

1) для снижения процессной погрешности в многоходовых одноядерных ИММ реализовать принцип пакетного разделенного сбора и обработки сигналов, соблюдая соотношение (4);

2) для исключения процессной погрешности строить многоходовые ИММ по схеме многоканальной обработки, решая в каждом канале задачи аналоговой обработки, аналого-цифрового преобразования, первичной и первого уровня вторичной обработки;

3) для уменьшения динамической погрешности использовать многоядерные решения на основе высокопроизводительных микроконтроллеров, обеспечивающие не только высокий уровень параллелизма процессов обработки сигналов, но и сокращение времени сетевого обмена.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Пьявченко О.Н. Проектирование локальных микрокомпьютерных систем. – Таганрог: Изд-во ТРГУ, 2005. – 238 с.
2. Пьявченко О.Н. Особенности построения перспективных многофункциональных интеллектуальных датчиков давления // Известия ЮФУ. Технические науки. Тематический выпуск: «Компьютерные и информационные технологии в науке, инженерии и управлении». – Таганрог: Изд-во ТТИ ЮФУ, 2008. № 11(88). – С.110-121.
3. Пьявченко О.Н., Мокров Е.А., Панич А.Е., Клевцов С.И., Пьявченко А.О., Удод Е.В., Федоров А.Г. Методы, модели, алгоритмы и архитектура прецизионных интеллектуальных датчиков давления / Под ред. д.т.н. профессора О.Н. Пьявченко. – Таганрог: Изд-во ТТИ ЮФУ, 2007. – 130 с.

Пьявченко Олег Николаевич

Технологический институт федерального государственного образовательного учреждения высшего профессионального образования «Южный федеральный университет» в г.Таганроге

E-mail: kafmps@ttpark.ru.

347900, г. Таганрог, ул. Петровская, 81

Тел. +7(8634)328052

Pyavchenko Oleg Nikolaevich

Taganrog Institute of Technological – Federal State-Owned Educational Establishment of Higher Vocational Education «Southern Federal University»

E-mail: kafmps@ttpark.ru

81, Petrovskay street, Taganrog, 347900, Russia

Phone: +7(8634) 328052

УДК 681.3

А.В. Дагаев, Ю.М. Бородянский, О.Д. Глод

ИНФОРМАЦИОННАЯ СИСТЕМА ХРАНЕНИЯ ДАННЫХ

В статье представлены основные понятия хеширования, исследовано поведение характеристик хеш-таблиц в различных алгоритмах хеширования, разработан класс хранения предложений и на основе его создана информационная система. Система позволяет хранить большие объемы информации и выполнять практически мгновенный поиск любых данных независимо от местонахождения в информационном хранилище. Она может применяться в библиотечных, справочных и поисковых системах.

Информационная система; алгоритм; элемент; класс; хеш-таблица; характеристика; хранилище.

A.V. Dagaev, Y.M.Borodyansky, O.D. Glod

INFORMATION SYSTEM OF DATA STORING

The main conception of hashing is presented in the article; also characteristics behavior of hash-tables in different hash algorithm is researched; class of storing sentences is developed and information system is created on the base it. The system allowed saving large information volume and practically momentary carry out the search of any data independently of it location in the information system. The one may be used in the library, inquiry and searching systems.

Information system; algorithm; element; class, hash-table; characteristic; store-house.

Введение

Информационные системы получили широкое развитие в последнее десятилетие. Они применяются практически во всех областях человеческой деятельности: в медицине, авиации, приборостроении, в энергетике, в научно-исследовательских комплексах и т.д. На развитие информационных технологий большое влияние оказало появление Интернет и больших сетей. Их появление также вызвало развитие средств и методов защиты данных. Сегодня методы шифрования применяют: в протоколах передачи данных, в реализации электронной цифровой подписи, для защиты конфиденциальной информации, для ограничения и контроля доступа к автоматизированным устройствам, в средствах идентификации, восстановления информации и т.д. Также методы шифрования применяют в современных СУБД, операционных системах, файловых системах контроля доступа, системах хранения информации. С совершенствованием технических характеристик компьютерной техники становится возможным хранить в оперативной памяти большие объемы информации, требуемые для интерактивной работы.

Представленная в докладе информационная система предназначена для хранения и получения быстрого доступа к данным, содержащимся в хранилище. Она может быть использована для языковых анализаторов, систем хранения больших объемов данных, применяемых, например, в крупных библиотеках, словарях и справочниках и др.

Методы хеширования информации

Сегодня известны десятки методов и алгоритмов хеширования информации, например алгоритмы: Рабина-Карпа, Кнута-Морриса-Пратта, Бойера-Мура [1,2] и др. Информацию по методам криптографии и хешированию можно встретить в большом числе книг, например в [4,5].

В результате работы алгоритмов на выходе получается уникальное хеш-число, необходимое для идентификации некоторой информации. На основе хеш-числа возможно создание специализированных таблиц, позволяющих хранить объемы данных, ограниченные только размером оперативной памяти. Применение хеш-таблиц стало возможно с начала 1990-х годов, вследствие появления новых мощных ЭВМ. Хеш-таблицы бывают открытые и закрытые. Размер закрытой таблицы не может быть изменен, у открытой таблицы размер может изменяться в зависимости от роста объема данных, содержащихся в ней.

В разработанной информационной системе применялась закрытая хеш-таблица, хотя с точки зрения авторов [6] данная схема хеширования является многомерной, что обусловлено неограниченным размером списков. В качестве приме-

ра было взято сто тысяч пар предложений и на их основе сделано информационное хранилище. Информационное хранилище представляет собой таблицу, содержащую цепочки с парами предложений, что отображено на рис. 1.

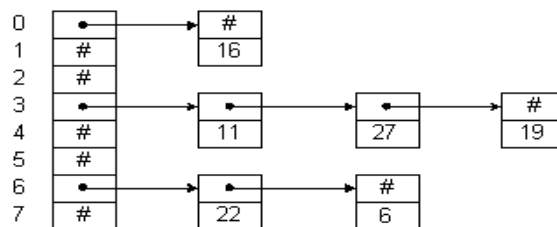


Рис. 1. Хеш-таблица

Ее можно рассматривать как совокупность связанных списков. Информация, хранящаяся в одной цепочке имеет одинаковый индекс. Индекс – это уникальное число однозначно идентифицирующее информацию. В данном случае индексом также является номер строки в таблице. Индекс вычисляется по следующей формуле: $ind = hash(Sent) \bmod S$, где $Sent$ – входная строка, $hash(Sent)$ – уникальное хеш-число; S – размер хеш-таблицы (простое число), \bmod – остаток от целочисленного деления, которым будет являться индекс. Можно усложнить процесс генерации хеш-числа, если воспользоваться идеей Дж. Картера и М. Вегмана: если ключ $Sent$ – составной (состоит из нескольких слов $x_1 \dots x_s$), то хеш-число может быть получено путем конкатенации хеш-чисел всех слов, входящих в предложение: $hash(Sent) = (hash_1(x_1) + hash_2(x_2) + \dots + hash_s(x_s)) \bmod S$.

Рассмотрим характеристики хеш-таблицы:

- 1) размер хеш-таблицы;
- 2) среднее время поиска элемента в таблице;
- 3) среднее время поиска информации в вырожденной таблице;
- 4) максимальная длина цепочки в строке таблицы;
- 5) средняя длина цепочки в строке таблицы;
- 6) число элементов, содержащихся во всех цепочках;

Размер хеш-таблицы – это количество содержащихся в ней строк. Значение размера хеш-таблицы должно быть простым числом, вследствие того, что мы должны получать как можно меньше повторов в индексах предложений, а остаток от деления на простое число будет являться индексом предложения. Размер хеш-таблицы должен быть достаточно большим, чтобы в ней оставалось большое число пустых ячеек, для возможности дополнения их новыми данными. Чем меньше таблица, тем больше среднее время поиска данных (ключа) в ней. Закрытые хеш-таблицы эффективны, когда максимальный размер входящего потока данных известен, и можно создать таблицу с намного большим количеством элементов. Доказано, что, если только закрытая таблица становится полной более чем на 50 процентов, то производительность ее резко падает.

По мере того, как таблица растет, увеличивается количество списков и, соответственно, среднее число узлов в каждом списке уменьшается. Пусть количество элементов равно n . Если размер таблицы равен 1, то таблица вырождается в один список длины n и получается вырожденная таблица. Если размер таблицы равен 2 и хеширование идеально, то мы будем работать с двумя списками по $n/2$

элементов в каждом. Следовательно, с увеличением размера таблицы, до определенного размера длина списков уменьшается. Если брать размер таблицы в десятки раз превышающий количество заполненных строк, то неэффективно будет использоваться оперативная память, поскольку будет храниться много пустых ячеек, а если брать размер таблицы, соизмеримый с количеством заполненных строк, то время поиска будет слишком большим и неприменимо для интерактивной работы с информационной системой. Следовательно, можно выбрать такой оптимальный размер хеш-таблицы, который позволит получать быстрый доступ к требуемой информации. Для использования в информационной системе алгоритма хеширования были рассмотрены различные методы хеширования, например: CRC32, CRC32B, GOST, HAVAL256, MD2, MD4, MD5, RIPEMD160, SHA256, TIGER192 и другие. При исследовании характеристик таблиц для разных алгоритмов было выяснено, что максимальная длина цепочки при принятых размерах таблицы составляет 58 элементов. Данная величина практически не зависит от алгоритма, а только от размера таблицы. На рис.2 показано, каким образом меняется количества элементов в цепочках при применении различных алгоритмов хеширования.

Цепочкой является список, содержащий больше одного элемента. Каждый элемент содержит пару предложений и указатель на следующий элемент. Чем меньше число элементов в цепочках, тем быстрее будет происходить поиск информации в хеш-таблице. Из рис.2. видно, что наилучшим из представленных методов для поиска данных является алгоритм MD4. Похожим на представленный является график изменения количества цепочек от применяемого метода. Можно отметить, что оно изменяется в пределах от 3903 для алгоритма MD4 до 4074 для алгоритма HAVAL160.

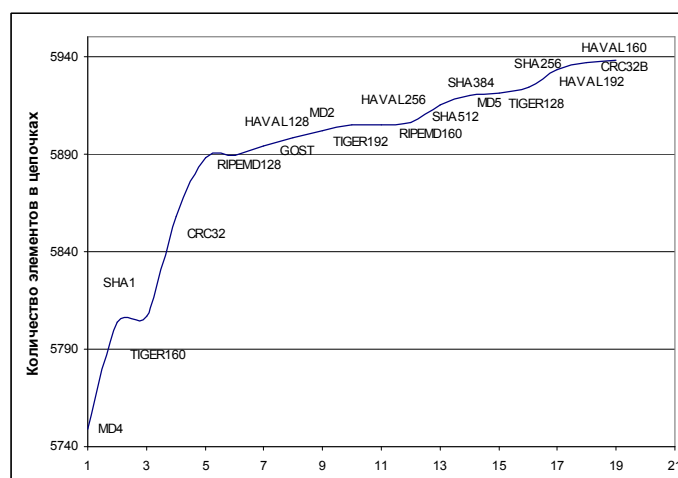


Рис.2. Количество элементов в цепочках в разных алгоритмах шифрования

В ходе исследования был проведен анализ изменения общего количества элементов в цепочках от размера хеш-таблицы (рис.3).

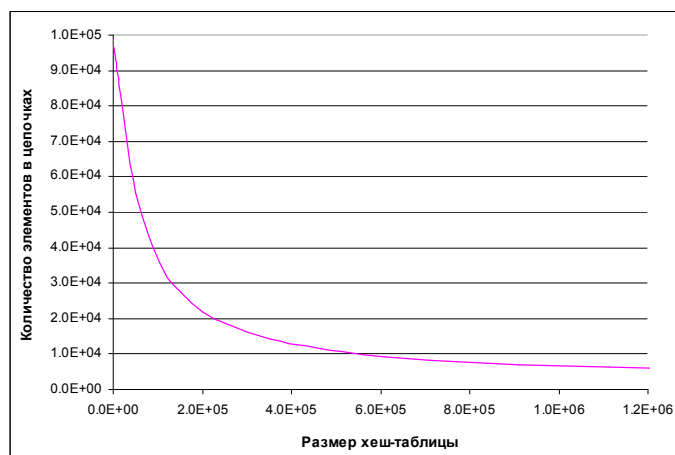


Рис.3. Изменение количества элементов в ячейках от размера таблицы

При моделировании использовался один из наиболее распространенных алгоритмов – CRC32 (циклически избыточный код). Алгоритм CRC32 предназначен для вычисления 4-байтового контрольного кода любой последовательности байтов, он применяется в частности в архиваторах PKZIP и ARJ, при проверке правильности передаваемых данных, подсчете контрольных сумм файлов т.д.

Как видно из рис.3, при небольших размерах хеш-таблицы количество элементов, содержащих в цепочках очень большое, а при размерах таблицы в несколько ячеек, оно стремится к общему числу элементов. При увеличении размера таблицы количество цепочек и соответственно элементов в них уменьшается и при размере таблицы, в несколько раз превышающем количество элементов, оно будет слабо изменяться. То есть для ста тысяч пар предложений вполне достаточно таблицы в 300 тысяч ячеек, а для двухсот тысяч – шестьсот тысяч ячеек. Дальнейшее увеличение размера таблицы приведет в конечном состоянии к числу, при котором количество элементов в списках будет равно нулю, но размер таблицы при этом может достигать несколько миллионов. Оперативная память при этом будет использоваться неэффективно, поэтому на практике приходится использовать размер таблицы, в три – четыре раза превышающий количество элементов. Можно отметить, что график времени поиска предложения от размера хеш-таблицы будет иметь вид, изображенный на рис.2.

Однако с увеличением скорости работы оперативной памяти он может иметь более пологий характер. Так был проведен эксперимент при размере таблицы в миллион ячеек и количестве предложений - 100 тысяч. Общее время поиска всех предложений при этом составило 2,5 секунды, размер таблицы с предложениями, хранящейся на диске, занял 50 мегабайт. Понятно, что на поиск одного предложения требуется гораздо меньшее время. Поэтому если все предложения хранятся в оперативной памяти, то время поиска предложений будет слабо зависеть от размера хеш-таблицы. Если перед поиском таблица будет загружаться с жесткого диска в оперативную память, то общее время поиска станет больше и если хранить информацию в отдельной базе данных, то время поиска еще увеличится. Рассмотрим график, представленный на рис.4.

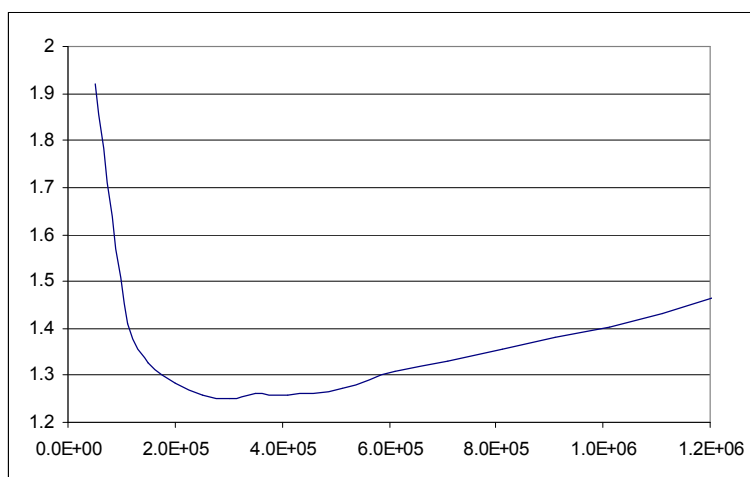


Рис.4. Зависимость количества цепочек от размера таблицы

Также для определения минимального размера таблицы, достаточного для хранения заранее заданного числа предложений, может служить характеристика средней длины цепочки (рис.5). При минимальном размере таблицы средний размер цепочки совпадает с размером цепочки, далее с увеличением размера таблицы он уменьшается, достигая размера 1-2 предложения на одну цепочку. Как видно из рис.5, при размере таблицы в 300 тысяч ячеек достигается минимум средней длины цепочки, он равен 1,25. При дальнейшем увеличении размера таблицы наблюдается незначительное увеличение среднего размера цепочки, что обусловлено тем, что скорость уменьшения цепочек больше скорости уменьшения числа содержащихся в них предложений. Это также говорит о небольшом числе длинных цепочек при большом размере таблицы, которое зависит от качества алгоритма хеширования и качества реализации его моделирования.

Разработка информационной системы

Информационная система была разработана в среде Delphi, под ОС Windows 2000/XP/2007. Для эффективной работы с информацией (парами предложений) был разработан специальный класс, оформленный в отдельном модуле. Он позволяет делать ряд следующих операций: загрузка неотформатированного текста из файла, разделенного точками, и перевод его в хранилище; сохранение хранилища в текстовом формате в кодировке UTF-16,32 со специальными разделителями; перевод текста из формата анти-строки в формат юникод-строки; генерация хеш-числа и индекса на основе пары предложений; очистка хеш-таблицы; загрузка информации из хранилища, находящегося на жестком диске; проверка на занятость строки таблицы и добавление пары предложений; получение индекса предложения; получение предложения по индексу; удаление пары предложений; получение количества пар предложений, содержащихся в таблице и др.

Приложение позволяет загружать текстовый пользовательский файл в верхнюю таблицу, загружать хранилище в нижнюю таблицу, редактировать и удалять предложения из хранилища, переносить предложения из пользовательского файла в хранилище, удалять информацию из пользовательского файла, выполнять поиск нужных данных как в пользовательском файле, так и в хранилище.

В ходе разработки приложения были проведены следующие мероприятия:

1. Рассмотрен ряд алгоритмов хеширования и проведено их исследование для задачи хранения предложений.
2. Показано, что при размере таблицы, в три раза превышающей количество хранящихся элементов, достигается минимальный средний размер цепочки.
3. Показано, что при размере таблицы, в десять раз превышающей количество хранящихся элементов, достигается барьер в четыре процента количества цепочек, при этом количество предложений в цепочках не превышает 6 процентов.
4. Разработан класс для хранения пар предложений, на базе его создано приложение, позволяющее заносить, редактировать и удалять данные из хранилища и пользовательского файла с парами предложений.
5. Разработанная программа и класс могут быть использованы в ряде задач хранения больших объемов информации: в различных СУБД, лингвистических, исследовательских, библиотечных и справочных системах - там, где время поиска требуемой информации должно быть минимально.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Томас Х. Кормен и др.* Глава 32. Поиск подстрок // Алгоритмы: построение и анализ = INTRODUCTION TO ALGORITHMS. – 2-е изд. – М.: «Вильямс», 2006. – С. 1296.
2. Karp and Rabin's original paper: Karp, Richard M.; Rabin, Michael O. (March 1987). «Efficient randomized pattern-matching algorithms». IBM Journal of Research and Development 31 (2), 249—260.
3. HashLib! 1.03 (C) Alex Demchenko, 2002, Moldova, Chishinev —библиотека, алгоритмов хеширования.
4. *Шнайдер Б.* Прикладная криптография.
5. *Щербаков Л.Ю., Домашен А.В.* Прикладная криптография.

Дагаев Александр Владимирович

Технологический институт федерального государственного образовательного учреждения высшего профессионального образования «Южный федеральный университет» в г. Таганроге

E-mail: adagaev@list.ru

347928, Таганрог, пер. Некрасовский, 44

Тел.: +7(8634)371787

Бородянский Юрий Михайлович

Технологический институт федерального государственного образовательного учреждения высшего профессионального образования «Южный федеральный университет» в г. Таганроге

E-mail: borodyansky@yandex.ru

Глод Ольга Денисовна

Технологический институт федерального государственного образовательного учреждения высшего профессионального образования «Южный федеральный университет» в г. Таганроге

E-mail: glod@tsure.ru

Dagaev Alexander Vladimirovich

Taganrog Institute of Technological – Federal State-Owned Educational Establishment of Higher Vocational Education «Southern Federal University»

e-mail: adagaev@list.ru

44, Nekrasovskiy, Taganrog, 347928, Russia

Phone: +7(8634)371787

Borodyansky Yuriy Mihailovich
Taganrog Institute of Technological – Federal State-Owned Educational Establishment
of Higher Vocational Education «Southern Federal University»

E-mail: borodyansky@yandex.ru

Glod Olga Denisovna

Taganrog Institute of Technological – Federal State-Owned Educational Establishment
of Higher Vocational Education «Southern Federal University»

E-mail: glod@tsure.ru

УДК 621.315.434.001

В.Л. Земляков, С.Н. Ключников, А.И. Кулинич

ОПРЕДЕЛЕНИЕ ПЬЕЗОМОДУЛЯ ПЬЕЗОКЕРАМИЧЕСКИХ МАТЕРИАЛОВ НА ОБРАЗЦАХ ЭЛЕМЕНТОВ В ФОРМЕ ДИСКА

Получены приближенные формулы, позволяющие уменьшить объем первичных измерений и упростить процедуру определения пьезомодуля пьезокерамических материалов в динамическом режиме на образцах элементов в форме диска.

Пьезокерамические материалы; пьезомодуль; диск; коэффициент Пуассона; аппроксимации; метод наименьших квадратов.

V.L. Zemlyakov, S.N. Kliuchnikov, A.I. Kulinich

THE PIEZOMODULE DETERMINATION FOR PIEZOCERAMIC MATERIALS ON ELEMENT SAMPLES IN THE FORM OF DISK

The approximate formulas which allow to reduce the amount of initial measurements and to simplify the piezomodule determination of piezoceramic materials in dynamic mode on element samples in the form of disk are derived.

Piezoceramic materials; piezomodule; disk; Poisson's ratio; approximation; least-squares method.

Для определения параметров пьезокерамических материалов чаще всего используют пьезокерамические элементы (ПКЭ) в форме столбика, стержня и диска. Для первых двух типов ПКЭ измерения осуществляются достаточно просто. Для элементов в форме диска ситуация более сложная, поскольку необходимо дополнительно проводить измерения частоты первого обертона f_{p1} (отношение частоты этого обертона к частоте резонанса f_{p1}/f_p обычно лежит в интервале 2,54 – 2,57), затем рассчитывать коэффициент Пуассона σ , а также находить наименьший корень частотного уравнения, содержащий функции Бесселя нулевого и первого порядков по таблицам. Формула для расчетов получается громоздкой и не удобной для инженерных расчетов.

В работе [1] уже рассматривалась возможность приближенного вычисления коэффициента электромеханической связи пьезоматериала на образцах ПКЭ в форме диска. Другим важным параметром является пьезомодуль. Возможность его определения по приближенным формулам и рассматривается в данной работе.

Будем использовать следующие исходные соотношения [2]:

$$\sigma = \frac{5,33 - 1,86(f_{p1}/f_p)}{0,633(f_{p1}/f_p) - 0,2}; \quad (1)$$