

# IN405 – Feuille de TD #4

## Système de fichiers et méta-données

*Objectif : manipulation des méta-données de fichiers à l'aide d'appels système.*

Instructions :

- vous aurez besoin de l'archive `td4-contents.tar`, la description de l'archive est donnée dans le fichier `README.md`, à lire avant de vous lancer dans l'implémentation ;
- l'exercice 4.2 est un TP noté :
  - il doit être fait en monôme et rendu **au plus tard le dimanche 21 mars, 23h59 (heure française métropolitaine)** sur moodle, dans la section 'Travaux Dirigés notés' ;
  - le rendu consiste en une archive `.tar` constituée des fichiers sources, du matériel de compilation et d'un fichier `README.md` décrivant comment utiliser votre programme ;
  - un malus de point sera attribué pour tout retard ou pour toute correspondance suspecte avec les autres rendus.

### Exercice 4.1 - Implémentation de la commande `ls`

Votre objectif est de reproduire le comportement de la commande `ls` en C.

**1-** En utilisant les fonctions `stat()` et/ou `lstat()`, écrivez les fonctions répondant aux comportements suivants :

- Affichage du type de fichier : socket (sock), lien symbolique (link), fichier régulier (file), device (devc), répertoire (diry), FIFO (fifo).
- Affichage des permissions d'accès au fichier : `rw-rw-rw-`.
- Affichage du propriétaire du fichier.
- Affichage de sa taille

**2-** En utilisant les fonctions `opendir()` et `readdir()`, affichez le contenu d'un répertoire donné.

**3-** A partir des codes des deux exercices précédents, écrivez une fonction affichant le contenu d'un répertoire, et indiquant pour chaque item, son type, ses permissions en écriture, son propriétaire et sa taille dans le format suivant :

```
type rwxrwxrwx owner size name1
type rwxrwxrwx owner size name2
```

## Exercice 4.2 - Protocole Phantom

L'exercice est composé de deux parties : une première consistant en la lecture d'un répertoire et des méta-données de l'ensemble de ses entrées, une seconde consistant en la création d'un répertoire dit 'phantom' à partir de méta-données.

Pour ceci, vous aurez à créer/modifier les fichiers `internals.c`, `api.c` et `main.c` en respectant les prototypes de fonctions et les structures de données fournies, décrites succinctement ici-même :

- `struct file` – structure de base d'une liste chaînée contenant les méta-données pour une entrée de répertoire, peut représenter un fichier régulier, un répertoire ou un lien symbolique ;
- `capture_file()` – fonction récupérant les méta-données d'un fichier présent dans le système de fichier afin de remplir la structure de données correspondante ;
- `browse_directory()` – fonction (récursive) parcourant un répertoire afin de remplir la structure de données correspondante ;
- `create_regular()` – fonction créant dans le système de fichiers un fichier régulier à partir de la structure de données correspondante ;
- `create_directory()` – fonction créant dans le système de fichiers un répertoire à partir de la structure de données correspondante ;
- `create_link()` – fonction créant dans le système de fichiers un lien symbolique à partir de la structure de données correspondante ;
- `capture_tree()` – fonction stockant le contenu d'une hiérarchie dans la structure de données correspondante ;
- `create_phantom()` – fonction créant une hiérarchie à partir de la structure de données correspondante.

Le comportement de la fonction `main()` est donc de récupérer les méta-données d'un répertoire donné, et de créer le répertoire 'phantom' correspondant.

La notation du TP se fera par l'exécution de tests unitaires sur les fonctions dont les prototypes sont donnés dans `internals.h` et `api.h` et par des tests d'intégration sur quelques scénarios. La non compilation du projet entraînera une pénalité.

D'autres informations :

- pour que la taille d'un fichier régulier soit correcte, remplissez le fichier du répertoire 'phantom' avec n'importe quel caractère jusqu'à obtenir la bonne taille ;
- ce qui est décrit dans les fichiers fournis est immuable : en modifier le contenu fera échouer les tests unitaires, et feront baisser la note ;
- la note donnée est proportionnelle au nombre de tests réussis : si vous bloquez sur une fonction, n'hésitez pas à passer sur une autre.