UNIVERSITÉ DE REIMS CHAMPAGNE-ARDENNE

**Licence 3 Informatique**
**Info0601 – 2021/2022**
***Resp.* : Cyril Rabat**

# TP n°2

# Les fichiers bas-niveau

*Le but de ce TP est de manipuler les fonctions bas-niveau associées aux fichiers. Dans tous vos codes, vous devez tester systématiquement* ***tous les retours*** *des appels système.*

## 1 Premiers pas avec les fichiers bas-niveau

Dans cet exercice, nous allons exploiter les fonctions de base : `open`, `close`, `read` et `write`.

### Questions

1. Faites un programme qui demande à l'utilisateur de saisir un entier et qui le sauvegarde dans le fichier `toto.bin`.
2. À l'aide de la commande `od`, vérifiez le contenu du fichier. À quoi correspondent les suites de '00' ?
3. Que se passe-t-il si vous ouvrez le fichier avec un éditeur de code «classique» ?



Pour visualiser le contenu du fichier binaire avec `od`, utilisez la commande suivante : `od -t x1 toto.bin`

4. Modifiez votre programme et écrivez maintenant une chaîne de caractères (lue au clavier) après l'entier. Observez le contenu du fichier avec `od` et votre éditeur de code.

## 2 Déplacements dans un fichier

La fonction `lseek` permet de se déplacer dans un fichier. Elle retourne la nouvelle position dans le fichier (ou -1 en cas d'erreur).

### Questions

1. Comment utiliser `lseek` pour connaître la position courante dans un fichier sans se déplacer ?
2. Utilisez `lseek` pour déterminer la taille d'un fichier dont le nom est passé en argument.
3. Écrivez un programme qui sauvegarde les 10 premiers entiers dans un fichier. Affichez la position actuelle.
4. Modifiez votre programme : une fois les entiers sauvegardés, déplacez-vous à la position 1024. Que se passe-t-il ? Quelle est la taille du fichier ?
5. Et si vous écrivez maintenant un caractère à la position 1024 ?
6. Utilisez la commande `stat` sur le fichier généré.

# 3 Copie de fichier

Le but de cet exercice est de réaliser une copie d'un fichier quelconque (fichier texte ou binaire) dans un autre fichier.

## Questions

1. Écrivez un programme permettant de tester l'existence d'un fichier dont le nom est passé en argument, en utilisant uniquement l'appel système `open` (en utilisant les bonnes options et la gestion d'erreurs avec `errno`).
2. Modifiez le programme. Il prend en arguments le nom d'un fichier source et le nom d'un fichier de destination. Si le fichier source n'existe pas ou si le fichier de destination existe déjà, une erreur est affichée. Le fichier source ne doit pas être créé (dans cette question).
3. Comment copier un fichier dans un autre sans placer l'intégralité de son contenu en mémoire ?
4. Modifiez le programme pour qu'il procède à la copie (sans placer l'intégralité du contenu du fichier source en mémoire).



Pour vérifier que la copie s'est bien déroulée, commencez par comparer la taille des deux fichiers. Vous pouvez utiliser également la commande `diff`.

# 4 Phrase cachée

Un fichier binaire contenant des données sensibles a été intercepté. Nos experts ont réussi à comprendre comme le déchiffrer et font appel à vous pour coder le programme et retrouver la phrase cachée (car ils n'ont pas suivi le cours d'INFO0601, eux...). Vous trouverez le fichier sur le git.

Au début du fichier se trouve une position (un `off_t`). Il faut se déplacer à cette position qui comprend une position gauche, un caractère et une position droite. Une des deux positions peut être nulle ou les deux (dans ce cas, il s'agit d'un cul-de-sac). Il faut explorer tous les chemins pour trouver la phrase cachée. Elle termine forcément par un point, dans un cul-de-sac.

## Questions

1. Quelles structures sont nécessaires pour ce programme ?
2. Écrivez les structures nécessaires.
3. Écrivez le programme demandé et trouvez la phrase cachée.