

TP6 : Projection d'un fichier en mémoire: mmap

Matthieu Boutier (*boutier@pps.univ-paris-diderot.fr*)

1 Compter en parallèle

1. Récupérez le corrigé de l'exercice du TP4 qui comptait en parallèle utilisant un fichier en mémoire.
2. Utilisez `mmap(2)` à la place de `read(2)` pour lire l'intégralité du fichier.
3. Utilisez `mmap(2)` à la place du fichier d'échange afin de synchroniser les données. Il ne sera pas la peine d'utiliser un descripteur de fichier valide (donnez -1).

2 Modification d'un fichier

Dans cet exercice, nous allons lire et modifier (ou non) un fichier, en utilisant `mmap(2)`. Bien sûr, toutes vos entrées/sorties se feront à l'aide d'appels systèmes (`putchar(3)` est toléré pour l'affichage sur la sortie standard).

1. Créez un fichier `txt2MAJ.c`.
2. L'appel système `mmap` permet de projeter des fichiers en mémoire, à partir de leur descripteur de fichier. Commencez-donc par ouvrir le fichier d'entrée.
3. Utilisez `mmap()` pour afficher sur la sortie standard le contenu d'un fichier texte, dont le nom est passé en argument à `main()`. Rappel, on peut avoir la taille d'un fichier par `fstat(2)`.
4. Avant de faire l'affichage, modifiez le fichier en mémoire pour mettre le texte en majuscules. Attention, la modification ne doit pas être répercutée sur le fichier disque, mais seulement sur la projection mémoire !
5. Maintenant on veut pouvoir donner le choix à l'utilisateur de répercuter ou non la modification. S'il souhaite modifier le fichier, il rajoute "-w" comme deuxième argument de la ligne de commande. Exemple d'utilisation :

```
cat fichier.txt
Ceci est un fichier Texte
./a.out fichier.txt
CECI EST UN FICHER TEXTE
cat fichier.txt
Ceci est un fichier Texte
./a.out fichier.txt -w
CECI EST UN FICHER TEXTE
cat fichier.txt
CECI EST UN FICHER TEXTE
```

3 Copie de fichier

Nous allons maintenant utiliser une copie de fichier rapide, utilisant `mmap(2)`.

1. Créez un fichier `copie.c`, qui commence par ouvrir les fichiers d'entrée et de sortie dans le but de les projeter en mémoire : le fichier de sortie sera ouvert en écriture *et* lecture.
2. Lorsqu'on essaie de lire ou d'écrire sur une zone projetée en mémoire, il faut que le fichier associé ait alloué assez d'espace : `mmap` ne sera pas capable d'étendre le fichier. Utilisez `fstat(2)` pour récupérer la taille du fichier source, et `ftruncate(2)` pour régler la taille du fichier destination.
3. Projetez les deux fichiers en mémoire, en utilisant `mmap`. N'oubliez pas de libérer la mémoire !
4. Finissez le travail : copiez les fichiers ¹.

1. Et veillez à ce que l'espace alloué soit libéré, les descripteurs de fichier fermés, etc.