

TP/TD 2 : Copier

Le but de ce TD est d'écrire un clone de `cp` en C, pour vous faire manipuler les appels systèmes en C.

1 Le cœur du copieur

Question 1

Dans un fichier `copy.c`, écrivez une fonction `copy` à deux arguments de type `const char*` vide, et une fonction `main` qui analyse les arguments de la ligne de commande qui doivent d'être de la forme

```
<file1> <file2> ... <filen> <dest>
```

qui teste si `dest` est bien un dossier (indice: utilisez `stat`) et qui appelle ensuite `copy(file1, dest), ..., copy(filen, dest)`.

Question 2

Écrivez une fonction `destination` qui attend deux `const char *` comme `copy` (le premier argument est la source et le second le répertoire de destination) et qui renvoie un `char *` qui correspond au fichier de destination.

Question 3

Complétez `copy` pour calculer le chemin du fichier de destination, ouvrir le fichier source et le fichier de destination et de copier les données via `read` et `write` sur un tampon.

Si le fichier destination existe déjà, affichez un message d'erreur.

Question 4

Testez votre programme avec différentes taille pour le tampon. Quelle est la meilleure taille? Vous pouvez utiliser `time <commande>` dans votre shell pour savoir le temps qu'a pris une commande.

2 Une barre de progression

Question 1

Dans un terminal, tapez `echo "$COLUMNS"`. Redimensionnez votre terminal et réessayez.

Déduisez-en une fonction C `line_width` qui ne prend pas d'argument et qui renvoie le nombre de caractère qui tient sur une ligne dans le terminal.

Indice: utilisez la fonction `system` de la librairie standard C.

Question 2

Écrivez une fonction `file_size` qui prend un argument un `const char*` représentant un nom de fichier et qui renvoie sa taille en octets.

Question 3

Dans la boucle de `copy`, en combinant les fonctions précédentes, affichez une barre de progression (comme `wget`) pour indiquer la progression de la copie. Indice: si vous affichez le caractère `\r` (par exemple en faisant `printf("\r");` le curseur reviendra à la ligne.

3 Une copie récursive

Question 1

Écrivez une fonction `copy_dir` vide et modifier `copy` pour appeler `copy_dir` lorsque son premier argument est un dossier.

Question 2

Complétez `copy_dir`. Comment gérer la récursion mutuelle entre `copy` et `copy_dir`?

4 Une copie en mémoire

Question 1

Lisez le manuel de `mmap`.

Question 2

Modifier `copy` pour utiliser `mmap` à la place de `read` et `write`. Que remarquez vous en terme de performance?