

TP n°2

Les fichiers bas-niveau

Le but de ce TP est de manipuler les fonctions bas-niveau associées aux fichiers vues en cours. Il est important de répondre à toutes les questions avant de commencer le deuxième exercice qui complète le premier projet.

1 Un peu d'entraînement

Pour l'ensemble des questions suivantes, la gestion d'erreurs doit être correctement mise en place pour tous les appels systèmes.



Pour visualiser le contenu d'un fichier binaire, vous pouvez utiliser la commande suivante : `od -t x1 fichier.bin`.

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).
2. À l'aide de `lseek`, modifiez votre programme pour qu'il affiche la taille du fichier.
3. Écrivez un programme qui prend en argument le nom d'un fichier. S'il n'existe pas, un fichier vide est créé. Le programme demande ensuite à l'utilisateur de saisir une chaîne de caractères qui est ajoutée à la fin du fichier. Ensuite, toutes les chaînes de caractères du fichier sont affichées, une par une, une sur chaque ligne.
4. Écrivez un programme qui permet de réaliser une copie d'un fichier dans un autre fichier. Indication : la copie doit être réalisée bloc par bloc. Si le fichier de destination existe déjà, une erreur doit être affichée.

2 Un éditeur hexadécimal

Nous souhaitons réaliser un éditeur hexadécimal qui permet de visualiser le contenu d'un fichier binaire quelconque et de le modifier (voir comme exemple l'éditeur en ligne <https://hexed.it/>). Le contenu est affiché par bloc de 16×8 , à la fois en hexadécimal et en caractères (pour les caractères qui peuvent être affichés comme les chiffres, les lettres, etc.).

Questions

1. Écrivez des fonctions permettant :
 - De lire le n^{ime} bloc de X octets d'un fichier
 - D'afficher un bloc d'octets au format hexadécimal (utiliser les formats de `printf`)

- D’afficher un bloc d’octets au format caractère (les caractères inconnus peuvent être affichés)
 - De modifier un octet d’un fichier
 - De supprimer le n^{ime} octet d’un fichier par un “.”)
2. Écrivez un programme pour tester vos fonctions. Le fichier binaire à ouvrir doit être spécifié en argument du programme.
3. En exploitant les fonctions précédentes (et en les adaptant si nécessaire), écrivez l’éditeur hexadécimal avec l’interface de l’application du TP précédent.
- Le premier bloc de 16×8 octets du fichier est lu et affiché
 - Les flèches permettent de passer au bloc suivant/précédent
 - En cliquant sur un octet (soit sur la partie caractères, soit sur la partie hexadécimale), on le met en évidence (on peut l’afficher en noir sur fond blanc, par exemple)
 - En appuyant sur la touche “entrée”, on édite la valeur de l’octet
 - En appuyant sur la touche “supprimer”, on supprime l’octet



- L’utilisation de `ncurses` n’est utile que pour la dernière question.
- Vous ne devez jamais mettre l’intégralité du fichier en mémoire (au maximum un bloc)