

А.Б. Бессонов
(Уральская государственная
лесотехническая академия)

РАЗРАБОТКА МЕТОДА АДРЕСАЦИИ ДАННЫХ

В настоящее время проблема обеспечения быстрого доступа в область хранения данных остается актуальной при реализации информационных систем.

Большое количество исследовательских работ посвящено разработке алгоритмов доступа. Необходимость исследований в этой области определяется, прежде всего, тем, что эффективность доступа к информации зависит в первую очередь от значения информационных составляющих. Алгоритмы доступа, дающие хорошие результаты на одной совокупности данных, не представляют никакой практической ценности для другой совокупности.

Перспективным направлением исследований является область создания универсальных алгоритмов для любого набора данных.

На практике используются два класса алгоритмов доступа к данным в памяти ЭВМ. Первый класс основан на отображении одного ключевого элемента данных в один адрес памяти, а второй (хеширование) - на отображении N ключей в M адресов физической памяти ($N > M$).

Преимущество алгоритмов хеширования над алгоритмами, принадлежащим к первому классу, заключается в том, что требуется существенно меньше оперативной и внешней памяти ЭВМ.

Разработано множество различных алгоритмов, относящихся

к классу 2, для преобразования ключа в адрес. Основным требованием к таким алгоритмам является равномерность распределения удельной плотности адресов от значения адреса. В действительности же существующие алгоритмы не обеспечивают равномерности получаемого распределения, что приводит к направлению ключей в область переполнения.

Предлагаемое выражение для функции хеширования, можно записать в виде:

где

$$K_j = f_{1j}(a_1) + \dots + f_{nj}(a_n),$$

K_j - значение j -го разряда;

a_i - значение i -го разряда ключа;

f_{ij} - преобразование, применяемое к i -ому разряду ключа при формировании j -го разряда адреса, $i=1, 2, \dots, n$; $j=1, 2, \dots, m$.

Задачу хеширования можно рассматривать как задачу избыточного кодирования. Если рассматривать запись как информацию, определяющую адрес, то можно сказать, что эта информация избыточна. Избыточность может быть устранена кодированием, т.е. однозначным преобразованием одной исходной последовательности в другую, определенной над исходным множеством символов.

Для того, чтобы все значения n -разрядного ключа были равномерно распределены функцией хеширования во все m -разрядные значения адреса, необходимо, чтобы кодовое расстояние в множестве кодов из n разрядов информационной и m разрядов контрольной частей было не менее двух.

Анализ преобразований f_{ij} функции хеширования на основе утверждения приводит к следующему.

Следствие 1. Если f_{ij} - подстановка, то для того, чтобы все значения 2-разрядного ключа распределить равномерно во все значения 2 разрядного адреса, необходимо, чтобы

$$\forall(a, b \in A) \left[\begin{array}{l} f_{11}(a) + f_{12}(a) + f_{21}(a) + f_{22}(a) \neq \\ \neq f_{11}(b) + f_{12}(b) + f_{21}(b) + f_{22}(b) \end{array} \right]$$

Следствие 2. Если f_{ij} - не подстановка, то для того, равномерного распределения всех значений 2-разрядного ключа во все значения 2-разрядного адреса, необходимо, чтобы

$$\forall(a, b \in A) [\forall_j f_{1j}(a) + f_{2j}(a) \neq f_{1j}(b) + f_{2j}(b)]$$

Следствие 3. Для того, чтобы все значения 2-разрядного ключа распределить равномерно в k значений 2-разрядного адреса при $k < 100$, необходимо:

если f_{ij} подстановка, то $k = \frac{100}{l}$, где l - количество пар элементов множества A , для которых выполняется

$$\forall(a, b \in A) \left[\begin{array}{l} a \neq b \rightarrow f_{11}(a) + f_{12}(a) + f_{21}(a) + f_{22}(a) = \\ = f_{11}(b) + f_{12}(b) + f_{21}(b) + f_{22}(b) \end{array} \right],$$

если f_{ij} не подстановка, то $k = r_1 * r_2$, где r_1 и r_2 - количество различных элементов в преобразованиях γ_1 и γ_2 соответственно, а γ_1 и γ_2 определяются выражениями

$$\begin{aligned} \forall(a \in A) [\gamma_1(a) &= f_{11}(a) + f_{21}(a)] \\ \forall(a \in A) [\gamma_2(a) &= f_{21}(a) + f_{22}(a)] \end{aligned}$$

Следствия 1, 2, 3 позволяют динамически, по мере накопления информации о распределении ключей, определять вектора преобразований f_{ij} для обеспечения максимально равномерного распределения адресов от значения адреса.

Использование обобщенного алгоритма кодов для функции хеширования обеспечивает независимость программного обеспечения от вида функции, так как при изменении функции меняются только вектора преобразований. Вектора преобразований зависят от структуры распределения ключей по адресам.