

Programmation Procédurale

Feuille 6

Programmation modulaire: Table

Les exercices de cette feuille de TD mettent l'accent sur la construction de programmes modulaires. Le programme que vous devez écrire devra être formé de deux fichiers et la compilation devra se faire avec la commande make.

Pour réaliser ce TD, vous devez utiliser l'archive qui est distribuée sur le site du cours.

Exercice 1: La structure de données *Table*

Nous allons définir une table triée par ordre croissant d'identificateurs dont l'implémentation est masquée pour l'utilisateur. A chaque identificateur est associé le nombre d'occurrences de celui-ci (nombre de fois ou il a été ajouté dans la table).

Pour ce faire, nous allons définir

1. le type `Table` comme un type pointeur vers une structure C non définie ainsi qu'un ensemble de fonctions permettant la manipulation de cette table.
2. le type `t_fonction` comme un pointeur vers une fonction ayant le prototype suivant :

```
void (*fonction) (char *elt, int nb);
```

Ces deux définitions devront être ajoutées au début du fichier `table.h`.

Exercice 2: Implementation à l'aide d'une liste simplement chaînée

Nous allons implémenter maintenant la table sous la forme d'une liste ordonnée simplement chaînée. Cette implémentation se fera dans le fichier `table_liste.c` (ne pas oublier d'inclure `table.h`). Définir la structure de données équivalente (chaque cellule de la liste contiendra une chaîne de caractères, un entier et un pointeur vers la cellule suivante). Implémenter les fonctions de manipulation de la table.

Exercice 3: Implémentation à l'aide d'un tableau

Nous allons implémenter la table définie précédemment sous la forme d'un tableau trié alloué dynamiquement. Cette implémentation se fera dans le fichier `table_tableau.c`. On pourra commencer par une version qui utilise un tableau de taille fixe, puis par une version où le tableau est redimensionné si nécessaire (comme avec la fonction `realloc` vu dans la feuille précédente).