ORAL PROBATOIRE DU CYCLE DU C.N.A.M.

INFORMATIQUE

OPTION GESTION

SUR QUELQUES TECHNIQUES D'ADRESSAGE PAR "HASH-CODE"

Monsieur GOUARDERES Guy

TOULOUSE, le 30 Avril 1976

PLAN DE L'EXPOSE

I - GENERALITES SUR LES PROBLEMES D'ADRESSAGE
DEFINITION D'UN "HASH-CODE"

II - ETUDE DE QUELQUES FONCTIONSDE "HASH-CODE"
1. Méthodes multiplicatives
2. Méthodes logiques
3. Méthodes additives
4. Méthodes par changement de base
5. Méthodes par division
6. Méthodes combinatoires
7. Méthodes statistiques

III - TRAITEMENT DES COLLISIONS
1. Résolution linéaire
2. Résolution par quotient linéaire
3. Résolution aléatoire
4. Résolution quadratique
5. Résolution par quotient quadratique
6. Résolution par chaînage

IV - COMPARAISONS ET EFFICACITES DES DIFFERENTES METHODES

V - TRAITEMENT DES DEBORDEMENTS

VI - CONCLUSION

I - GENERALITES

Parmi les méthodes d'accès, celles par "Hash-code" ne sont pas les plus utilisées. Elles offrent pourtant des solutions rapides et élégantes, notamment pour les compilateurs ou les traducteurs.

Dans ce qui suit, nous nous efforcerons de décrire les plus courantes en mentionnant quand c'est possible des références bibliographiques qui permettront au lecteur d'approfondir la question ou de trouver un exemple d'application.

La littérature française est peu prolixe sur ce sujet et les ouvrages étrangers (Anglo-saxons essentiellement) peu aisés à trouver.

Pour les méthodes statistiques en particulier nous n'avons pu trouver de références d'articles spécifiques.

Nous avons plus particulièrement développé la partie traitant des collisions, car c'est un problème inhérant à tout "Hash-code", et sa solution est essentielle pour l'efficacité de la méthode.

Enfin, nous nous sommes grandement inspiré de l'exposé de D.E. KNUTH dans "The Art of Computer programming" (1973) en le complétant par des articles originaux ou plus récents (cf. Annexe bibliographique).

Définition d'un "Hash-code" : c'est une fonction $\mathcal{H}$ qui associe directement une adresse à son contenu, autrement dit, qui transforme l'argument en une adresse correcte sans recherche dans la table ou dans la zone d'information.

Il faut donc connaître à priori le champ des possibilités de cette adresse, c'est-à-dire la taille de la zone d'information. D'autre part, l'accès se faisant de la même façon sur toutes les adresses, il faut imposer une taille fixe à chaque enregistrement.

Dans ce qui suit, on utilisera les notations suivantes :

- $\mathcal{H}$: fonction de "Hash-code"
- K : argument de recherche (ou clé)
- $\mathcal{A}_K$: adresse réelle de l'information associée à K alors

$$\mathcal{H}(K) = \mathcal{A}_K$$

- $\mathcal{H}$ doit se calculer rapidement.
- $\mathcal{H}$ doit assurer un bon éparpillement des adresses dans la zone d'information, ce qui diminue les risques de collision, c'est-à-dire d'avoir une même adresse pour des arguments différents ; c'est cette idée d'indépendance Argument/Adresse (ceci afin d'assurer la meilleure répartition) qui va guider tous les procédés décrits plus loin.

En fait, il n'existe pratiquement pas de fonction simple répondant dans tous les cas à ces deux contraintes. Toutefois, pour des tables pas trop étendues (ne dépassant pas 10^4 éléments), l'expérience prouve que les fonctions les plus utilisées donnent d'assez bons résultats et inversement.

L'idée originale de telles méthodes est due à H.P. LUHN (I.B.M. 1953, rapport interne), mais ce n'est qu'en 1956 qu'elles sont décrites pour la première fois (A.I. DUMEY, 1956). Depuis, d'autres méthodes ont été développées par R. MORRIS, W.D. MAURER, W.W. PETERSON, D.E. KNUTH, etc...

Le terme anglais de "Hash-code" trouve son origine dans le verbe To hash (hâcher, découper) et hash (hâchis, mélange des petits bouts découpés). On trouve aussi le terme de "scatter storage" (adresse dispersé) d'où les noms français de table mêlée, adressage dispersé pour ces techniques d'adressage par contenu ; toutefois, ces deux concepts ne recouvrent pas l'entière signification du mot "Hash-code", c'est pourquoi nous emploierons celui-ci dans ce qui va suivre.

Avant de décrire quelques unes des méthodes les plus courantes, rappelons la difficulté du choix d'une fonction de "Hash-code". Un exemple va illustrer la chose : soit à répartir 31 éléments dans une table de 41 positions : il y a 41^{31} façons de disposer ces 31 éléments, mais seulement $\frac{41!}{10!} = 10^{43}$ solutions distinctes, soit 1 sur 10 millions !

Remarque

La taille des zones mémoires doit être assez grande par rapport à celle de l'argument pour assurer une bonne dispersion ; en pratique, la règle est la suivante :

Si k est le nombre de chiffres de l'argument, alors la taille M requise pour la table est 2^k.