

## TD #3 – Système de fichiers

### Préambule

*Objectif* : Reproduction du comportement de commandes shell liées à la manipulation de fichiers

*Fichiers additionnels* :

- `filecore.py` – programme contenant les prototypes de fonctions des énoncés du TD ; ne changez pas les prototypes

### Commandes simples

#### **cat** – Affichage du contenu d’un fichier

Écrivez le corps de la fonction `file_cat()` qui permet d’afficher sur la sortie standard le contenu du fichier `path_in`.

#### **cp** – Copie d’un fichier

Écrivez le corps de la fonction `file_copy()` qui permet de copier le contenu du fichier `path_in` dans le fichier `path_out`. Si `path_out` existe, il doit être écrasé par la copie.

#### **mv** – Déplacement/renommage d’un fichier

Écrivez le corps de la fonction `file_move()` qui permet de changer le nom du fichier `path_in` par `path_out`. Si `path_out` existe, il doit être écrasé par le fichier déplacé.

#### **find** – Recherche d’un fichier

Écrivez le corps de la fonction `file_find()` qui permet de chercher récursivement dans le répertoire `path` tous les fichiers se nommant `filename`. L’affichage final liste les chemins relatifs de chaque fichier, à partir de `path`.

## **diff – Différence entre deux fichiers**

Écrivez le corps de la fonction `file_diff()` qui permet d'indiquer si deux fichiers `path_a` et `path_b` ont le même contenu. Le résultat retourné est un booléen valant `true` si le contenu est identique, `false` sinon.

## **sed – Commandes d'édition**

### **Remplacement d'un caractère**

Écrivez le corps de la fonction `file_sed_char()` qui permet de remplacer chaque occurrence de caractère `target` de `path` par le caractère `modif`.

### **Remplacement d'une chaîne de caractères**

Écrivez le corps de la fonction `file_sed_string()` qui permet de remplacer chaque occurrence de `target` de `path` par la chaîne `modif`.

## **grep – Commandes de recherche**

### **Recherche d'une ligne**

Écrivez le corps de la fonction `file_grep_line()` qui permet d'indiquer si le fichier `path_in` contient la ligne EXACTE `target`. Le résultat retourné est un booléen valant `true` si la ligne est présente, `false` sinon.

### **Recherche d'un caractère**

Écrivez le corps de la fonction `file_grep_char()` qui permet de lister les octets du fichier `path_in` valant le caractère `target`. Exemple :

```
$ cat ex.txt
aBcdabcD
aBCdABcd
$ python3
>>> from filecore import file_grep_char
>>> file_grep_char('ex.txt', 'B')
1
10
14
```

### **Recherche d'une chaîne de caractère**

Écrivez le corps de la fonction `file_grep_string()` qui permet de lister les emplacements de la chaîne `target` dans le fichier `path_in` sous la forme d'un couple `i_line:i_byte_in_line`. Exemple :

```
$ cat ex.txt
aBcdabcD
aBCdABcd
$ python3
>>> from filecore import file_grep_string
>>> file_grep_string('ex.txt', 'cd')
0:2
1:6
```