

Systeme TD 2

EXERCICE 1 : utilisation des fonctions

Tout d'abord créer un fichier **test1.txt** avec 3 lignes écrites.

Écrire un programme effectuant les instructions suivantes :

- Création : `creat(nom, mode);` // créer un fichier **test2.txt**
- Ouverture :
 - `int fd1 = open(char[], O_RDONLY);` // ouvrir en lecture le fichier **test1.txt**
 - `int fd2 = open(char[], O_WRONLY);` // ouvrir en écriture le fichier **test2.txt**
- Lecture : `int read_size = read(fd, buffer, taille);` // lire 10 caractères de **test1.txt**
- Écriture : `int write_size = write(fd, buffer, taille);` // écrire ces 10 caractères dans **test2**
- Fermeture : `close(fd);` // fermer les fichiers **test1.txt** et **test2.txt**
- Renommer : `rename(ancienNom, nouveau);` // renommer **test2.txt** en **test3.txt**
- Suppression : `unlink(nom);` // supprimer le fichier **test1.txt**

EXERCICE 2 : utilisation de paramètre

Proposer un programme qui prend en paramètre (*argv*) deux noms.

Le but du programme est de copier le contenu du fichier portant le nom du premier argument, dans un fichier portant le nom du second argument, puis d'effacer le premier fichier.

Précision : le deuxième fichier n'existe pas, il sera créé par le programme

EXERCICE 3 : lecture partielle

Faire un programme permettant d'ouvrir un fichier contenant au moins 30 caractères, de lire les 10 caractères suivant les 15 premiers.

Fonction utile

`lseek(fd, nombre_caracteres_a_decaler, option);`

Options : `SEEK_CUR`, `SEEK_SET`, `SEEK_END`

EXERCICE 4 : utilisation des fonctions, la suite

Écrire un programme effectuant les instructions suivantes :

- Création : `s = mkdir(chemin, mode);` // créer un dossier **TEST**
- Création de lien : `s = link(fichier, lien);` // définir un lien du fichier **test.txt** vers **t_link**
- Suppression de lien : `s = unlink(chemin);` // supprimer le lien
- Changement de répertoire : `s = chdir(chemin);` // aller dans le répertoire **TEST**
- Création : créer un fichier **test_td2.txt** dans le répertoire **TEST**
- Suppression : `s = rmdir(chemin);` // supprimer le dossier **TEST**

EXERCICE 5 : *listing* d'un dossier

Faire un programme qui liste l'ensemble du contenu d'un dossier passé en paramètre (argv) en affichant le nom et le type de chaque objet listé (fichier, lien, répertoire...).

EXERCICE 6 : *listing* récursif de dossier

Variante du programme précédent.

Lister le contenu d'un répertoire et des répertoires situés à l'intérieur.

EXERCICE 7 : création d'une arborescence

Écrire un programme demandant à l'utilisateur de saisir des noms de dossier à créer.

EXERCICE 8 : un peu plus

Écrire un programme se présentant sous forme d'un menu, avec pour choix la création de fichier ou de dossier, et la possibilité de naviguer dans les dossiers.

EXERCICE 9 : un terminal

Écrire un programme permettant de simuler un terminal. Ce dernier pourra :

- afficher le contenu d'un répertoire
- créer des fichiers
- les déplacer
- ...