une **arborescence de répertoires** (dossier).
- Le fichier est la plus petite entité logique de stockage sur un disque.
- Une partition est une partie d'un disque dur destinée à accueillir un système de fichiers.
- Le **formatage** prépare une partition d'un support de données de stockage en y inscrivant un système de fichiers, de façon à ce qu'il soit reconnu par le système d'exploitation de l'ordinateur.
- Il existe de nombreux systèmes de fichiers différents : FAT, NTFS, HFS, ext2, ext3, UFS, etc.



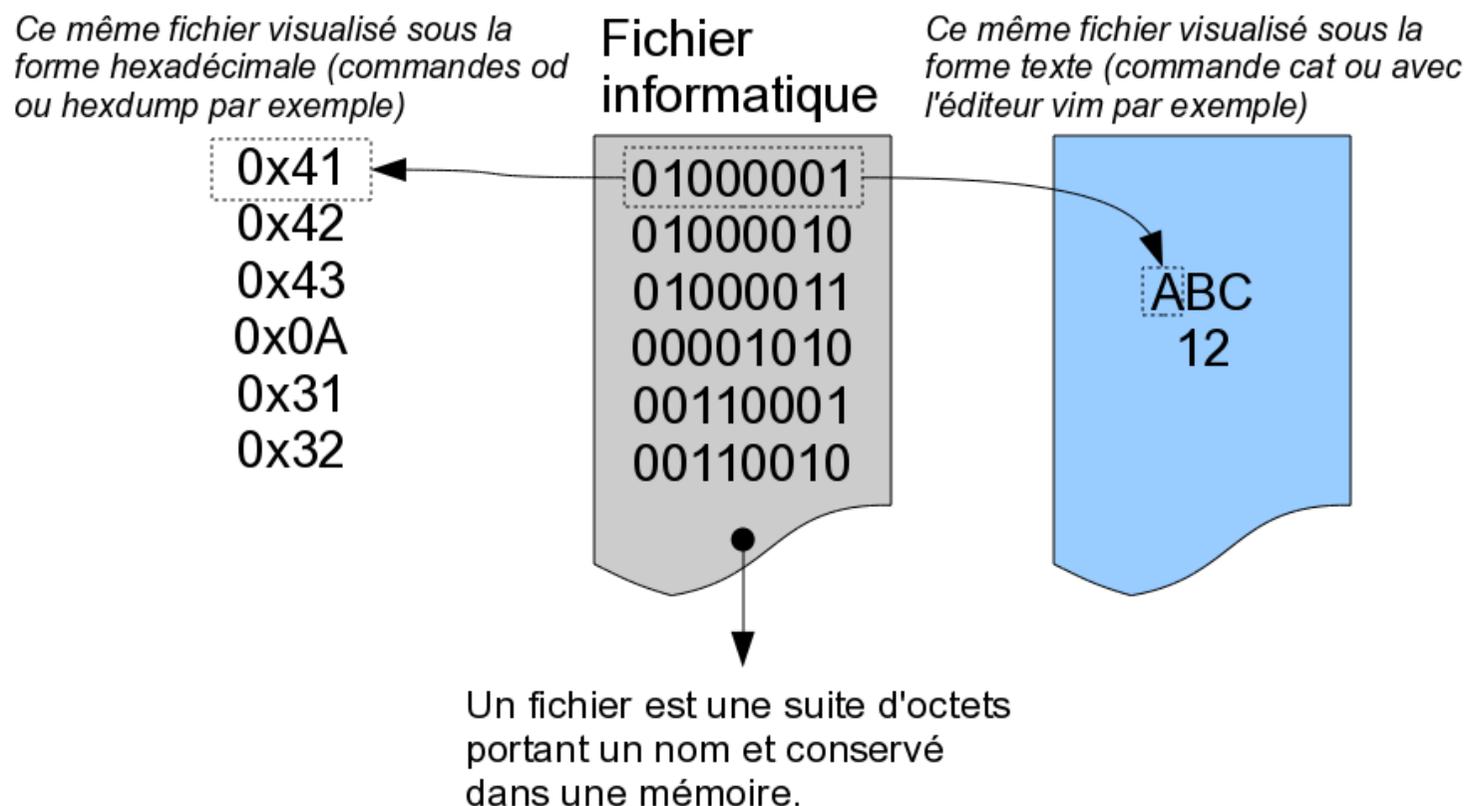
- Un **fichier** est une **suite d'octets** portant un nom et conservé dans une mémoire.
- Le contenu du fichier peut représenter n'importe quelle donnée binaire : un programme, une image, un texte, etc.
- Les fichiers sont classés dans des groupes appelés **répertoires**, chaque répertoire peut contenir d'autres répertoires, formant ainsi une **organisation arborescente**.
- Les fichiers sont la plupart du temps conservés (stockés) sur des **mémoires de masse** tels que les disques durs. Des systèmes de fichiers existent aussi pour la mémoire RAM.
- Dans un système d'exploitation multi-utilisateurs, les programmes qui manipulent le système de fichiers effectuent des **contrôles d'accès** (notion de **droits**).
- Les données descriptives (métadonnées) comme les dates de création et de modification, le propriétaire du fichier ainsi que les droits d'accès sont conservés dans le système de fichiers lui-même.

- Un système de fichiers découpe l'espace d'une partition en **blocs d'allocation** de taille fixe (paramétrable). Ceci induit que la taille physique (réellement occupée) est légèrement supérieure à la taille du fichier (exprimée en octets).
- L'**extension** d'un fichier, suffixe ajouté au nom du fichier pour indiquer la nature de son contenu, est seulement destinée à l'utilisateur. L'usage des extensions est une pratique généralisée sur **Windows** et une pratique courante sur **Unix**.
- Chaque fichier est vu par le système de fichiers de plusieurs façons :
  - un **descripteur** de fichier (souvent un entier unique) permettant de l'identifier au niveau système
  - une entrée dans un répertoire permettant de le situer et de le nommer
  - des métadonnées sur le fichier permettant de le définir et de le décrire
  - un ou plusieurs blocs (selon sa taille) permettant d'accéder aux données du fichier (son contenu)

- Les fichiers « **texte** » ont un contenu pouvant être interprété comme du texte (une suite de bits représentant un caractère), la plupart du temps en **codage ASCII**.
- Les fichiers « **binaire** » respectent un **format de fichier** (convention normalisée ou non) utilisé pour représenter et stocker des données.
- Un fichier (texte ou binaire) qui a subi une transformation par un algorithme en vue de diminuer sa taille est appelé **fichier compressé**. Le fichier transformé est un fichier binaire.
- Une **archive** est un fichier (binaire) dans lequel se trouve regroupé des fichiers ou tout le contenu d'une arborescence. Le but principal d'une archive est de tout contenir (données + descriptions) en un seul fichier. Les archives sont souvent compressés.

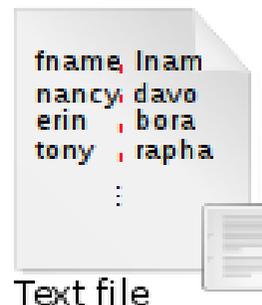
# Les fichiers « texte »

- Les fichiers « **texte** » ont un contenu pouvant être interprété comme du texte (une suite de bits représentant un caractère), la plupart du temps en **codage ASCII**. On utilise souvent un **éditeur de texte** pour les manipuler. Exemples : code source, fichiers de configuration, ...



# Les séparateurs

- Si les fichiers « **texte** » ne sont qu'une suite de caractères encodés, il est tout de même possible de les structurer.
- Pour cela, on utilise des **séparateurs** (*flag*) ou, délimiteur ou marqueur.
- Un séparateur est une **séquence de un ou plusieurs caractères** servant à délimiter la frontière entre différentes zones (**champs**) de texte ou autres flux de données.

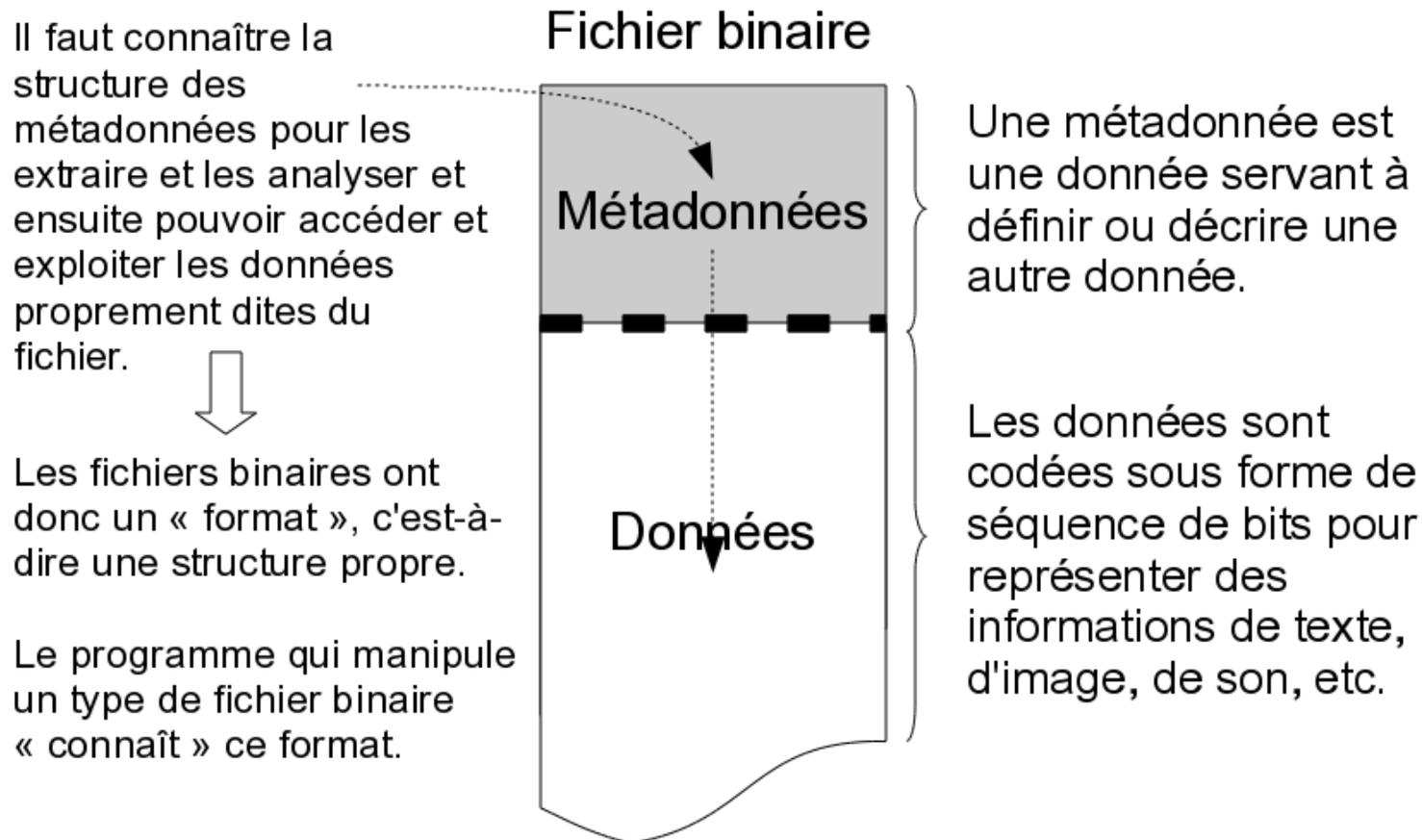


- On utilise généralement :
  - de simples caractères ASCII comme la virgule ',', les deux points ':', le point-virgule ';', etc ...
  - des caractères de contrôle du code ASCII : la tabulation '\t', le changement de ligne '\n', le retour chariot '\r', etc ...



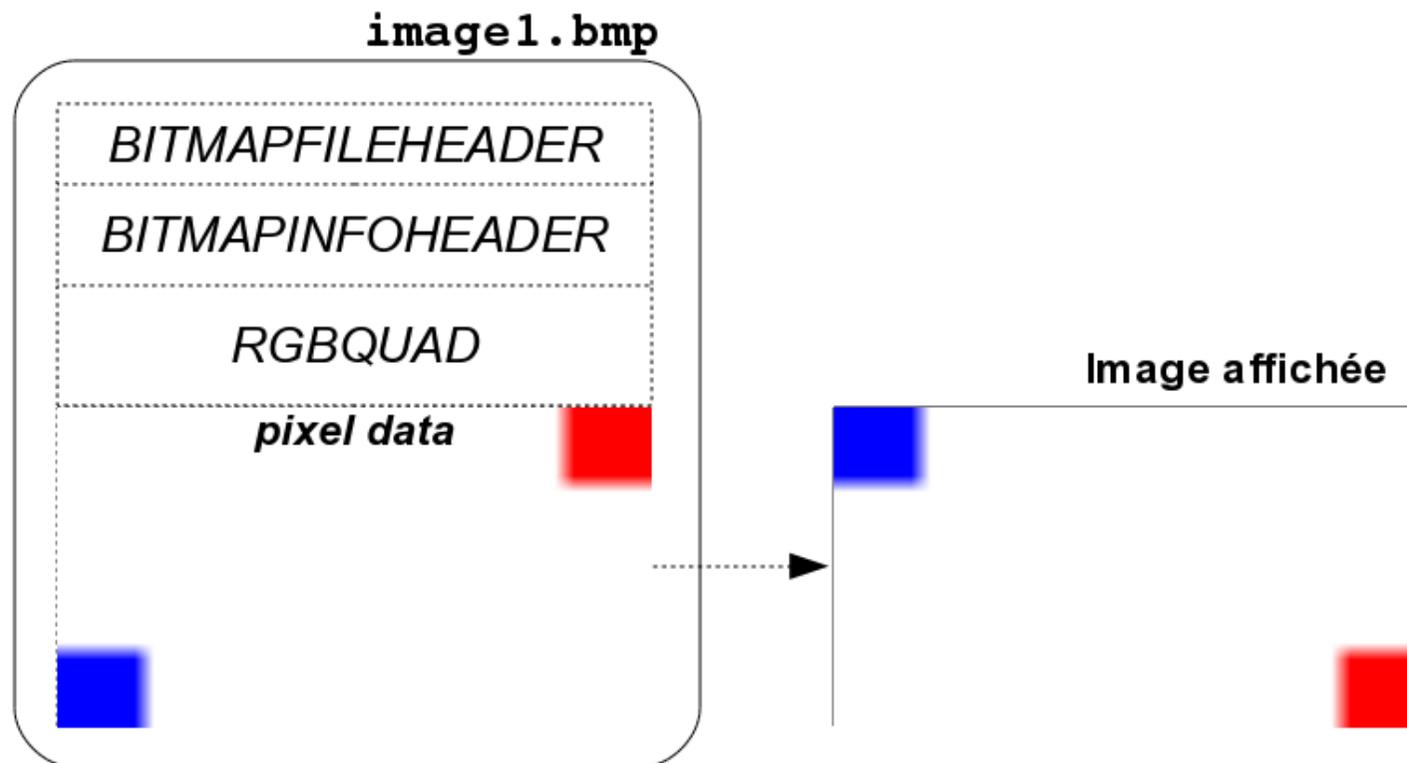
# Les fichiers « binaire »

- Les fichiers « **binaire** » correspondent souvent à un **format** précis lié à un logiciel applicatif spécifique. Exemples : code machine (exécutable), fichiers multimédias (images, sons, vidéos, traitement de texte, etc.). Tout ce qui n'est pas un fichier texte est un fichier binaire !



# Exemple de fichier binaire

- Les images BMP sont stockées dans des fichiers « **binaire** » qui respectent un format de fichier utilisé pour représenter et stocker les données (les *pixels*, des points en couleurs).



# Fichier BMP

Champ	Valeur en hexadécimale	Valeur décodée
Type (bfType) en ASCII	<b>42 4d (code ascii)</b>	<b>BM</b>
Taille (bfSize) en octets	<b>B6 07 00 00</b> donc 00 00 07 B6	<b>1974</b>
Offset (bfOffBits) en octets	<b>36 00 00 00</b> donc 00 00 00 36	<b>54</b>

Champ	Valeur en hexadécimale	Valeur décodée
Taille de cette entête (biSize) en octets	28 00 00 00 donc 00 00 00 28	<b>40</b>
Largeur (biWidth) en pixels	20 00 00 00 donc 00 00 00 20	<b>32</b>
Height (biHeight) en pixels	14 00 00 00 donc 00 00 00 14	<b>20</b>
Codage des couleurs (biBitCount) en bits	18 00 donc 00 18	<b>24</b>
Taille des données image (biSizeImage) en octets	80 07 00 00 donc 00 00 07 80	<b>1920</b>
Nb pixels par mètre en X (biXPelsPerMeter)	13 0b 00 00 donc 00 00 0b 13	<b>2835</b>
Nb pixels par mètre en Y (biYPelsPerMeter)	13 0b 00 00 donc 00 00 0b 13	<b>2835</b>

