

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 de décors

Nous souhaitons réaliser un éditeur pour créer des décors sur lequel faire tomber la neige. Le but est de réutiliser la même interface que l'application du TP précédent. Cette fois-ci, lorsque l'utilisateur clique avec le bouton gauche dans la fenêtre de simulation, cela ajoute un obstacle ou le supprime s'il y en a déjà un. Lorsque l'utilisateur quitte l'éditeur, le décor est sauvegardé dans le fichier.

Pour exécuter l'éditeur, il y a deux solutions :

- Si l'utilisateur spécifie un nom de fichier et qu'il existe, il est ouvert en mode édition.
- Si l'utilisateur spécifie un nom de fichier et un nom de décor, le fichier est créé (s'il n'existe pas déjà, sinon une erreur est affichée).

Le fichier correspondant à une carte contient :

- Le titre du décor qui est affiché à la place du titre de la fenêtre de simulation (`size_t` suivi des caractères de la chaîne incluant le `\0`)
- Les cases (`unsigned char × 30 × 15 : 0` pour un vide, `1` pour un obstacle)

Questions

1. Créez les fonctions permettant de créer un nouveau fichier ou d'en charger un existant.
2. Créer un `main` permettant de gérer les arguments.
3. Fusionnez votre programme avec l'interface de l'application du TP précédent.
4. Complétez votre programme.



L'utilisation de `ncurses` n'est utile que pour les deux dernières questions.