

## Programmation Système TD n° 1

Notions abordées :

- Primitives systèmes
- Gestion de fichiers (`open()`, `close()`, `read()`, `write()`, `lseek()`)

### Exercice 1

- a) Nommer quelques primitives système. Est-ce que `fopen()` est une primitive système ?
- b) Quelles sont les différences et ressemblances entre primitives systèmes et fonctions de bibliothèque ?
- c) Illustrer ces différences et ressemblances sur les fonctions et primitives d'entrée/sortie.

### Exercice 2

Pourquoi la fonction ci-dessous ne peut pas fonctionner ? Détailler toutes les fautes. (On ne demande pas de corriger cette fonction.)

```
int faux(char *nom)
{
    FILE *fp ;
    int c ;

    fp = open (nom, "r") ;
    read (fp, &c, 1) ;
    fclose (fp) ;
    return c ;
}
```

### Exercice 3

Utiliser la primitive système `open()` pour ouvrir un fichier qui sera spécifié sur la ligne de commande. Lire les 10 premiers octets du fichier avec la primitive système `read()`, puis les afficher sur la sortie standard en utilisant `write()`. Rappel : 0=entrée standard, 1=sortie standard, 2=sortie erreur standard.

### Exercice 4

Réécrire un programme `cp` simple (copie d'un fichier vers un autre) :

- a) octet par octet,
- b) par bloc de 256 octets.

### Exercice 5

Écrire un programme fonctionnant comme `cat`, c'est-à-dire ouvrant un par un les fichiers spécifiés sur la ligne de commande, les lisant, et les écrivant sur la sortie standard.

### Exercice 6

Écrire un programme qui affiche les 10 derniers octets d'un fichier :

- a) sans utiliser `lseek()`,
- b) en utilisant `lseek()`.