Bibliothèque de gestion de table

Médéric Hurier
mederic.hurier@etudiant.univ-nancy2.fr
Licence ISC parcours MIAGE

7 février 2012

Année 2011-2012

Table des matières

1 Utilisation

1.1 Compilation

Pour compiler le programme, exécutez la commande "make" dans le répertoire contenant les sources. Le résultat de cette étape est un exécutable nommé "test_table" au même niveau que le "Makefile".

Les fichiers intermédiaires (.o) sont automatiquement supprimés.

1.2 Exécution

Le programme ne prend qu'un seul paramètre : le nombre d'entrée dans la table de hachage. *Il ne s'agit pas d'une limite sur le nombre total d'enregistrement* pouvant être stocké, mais d'un indice de répartition. Le jeu de donné du programme allant jusqu'à 15 éléments, vous pouvez passer un nombre strictement positif, inférieur ou supérieur à 15.

La valeur par défaut est de 255. On peut noter que avec une valeur égale à 1, la bibliothèque se comporte comme une liste chainée.

Quelques exemples d'utilisation :
▷ './test_table'
▷ './test_table 3'
▷ './test_table 6000'

1.3 Résultat

Le résultat de l'exécution affiche les états de la table après plusieurs opérations :
▷ Création de la table
▷ Insertions de clés/valeur
▷ Suppression de valeur
▷ Recherche par paire de clé
▷ Destruction de la table (libération de la mémoire)

Les données d'entrées sont les candidats à l'élection présidentielle de 2012.

Voici une ligne de résultat : *"90 : {'HervéMorin'= Nouveau Centre}"*
▷ 1725 est le numéro de la case de la table
▷ 'HervéMorin' est la concaténation des clés (nom et prénom)
▷ 'Nouveau Centre' est la valeur de la clé (n-uplet)

Vous pouvez faire varier le paramètre d'entrée, et oberserver comment les collisions sont gérées.

2 Algorithmes et structures

2.1 Table de hachage

La bibliothèque implémente une table de hachage. C'est *une structure efficace*, car elle permet un accès et une insertion en O(1) dans la plupart des cas.

Une table de hachage permet de répartir les enregistrements dans des cases selon une fonction de hachage. Le nombre de case est fixe, mais elle peut gérer plus d'enregistrement et de manière plus sûr en gérant les collissions. La façon la plus simple de les gérer est de créer une liste chaînée pour chaque case.

Voici un diagramme présentant les structures de l'application.

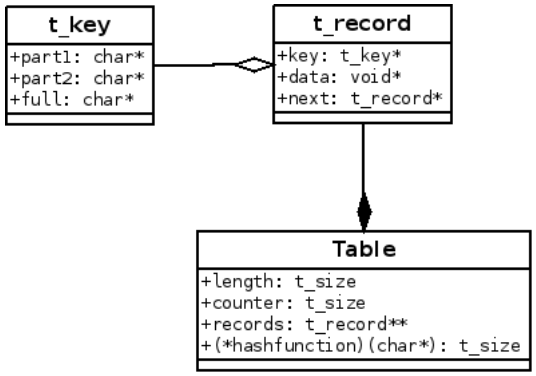

FIG. 1 – Structures de l'application

Détail des champs de la table de hachage :
- ▷ length : nombre de case de la table
- ▷ counter : nombre d'élément présent dans la table (utilise un compteur pour éviter de parcourir toute la table)
- ▷ records : une liste de pointeur vers des enregistrements
- ▷ hashfunction : fonction de hachage

2.2 Fonction de hachage

Le but d'une fonction de hachage est de renvoyer l'index d'une case pour une clé (une paire de chaîne de caractère). De nombreux algorithmes existent, avec des divergences en terme de rapidité et d'efficacité.

En analysant plusieurs fonctions j'ai sélectionné l'algorithme SDBM expliqué sur cette page. Il a l'avantage d'être rapide, générique et public.