

IN405 – Feuille de TD #3

Système de fichiers

Objectif : utilisation des appels système liés à la manipulation de fichier à travers l'écriture d'une bibliothèque d'entrée-sortie.

Instructions :

- vous aurez besoin de l'archive `td3-contents.tar.gz`.

Exercice 3.1 - Bibliothèque de fonctions d'E/S

L'objectif de ce premier exercice est de créer une bibliothèque de fonctions d'entrée/sortie utilisant les appels systèmes liés au système de fichier. A cet effet, il vous est demandé d'écrire le fichier `io_file.c`, composé du corps des fonctions énoncées dans `io_file.h`. **Attention : vous devez respecter les prototypes du fichier d'en-tête.**

Pour compiler le fichier `io_file.c` et en obtenir une bibliothèque, vous devez utiliser les commandes suivantes :

```
$ gcc -c -fPIC io_file.c
$ ar rcs libiof.a io_file.o
```

Les fonctions de la bibliothèque sont alors contenues dans l'archive `libiof.a`. Pour utiliser les fonctions d'une bibliothèque dans un autre programme, le compilateur doit avoir accès à deux ressources : les fonctions compilées (ici `libiof.a`) et les prototypes de fonctions apportées par le fichier d'en-tête (ici `io_file.h`). En supposant que ces ressources sont situées dans le même répertoire que votre programme issu de `test.c`, sa compilation se fait à l'aide de la commande suivante :

```
$ gcc test.c -L. -lio
```

L'option `-L` permet d'indiquer au compilateur un chemin supplémentaire pour la recherche de bibliothèque ; le chemin indiqué est le répertoire courant. L'option `-l` lie la bibliothèque `iof` (`libiof.a`) au programme.

1- Écrivez la fonction `IO_open()` qui ouvre le fichier dont le chemin est donné en paramètre. Attention à bien créer le fichier s'il n'existe pas.

- 2- Écrivez la fonction `IO_close()` qui ferme le fichier donné en paramètre.
- 3- Écrivez la fonction `IO_remove()` qui supprime le fichier donné en paramètre.
- 4- Écrivez la fonction `IO_char_read()` qui lit le prochain caractère du fichier donné en paramètre. Vérifiez si le fichier a été ouvert avec les bonnes permissions.
- 5- Écrivez la fonction `IO_char_write()` qui écrit le caractère dans le fichier donné en paramètre. Vérifiez si le fichier a été ouvert avec les bonnes permissions.

Exercice 3.2 - Utilisation de la bibliothèque

A l'aide des fonctions de votre bibliothèque, remplissez les corps des fonctions suivantes du fichier `main.c`. Le programme issu de la compilation consiste en l'exécution de tests vérifiant le bon fonctionnement de ces fonctions.

- 6- Écrivez la fonction `print()` ayant le même comportement que la commande `cat`.
- 7- Écrivez les fonctions `copy()` et `move()` ayant respectivement le même comportement que les commandes `cp` et `mv`.
- 8- Écrivez la fonction `are_the_same()` qui retourne 1 si les fichiers sont identiques, et 0 sinon.

Exercice 3.3 - *Amélioration de la bibliothèque*

Vous allez maintenant ajouter des fonctions de lecture/écriture à votre bibliothèque. Pour ceci, décommentez les prototypes restants de `io_file.h`, et écrivez les corps correspondants dans `io_file.c`. Ces fonctions peuvent être testées grâce au programme issu de `main_opt.c`.

- 9- Écrivez les fonctions de lecture/écriture d'une chaîne de caractère.
- 10- Écrivez les fonctions de lecture/écriture d'un nombre entier (comportement de `scanf()` avec `%d`).
- 11- Écrivez la fonction de lecture d'un mot (comportement de `scanf()` avec `%s`).