УДК 519.683

МЕТОД УМЕНЬШЕНИЯ ЧИСЛА КОЛЛИЗИЙ ПРИ РАБОТЕ С ХЕШ-ТАБЛИЦАМИ

© Кочемасова Ю.А., Семенова И.В.

Самарский национальный исследовательский университет имени академика С.П. Королева, г. Самара, Российская Федерация

e-mail: ko4emasova2010@yandex.ru

Использование хеширования позволяет организовать эффективный способ хранения и поиска информации. Основными аспектами процесса хеширования являются: хэш-функция, структура хранения данных и метод разрешения коллизий. Поиск новых хеш-функций чаще всего связан с попытками увеличения их безопасности. Наиболее часто используемой структурой данных является хеш-таблица, так как алгоритм поиска в ней в среднем может быть выполнен за O(1). Сократить время, необходимое на вставку и поиск элемента в таблице, можно при помощи эффективного метода разрешения коллизий [1–5].

Авторами предлагается метод, позволяющий минимизировать число коллизий за счет модификации алгоритма хеширования.

Входными данными для предлагаемого алгоритма является строка, для которой необходимо получить хеш-код, а также его длина (минимум 2).

Минимизации числа случаев, когда различным исходным данным будут соответствовать одинаковые хеш-значения, достигается за счет случайной составляющей, добавляемой к исходной строке.

На первом этапе формируется строка, в 2 раза длиннее исходной, путем добавления в конец исходной строки недостающего числа символов. Для определения того, какой символ должен быть дописан на данном этапе, находится сумма кодов текущих двух символов исходной строки (сначала первого и второго, далее второго и третьего и т. д.) и, согласно разработанной таблице сдвигов, осуществляется поиск символа в таблице ASCII, поставленного в соответствие именно этой сумме.

Далее полученная строка сжимается так, чтобы ее длина стала равна величине 2^n, ближайшей к требуемой длине хеш-значения. До тех пор пока не будет достигнуто требуемое уменьшение длины, в строке, полученной на первом этапе, вычисляется сумма кодов символов с номером (center-i), расположенным слева, и (center+i), расположенным справа от центрального символа с номером center, далее, согласно разработанной таблице сдвигов, осуществляется поиск символа в таблице ASCII, поставленного в соответствие полученной сумме, который и записывается в результирующую строку.

Полученная в результате сжатия строка затем преобразуется таким образом, чтобы ее длина стала равна требуемой длине хеш-значения, уменьшенной на 4. Достигается это за счет повторения N раз следующих действий: если номер итерации i кратен N, то в результирующую строку записывается символ, который соответствует сумме i-ого и (i+1)-ого символов сжатой строки, иначе в результирующую строку записывается сам i-й символ, где N вычисляется как отношение количества символов в строке после сжатия к количеству символов, которое нужно убрать для того, чтобы длина стала равна требуемой длине хеш-значения, уменьшенной на 4.

Окончательный результат хеширования формируется путем добавления в конец полученной строки четырех символов, каждый из которых находится в таблице ASCII, согласно разработанной таблице сдвигов, и соответствует сумме кодов всех символов в

исходной строке, длине исходной строки, сумме кодов всех символов в строке, полученной на первом этапе, и длине этой строки.

Между полученными строковыми хеш-значениями и ячейками хеш-таблицы устанавливается взаимно-однозначное соответствие.

Проведенные вычислительные эксперименты показали, что при использовании описанного метода хеширования с увеличением длины хеш-значения количество коллизий стремится к нулю. Так, при длине хеш-значения, равной 5 символам, число коллизий составляет меньше 0,2 % и продолжает уменьшаться с увеличением числа символов в хеш-значении.

Библиографический список

1. Multi-collision resistant hash functions and their applications / I. Berman, A. Degwekar, R.D. Rothblum, P.N. Vasudevan // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2018. № 10821 LNCS. P. 133–161.

2. A novel prime numbers based hashing technique for minimizing collisions / R.K. Bhullar, L. Pawar, V. Kumar, Anjali // Proceedings on 2016 2nd International Conference on Next Generation Computing Technologies. 2017. № NGCT 2016. P. 522–527.

3. Multi-collision resistance: A paradigm for keyless hash functions / N. Bitansky, Y.T. Kalai, O. Paneth // Proceedings of the Annual ACM Symposium on Theory of Computing. 2018. P. 1283–1296.

4. Efficient Hashing technique based on bloom filter for High-Speed Network / G. He, Y. Du, D. Yu // Proceedings 2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics. 2016. № IHMSC 2016. P. 58–63.

5. Zuo P.A., Hua Y. Write-Friendly and Cache-Optimized Hashing Scheme for Non-Volatile Memory Systems // Transactions on Parallel and Distributed Systems. 2018. № 29 (5). P. 985–996.