
ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

3. Krispin L. Gibkoe testirovanie. Prakticheskoe rukovodstvo dlya testirovshikov PR i gibkih komand / L. Krispin, J. Gregori. – M. : Vil'yams, 2010. – 464 p.
4. Litovkin D. V. Biblioteka modul'nogo testirovaniya, ispol'zuemaya pri obuchenii programmirovaniyu / D. V. Litovkin, O. A. Sychev // Izvestiya VolgGTU : mezhvuz. sb. nauch. st. – Volgograd, 2010. – Vyp. 9, № 11. – P. 106–109. – (Ser. Aktual'nye problemy upravleniya, vychislitel'noi tekhniki i informatiki v tekhnicheskikh sistemah).
5. Shabalina O. A. Obuchenie razrabotchikov programmnoho obespecheniya: primenenie komp'yuternykh igr i processa ih razrabotki / O. A. Shabalina, A. V. Kataev, P. N. Vorobkalov // Izv. VolgGTU : mezhvuz. sb. nauch. st. – Volgograd, 2010. – Vyp. 9, № 11. – P. 117–124. – (Ser. Aktual'nye problemy upravleniya, vychislitel'noi tekhniki i informatiki v tekhnicheskikh sistemah).
6. Shabalina O. A. Razrabotka obuchayushih komp'yuternykh igr dlya ispol'zovaniya v vuze / O. A. Shabalina, P. N. Vorobkalov, A. V. Tarasenko, A. V. Kataev // Kachestvo. Innovacii. Obrazovanie Evropeiskii centr po kachestvu. – 2008. – № 4. – P. 14–16.
7. Sinitzyn S. V. Verifikaciya programmnoho obespecheniya / S. V. Sinitzyn, N. Yu. Nalyutin. – M. : Binom, 2008. – 368 p.
8. Vorobkalov P. N. Ocenka kachestva elektronnykh obuchayushih sistem / P. N. Vorobkalov, V. A. Kamaev // Upravlenie bol'shimi sistemami : sb. tr. – 2009. – № 24. – P. 99–111.

УДК 004.428.4

ДЕМОНСТРАЦИЯ ПРОЦЕССА ПОИСКА ИНФОРМАЦИИ С ИСПОЛЬЗОВАНИЕМ ХЕШИРОВАНИЯ

Смирнова Марина Олеговна, кандидат педагогических наук, Астраханский государственный университет, 414056, Россия, г. Астрахань, ул. Татищева, 20а, e-mail: apsmir@yandex.ru.

Представлен программный продукт, с помощью которого можно продемонстрировать применение хеширования, являющегося самым быстродействующим методом программного поиска. Хеширование применяется при работе с наборами данных большого размера (браузеры, словари, компиляторы и т.п.) и в криптографии.

Дано описание основных компонентов программного продукта, реализованного на языке программирования Object Pascal в среде визуального программирования Delphi 7, и возможностей использования при изучении разделов, связанных с алгоритмами поиска, построенных на основе хеш-таблиц. Моделирование процесса поиска информации с применением хеш-таблиц в данном программном продукте реализовано на примере построения телефонного справочника, при этом используются таблицы с закрытой и открытой адресацией как примеры открытого и закрытого хеширования.

Типами данных, на которых основана реализация алгоритмов, являются массивы и записи, предназначенные для хранения как самих данных, так и ключей к ним. При этом первый тип хеширования строится на взаимосвязи массива и односвязных списков. Односвязные списки реализуются с помощью указателей на записи. Каждому ключу, хранящемуся в массиве, соответствует указатель на свой список с данными. Второй тип хеширования целиком строится на использовании массива, предположительно имеющего элементов больше возможного количества данных и замкнутого в кольцо.

Программный продукт снабжен справочной системой, позволяющей познакомиться с основами хеширования и принципами работы самой программы.

Разработанный программный продукт обеспечивает наглядными материалами поддержку тем, связанных с алгоритмами поиска, построенных на основе хеш-таблиц.

Ключевые слова: демонстрационная программа, хеширование, поиск информации, программный код, односвязные списки, массив, запись, ключ, указатель, типы данных.

DEMONSTRATION OF INFORMATION RETRIEVAL USING HASHING

Smirnova Marina O., Cand. in Pedag., Astrakhan State University, 20a Tatishchev str., Astrakhan, 414056, Russia, e-mail: mosmir1@yandex.ru.

This article presents a software product that demonstrates the application of hashing, which is the corresponding fast-method of software search. Hashing is used in working with large data sets (browsers, dictionaries, compilers, etc.) and in cryptography.

A description of the main components of the software, implemented in the programming language Object Pascal in a visual programming environment, Delphi 7, and the possibilities for the study of topics related to search algorithms that are based on hash tables are given. The modeling of the process of finding information using hash tables in this software product is implemented on the building of the phone directory with using a table for indoor and outdoor addressing, as examples of open and closed hashing.

Data types on which the implementation of algorithms is realised, are arrays and records designed to store both data and keys for them. In the addition this first type of hash is based on the relationship between the array and the singly-linked lists. Singly-linked lists are implemented using pointers to records. Each key stored in the array corresponds to a pointer to its list of the data. The second type of hash is based entirely on the use of the array, that presumably, has more elements than the possible amount of data, and is closed in a ring.

The software product is provided with help system, which allows to get acquainted with the basis of hashing and principles of the program.

The software product provides the support of topics that are related to algorithms of search based on hash tables with visual materials.

Key words: demonstration program, hashing, searching for information, program code, singly-linked lists, array, record, key, pointer, data types.

Хеширование является самым быстродействующим из известных методов программного поиска. В связи с этим изучение процесса хеширования является одной из важных составляющих при подготовке специалистов в области информационных технологий и защиты информации.

Оказать существенную помощь при изучении хеширования информации и ее поиска может разработанная демонстрационная программа «Хеш-таблицы и их применение».

Моделирование процесса поиска информации с применением хеш-таблиц в данном программном продукте реализовано на примере построения телефонного справочника. Справочник строится на основе таблиц с закрытой и открытой адресацией (открытое и закрытое хеширование) [1–3, 4, 5, 7, 8]. Ввод данных в справочнике можно осуществлять вручную и из текстового файла; также имеется возможность осуществлять поиск информации по выбранному или введенному номеру. Наличие системы справочного материала позволяет получить общее представление о хеш-таблицах, способах хеширования и принципах работы демонстрационной программы.

Данная демонстрационная программа написана на языке Object Pascal в среде визуального программирования Delphi 7 [6]. Структурно программа состоит из трех форм, несущих различную функциональную нагрузку. С каждой формой связан программный модуль.

ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ

В модуле рабочей формы, помимо объявления класса формы с ее компонентами, важное значение имеет описание записи и типа-указателя на нее:

```
type elem=^telephone;  
telephone=record  
    tel: longint;  
    fam, name, otch: string[50];  
    street: string[50];  
    build, flat: byte;  
    next: elem;  
end;
```

Для построения справочника требуется выбрать способ его заполнения: вводить элементы вручную или брать их из заранее заготовленного файла. Этой цели служит элемент `RadioGroup1`, позволяющий осуществить такой выбор.

Если выбран ручной способ ввода, то указывается количество записей в поле редактора `Edit3` и данные для справочника вносятся в таблицу `StrGrid1`. С помощью кнопки `Button1` введенные данные помещаются в переменную-указатель `q` на `elem`, для которой перед этим выделяется динамическая память. Одновременно телефонный номер (`StrGrid1.Cells[0,1]`) заносится в список `ComboBox1`. Данный процесс повторяется до тех пор, пока не будет внесено указанное количество записей.

В случае когда выбран ввод из файла, организуется открытие файла, содержащего исходные данные, а с помощью кнопки `Button1` запускается цикл, организующий считывание строк из файла и размещение их в соответствующие поля той же переменной-указателя `q`. Цикл работает, пока не исчерпаются все строки в файле, после чего файл закрывается.

Кроме выбора способа ввода данных, есть еще выбор типа хеш-таблицы, используемой для построения телефонного справочника. Если выбрано открытое хеширование (с помощью цепочек), то таблица будет строиться в динамической памяти при помощи одностороннего списка.

В основе списка лежит массив `tab`, элементами которого являются данные типа `elem`. Количество элементов в нем равно 10 (определяется числом вариантов окончания телефонного номера – его последней цифрой). Первоначально все элементы массива хранят нулевой адрес. Наличие в элементе такого адреса говорит о том, что с ним не связан никакой список с данными справочника.

Для вновь введенной записи справочника номер элемента в массиве `tab` определяется по последней цифре телефонного номера.

```
i:=q^.tel mod 10;  
q^.next:=tab[i];  
tab[i]:=q;
```

Далее запись, хранящаяся в переменной `q`, должна быть вставлена в голову односвязного списка, указатель на который хранится в *i*-ом элементе массива `tab`. Для этого адрес из этого элемента копируется в адресное поле переменной `q`, и тем самым новый элемент встает в голову списка. Остается только в *i*-ый элемент массива поместить адрес вновь созданного элемента. Этот процесс повторяется, пока поступают новые данные для справочника. В итоге массив `tab` становится набором адресов для созданных списков данных.

Если выбрано закрытое хеширование (с помощью открытой адресации), то таблица будет строиться на основе одного массива `datatab`, где размещаются все данные телефонного справочника, который будет построен.

При ручном вводе данных количество записей в справочнике заранее известно (количество строк списка `comboBox1`), поэтому остается только рассчитать количество элементов массива-справочника (хеш-таблицы) `datatab`:

```
m:=comboBox1.Items.Count;  
m:=(m+n)*3 div 2;
```

Количество должно быть таковым, чтобы места хватило для всех данных (а может быть, и с запасом).

Рассмотрим теперь построение хеш-таблицы, т.е. заполнение массива `datatab`. Для его заполнения используется дополнительный массив `htab`, который хранит признак заполнения требуемого элемента массива `datatab`.

Первоначально весь массив `htab` заполнен значением «ложь», это означает, что соответствующий элемент массива данных является свободным. Полученный с данными номер телефона служит основой для определения индекса элемента массива-справочника, который должен хранить этот номер:

```
readln(fin,st);  
num:=StrToInt(st);  
i:=num mod m;
```

Однако может случиться так, что данный элемент массива уже занят, так как раньше поступил номер с таким же признаком, поэтому, прежде всего, проверяется значение элемента массива `htab[i]`.

Если значение этого элемента «ложь», то необходимый элемент свободен и его можно заполнять, при этом элемент таблицы `htab[i]` получает значение «истина».

Если же элемент с найденным номером оказался занятым (`htab[i]=true`), то рассматривается элемент со следующим номером:

```
l:=i+1;  
while htab[l] and (l<m) do l:=l+1;  
if l=m then  
begin  
l:=0;  
while htab[l] and (l<i) do l:=l+1;  
end;
```

и снова идет проверка на возможность вставки данных в массив-справочник. Если и в него нельзя внести данные, то рассматривается следующий элемент и так далее, пока не найдется свободный элемент или не будет достигнут конец массива-списка (рассматриваются, конечно, элементы массива признаков `htab`). Если достигнут конец массива, а свободного места не нашлось, то процесс просмотра начинается с головы массива и до того места, с которого и начинался весь просмотр. Когда будет занесен последнее значение – хеш-таблица построена.

Процесс поиска информации в сформированном телефонном справочнике начинается с получения телефонного номера, данные для которого требуется найти. Это осуществляется с помощью списка номеров `ComboBox1`:

```
if ComboBox1.Text<>"" then  
begin  
num:=StrToInt(ComboBox1.Text);  
Edit1.Text:="";  
Edit2.Text:="";  
end;
```

Если в списке номер выбран, то значение поля `Text` элемента `ComboBox1` изменится и перестанет быть пустым, тогда переменная `num` получит значение этого поля, переведенное в числовой формат.

Процесс поиска информации в справочнике начинается при щелчке по кнопке Button2 и тоже зависит от выбранного способа хеширования. В случае выбора первого способа – открытого хеширования – определяется индекс элемента таблицы tab, и из этого элемента в переменную-указатель q заносится адрес первого элемента данной ветви списка $q:=tab[i]$. Здесь переменная i используется как индекс хеш-таблицы. Это первая часть вспомогательной информации, определяемой в процессе поиска.

После того как нужная позиция в хеш-таблице определена, начинается поиск в найденной ветви динамического списка. С помощью цикла while осуществляется движение по списку и на каждом его шаге производится проверка выбранного телефонного номера и номера просматриваемого элемента.

После завершения работы цикла требуется установить, при каких обстоятельствах закончилась его работа. Для этого анализируется значение переменной-флага. В том случае, когда значение флага осталось нулевым, делается вывод, что указанного телефонного номера в списке нет, и сообщается об этом.

Если флаг имеет значение 1, то это означает, что совпадение найдено и цикл досрочно завершил свою работу. В этом случае в таблицу выводятся сведения о владельце указанного телефона.

В случае выбора второго способа (закрытого хеширования) также определяется индекс элемента таблицы, но уже $htab: i:=num \bmod m$. Если значение элемента $htab[i]$ – «истина» и произошло совпадение выбранного (или введенного) номера с номером в поле tel элемента $datatab[i]$, то производится вывод данных о владельце телефона в таблицу StrGrid1.

Если же элемент таблицы $htab[i]$ имеет значение «истина», но совпадения выбранного (или введенного) номера с номером в поле tel элемента $datatab[i]$ не произошло, то делается вывод о том, что это место занято другим номером с тем же индексом и следует искать истинное расположение требуемой записи в списке. Поиск нужной записи в списке организуется по тому же принципу, что и поиск места для записи данных. Если поиск не увенчался успехом, то также выводится сообщение об отсутствии номера в списке телефонной книги, если же поиск удачен, то выводятся найденные данные телефонного справочника.

Осталось отметить, что в программе имеется третья форма, предназначенная для отображения информации, связанной с хешированием, и справочной информации.

Таким образом, разработанный программный продукт обеспечивает наглядными материалами поддержку тем, связанных с алгоритмами поиска, построенных на основе хеш-таблиц.

Список литературы

1. Ахо А. Структуры данных и алгоритмы / А. Ахо, Дж. Хопкрофт, Дж. Ульман. – М. : Вильямс, 2007. – 340 с.
2. Вирт Н. Алгоритмы + структуры данных = программы / Н. Вирт. – М. : Мир, 1985. – 227 с.
3. Кнут Э. Искусство программирования / Э. Кнут. – М. : Вильямс, 2007. – Т. 2: Получисленные алгоритмы. – 278 с.
4. Керниган Б. Практика программирования / Б. Керниган, Р. Пайк. – СПб. : Невский диалект, 2006. – 340 с.
5. Кормен Т. Алгоритмы: построение и анализ / Т. Кормен, Ч. Лейзерсон, Р. Ривест. – М. : МЦНМО, 2001. – 321 с.
6. Культин Н. А. Delphi 7 программирование на Object Pascal / Н.А. Культин. – СПб. : Невский диалект, 2007. – 370 с.
7. Левитин Г. Н. Алгоритмы. Введение в разработку и анализ / Г. Н. Левитин. – М. : Вильямс, 2008. – 287 с.
8. Чмора А. Современная прикладная криптография / А. Чмора. – М. : Гелиос АРВ, 2005. – 232 с.

References

1. Akho A. Strukturi dannih i algoritmi / A. Akho, J. Hopcroft, J. Ulman. – M. : Viliams, 2007. – 340 p.

2. Chmora A. Sovremennaiia prikladnaia kriptografiia / A. Chmora. – M. : Gelios ARV, 2005. – 232 p.
3. Kernigan B. Praktika programmirovaniia / B. Kernigan, R. Paik. – SPb. : Nevskii dialect, 2006. – 340 p.
4. Knut A. Iskusstvo programmirovaniia / A. Knut. – M. : Viliams, 2007. – T. 2: Poluchislennii algoritmi. – 278 p.
5. Kormen T. Algoritmi: postroenie i analiz / T. Kormen, Ch. Leizerson, R. Rivest. – M. : MCNMO, 2001. – 321 p.
6. Kultin N. A. Delphi 7 programmirovanie na Object Pascal / N. A. Kultin. – SPb. : Nevskii dialect, 2007. – 370 p.
7. Levitin G. N. Algoritmi. Vvedenie v razrabotku i analiz / G. N. Levitin. – M. : Viliams, 2008. – 287 p.
8. Virt N. Algoritmi + structuri dannih = programmi / N. Virt. – M. : Mir, 1985. – 227 p.

УДК 004.652.5

ПРИМЕР РАБОТЫ АЛГОРИТМА ТРАНСФОРМАЦИИ КУБА ДАННЫХ С ИСПОЛЬЗОВАНИЕМ МАТРИЦ ТРАНСФОРМАЦИИ

Спандерашвили Дмитрий Викторович, кандидат технических наук, Астраханский государственный университет, 414056, Россия, г. Астрахань, ул. Татищева, 20а; начальник отдела организации эксплуатации систем коммутации и сетей доступа Астраханского филиала ОАО «Ростелеком», 414000, Россия, г. Астрахань, Театральный пер., 7/8, e-mail: spanderashvili@gmail.com.

С течением времени в многомерных структурах данных могут происходить изменения. Для проведения достоверного анализа данных необходимо отслеживать все изменения в структуре и элементах измерений. В статье в краткой форме дано формальное описание механизма трансформации куба данных из одной структурной версии в другую, для обеспечения возможности проводить анализ данных из куба данных, претерпевшего структурные изменения. В первой части статьи дается описание механизма использования матриц трансформации для приведения куба данных к заданной структурной версии, а во второй части приводится пример использования алгоритма трансформации на тестовых данных. В краткой форме в статье дано формальное описание механизма темпорального хранения данных. Сформулированы предпосылки применения темпорально-многомерного хранилища данных, вводится формальное описание темпорально-многомерного хранилища данных, определяется механизм формирования соответствия между структурными версиями, описан механизм применения матриц трансформации. В качестве примера работы алгоритма трансформации куба данных из одной структурной версии в другую рассматривается тестовый пример, в котором требуется сравнить данные куба одной структурной версии с данными куба данных из другой структурной версии. Путем предлагаемого метода проводятся преобразования, благодаря которым такое сравнение становится возможным.

Ключевые слова: *многомерные структуры данных, изменения, анализ данных, измерение, трансформация, куб данных, структурная версия, матрицы трансформации, алгоритм трансформации, темпорально-многомерные данные.*