Alparlan Yuksel

Rapport FileDisk

1/ Lancement du programme

Pour lancer le programme :

- Aller dans le dossier source
- Executer la commande « make »
- Lancer le programme .prog avec la commande « ./prog »

2/ Description

Le programme simule la gestion d'un fichier avec les fonctions tel que create, write et read. Pour ce faire, il existe deux fichier MyStdIO et MyDiskBlockIO , ainsi que leur fichier header respectifs. MyStdIO est utilisé en tant qu'interface entre l'utilisateur et la gestion du Disk. MyDiskBlockIO manipule les données et utilise les libraires nécessaires afin de pouvoir écrire ou lire sur le Disk. C'est donc ce fichier qui contient l'implémentation concrète de ces fonctions.

Aussi, il y'a un fichier main, qui simule un scénario afin d'utiliser les fonctionnalité du programme.

1. MyStdIo.create(myfile) appelle MyDiskBlockIO.createFile(file)

Cette fonction créer un fichier avec le nom donné en argument. Concrètement il créer un dossier dans le repertoire courant du projet qui fera office de fichier.
Cette fonction return le descripteur de fichier.

2. On créer un tableau de short de taille 96 que l'on souhaite écrire dans le fichier créer précédemment.

3. MyStdIo .writ(file, data) appelle MyDiskBlockIO.writeFile(file, data)

Ici, dans un premier temps on calcule le nombre de block de 10 qui vont être écrit dans le dossier, ainsi que le dernier block inferieur a 10 qui sera le reste.
Ensuite on boucle en appelant createBlock et writeBlock , qui l'une créer les block de donnée (fichier texte) et l'autre écrit les donnée du tableau de short, 10 par 10.

Pour le reste on effectue le même processus, avec des arguments diffèrent selon le reste des octet a écrire.

A ce stade notre dossier source contient un dossier avec des blocks de données, numérotés par ordre croissant.

4. MyStdIo.readd(file) appelle MyDiskBlockIO.readFile(file)

Cette fonction sert lire les données d'un fichier (ici d'un dossier).
Pour cela ouvre le dossier pour compter le nombre de fichier texte (blocs) existant dans ce dernier.
Puis, on lit chaque fichier par ordre croissant, en récupérer chaque caractère utilise (on omet les caractères tel que le retour chariot, etc..).