TP numéro 3
Tables de hachage

Programmation impérative avancée, ENSIIE

Semestre 2, 2015–16

On reprend l'interface des dictionnaires des séances précédentes, ainsi que le fichier de tests. On complétera le `Makefile`.

Exercice 1 : Tables de hachage

1. Dans un fichier `hashtbl.c`, écrire une fonction `hash` qui prend en argument une clef k et un entier m et qui implémente la méthode de la multiplication pour calculer la valeur de hachage de k pour une table de taille m. On utilisera comme constante A 0,6180339887. (Utiliser un `#define`.)
2. Implémenter les dictionnaires à l'aide de tables de hachage. On n'oubliera pas le redimensionnement dynamique.
3. Modifier le `Makefile` pour permettre la compilation du programme de test `test_hashtbl` utilisant cette implémentation.
4. Compiler, tester.

Exercice 3 : Comparaison de complexité

5. Si vous ne l'avez pas déjà fait aux séances précédentes, modifier le fichier `test.c` pour effectuer les opérations suivantes, où n est un entier passer en paramètre au programme :
   - Créer un dictionnaire de taille $\frac{n}{10}$.
   - Pour i allant successivement de 0 à $n-1$, insérer une association entre i et la chaîne contenant sa valeur.
   - Supprimer $\frac{n}{2}$ éléments associés à un entier choisi aléatoirement entre 0 et $n-1$. On libérera la mémoire allouée pour la chaîne de caractère.
   - Afficher, pour chaque entier entre 0 et $n-1$, soit la valeur associée dans le dictionnaire si elle existe, soit un message disant qu'elle n'existe pas.
6. Instrumenter le code de `test.c` avec la fonction `clock()` (cf. man) pour afficher le temps moyen en μs nécessaire pour réaliser chacune des opérations d'insertion, de recherche et de suppression. (On enlèvera les affichages.)
7. Compiler et comparer les quatres implémentations des dictionnaires.