

# info201 : Système d'exploitation

## TD 3 : fichiers

Pierre Hyvernat  
Laboratoire de mathématiques de l'université de Savoie  
bâtiment Chablais, bureau 17, poste : 94 22  
email : Pierre.Hyvernat@univ-smb.fr  
www : <http://www.lama.univ-smb.fr/~hyvernat/>

### Exercice 1 : questions de cours

*Question 1.* Qu'est-ce qu'un *fichier binaire* ? Donnez quelques exemples.

Qu'est-ce qu'un fichier texte *structuré* ? Donnez quelques exemples de formats de fichiers texte structurés.

*Question 2.* Que sont les *métadonnées* d'un fichier ? Donnez des exemples.

Est-ce que les métadonnées d'un fichiers occupent de l'espace disque ?

*Question 3.* Expliquez pourquoi les *données* d'un fichier de  $n$  octets occupent en général strictement plus que  $n$  octets sur son support de stockage.

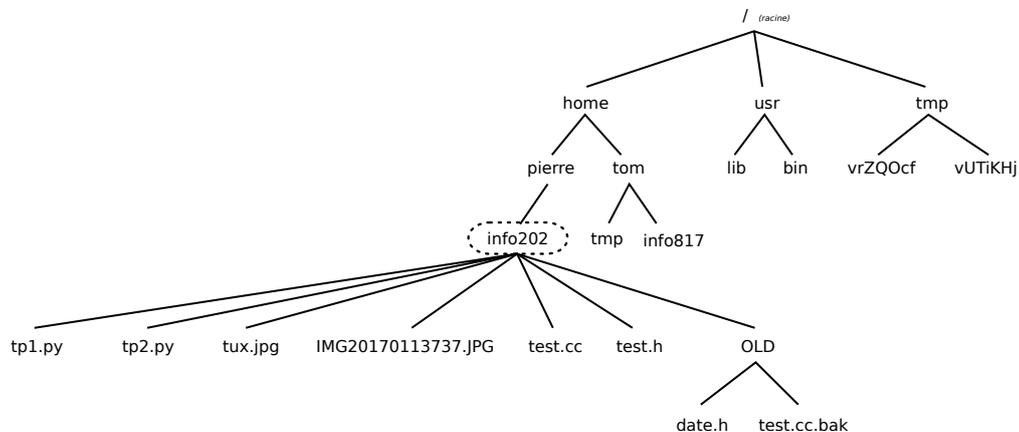
*Question 4.* Si les blocs du système de fichiers font 4kio, quel est l'espace gaspillé (en pourcentage de la taille du fichier) sur un fichier de

- 1 octet,
- 1 kibioctet + 1 octet,
- 1 mébioctet + 1 octet,
- 1 gibioctet + 1 octet ?

### Exercice 2 : noms de fichiers, chemins et motifs

*Question 1.* Qu'est-ce qu'un chemin d'accès ? Quelles différences y a t'il entre un chemin *absolu* et un chemin *relatif* ?

*Question 2.* Voici un morceau d'une arborescence de fichiers Unix : (**info202** est le répertoire courant)



Donnez des chemins absolus et relatifs pour accéder aux répertoires OLD, tom, info202, vrZQOcf et tmp.

*Question 3.* Pour chacun des motifs shell suivants, donnez la liste des fichiers correspondants dans l'arborescence ci dessus. (`info202` est le répertoire courant)

|                    |                     |
|--------------------|---------------------|
| <code>*.py</code>  | <code>*.*/*h</code> |
| <code>*.*</code>   | <code>*.*</code>    |
| <code>*[ch]</code> | <code>*/?*k</code>  |

### Exercice 3 : un système de fichiers simple

On considère un système de fichier où chaque fichier est stocké sur le disque de la manière suivante :

- les premiers octets contiennent le nom du fichier, en ASCII, suivi de l'octet 00,
- l'octet suivant contient un 1 en bit de poids faible (les 7 autres bits ne seront pas utilisés mais pourraient contenir d'autres méta-données)
- les 4 octets suivants contiennent la taille (en octets) du contenu du fichier,
- les octets suivants contiennent les données du fichier

Par exemple, le fichier `test.txt` contenant la chaîne `TEST` est codé par la suite de 16 octets suivante

|     |     |     |     |    |     |     |     |    |    |    |    |    |          |     |     |     |     |
|-----|-----|-----|-----|----|-----|-----|-----|----|----|----|----|----|----------|-----|-----|-----|-----|
| 74  | 65  | 73  | 74  | 2e | 74  | 78  | 74  | 00 | 01 | 00 | 00 | 00 | 04       | 54  | 45  | 53  | 54  |
| 't' | 'e' | 's' | 't' | .' | 't' | 'x' | 't' |    |    |    |    |    | 4 octets | 'T' | 'E' | 'S' | 'T' |

Chaque répertoire est codé de la manière suivante :

- les premiers octets contiennent le nom du répertoire, en ASCII, suivi de l'octet 00,
- l'octet suivant contient un 0 en bit de poids faible,
- pour chaque fichier ou répertoire contenu dans le répertoire on écrit ensuite l'adresse, sur 4 octets, du fichier ou répertoire correspondant.
- la liste se termine par les 4 octets `ff ff ff ff` (en hexadécimal).

Le répertoire "racine" n'a pas de nom et commence au début de la partition.

*Question 1.* Faites un schéma représentant l'état du disque si la partition contient l'arborescence `/tmp/projet` : contient uniquement la chaîne "TODO" `/tmp/a` : un fichier vide

*Question 2.* Comment peut-on détecter si un octet (ou une suite d'octets) est occupée par un fichier ?

*Question 3.* Quelles opérations doivent être faites par le système lors de

- la suppression d'un fichier,
- la création d'un fichier,
- la modification d'un fichier.

*Question 4.* Ce système de fichiers vous semble-t-il raisonnable pour une utilisation interactive (disque dur d'un ordinateur portable par exemple) ? Justifiez

Ce système de fichiers vous semble-t-il raisonnable pour une utilisation statique (sauvegarde sur un CD-ROM ou une bande magnétique par exemple) ? Justifiez

*Question 5.* Les systèmes de fichiers standard (Unix / Windows / Mac) utilisent des *blocs* de taille fixe (par exemple, 4kio) pour stocker les données de chaque fichier : ces données sont découpées, et le contenu du fichier se retrouve dispersé dans plusieurs blocs indépendants.

Parmi les opérations suivantes, lesquelles deviennent plus simples lorsque le système de fichiers utilise des blocs : "recherche de blocs libre", "suppression d'un fichier", "création d'un fichier", "modification d'un fichier".

Donnez un inconvénient à l'utilisation de blocs de taille fixe pour stocker les fichiers.

*Question 6.* Comment est-il possible de référencer les blocs utilisés par un fichier ?

Essayer de détailler les avantages et inconvénients de votre (vos) méthode (méthodes).