АВТОМАТИЗАЦИЯ ПРОЕКТИРОВАНИЯ В МИКРОЭЛЕКТРОНИКЕ.
ТЕОРИЯ, МЕТОДЫ, АЛГОРИТМЫ
(Вычислительные системы)
1978 год Выпуск 77

УДК 519.688

О ПОСТРОЕНИИ ОБОБЩЕННОЙ ПАМЯТИ R-МАШИНЫ ДЛЯ ЗАДАЧ ОБРАБОТКИ ТЕКСТОВОЙ ИНФОРМАЦИИ

Ю.Г.Косарев, Н.А.Чужанова

I. Реализация в ЭВМ моделей специализированных памятей, таких как стек, регистр, список и т.п., существенно обогащает арсенал программных средств. Более того, как показано, например, в [I], могут быть созданы наборы из сравнительно небольшого числа памятей, которых оказывается достаточно для реализации любого процесса преобразования данных. По-видимому, в наиболее последовательной и доведенной до практической реализации форме данная концепция представлена в работах И.В.Вельбицкого [I-3]. При таком подходе процесс программирования сводится к последовательному применению памятей из некоторого фиксированного набора. При этом сначала выбирается подходящая память R-машины, структура и набор операций которой соответствует структуре обрабатываемых данных. Далее происходит погружение этих данных в выбранную память, которое заключается в присвоении формальным параметрам памяти конкретных (фактических) значений и фиксации конкретных параметров структуры данных. Затем происходит собственно программирование, которое заключается в указании последовательности операций над данной памятью. При этом программист избавляется либо от необходимости запоминать адреса первого и последнего элементов массива при работе в режиме последовательного доступа (как это имеет место в стековых, регистровых, счетчиковых и т.п. типах памятей), либо от необходимости образовывать адрес методами ассоциативного кодирования при работе в режиме прямого доступа (например, табличные памяти).

Вместе с тем есть задачи, когда над одними и теми же совокупностями данных необходимо выполнять последовательность операций,

определенных над памятями различных типов*). Нередко к одним и тем же данным нужно обращаться то в режиме последовательного, то в режиме прямого доступа.

Цель данной работы - попытаться расширить арсенал программных средств путем введения некоторой памяти, которая сочетала бы в себе (обобщала) возможности стековой, бобслей, регистровой, счетчиковой, табличной и списковой памятей (далее они называются простейшими).

Вполне понятно, что создание такой обобщенной памяти потребует несколько иного подхода, чем при использовании простейших памятей. Погружение данных в простейшие памяти предполагает полное соответствие между структурами данных и памяти, в которую они погружаются. При этом допустимы в с е операции, определенные для данной памяти. В отличие от этого, для обобщенной памяти принятие такого же требования нецелесообразно, так как ограничило бы ее применение лишь небольшим кругом структур данных. Поэтому приходится вводить описание свойств структур данных и указывать для каждого сочетания свойств допустимый набор операций над этими данными.

В соответствии с этим в работе выделяется тот необходимый набор свойств структур данных, которые полностью определяют допустимый набор операций, приводится набор операций, способ описания памяти.

2. Каждая из простейших памятей может работать с данными определенной структуры. Без потери общности будем считать, что данные организованы в виде массива, имеющего иерархическую структуру. Массив состоит из элементов, которые в свою очередь могут быть либо массивами, либо состоять из полей. Под полем понимается единица поименованных данных, состоящих из любого количества битов или байтов. Любое поле может вновь делится на поля и т.д.

3. Анализ ограничений, налагаемых каждой из памятей на массивы, показывает, что для описания последних достаточно указать следующие три свойства:

а) однородность элементов по размерам,

---

*) С примерами таких задач авторы столкнулись при решении некоторых задач прикладной лингвистики [4] и автоматизации редакционно-издательских работ [5].

б) однородность структуры начальных частей элементов, т.е. состав, порядок расположения и размеры N первых структурных единиц элемента, где $N \leq L$ и L – размер элемента,

в) типы структурных отношений между элементами.

По первым двум свойствам можно выделить четыре типа массивов (табл.I), где "I" означает, что данное свойство выполнимо, "–" – что оно безразлично.

Таблица I

Тип массива	Обозначение	Свойства	
		Однородны структуры начал элементов	Элементы однородны по размерам
Неопределенный	U0	–	–
Однородный по размерам	U1	–	I
Частично-однородный	U2	I	–
Однородный	U3	I	I

Между приведенными четырьмя типами массивов существуют соотношения частичной упорядоченности по вложенности: $U3 \leq U2 \leq U\emptyset$, $U3 \leq U1 \leq U\emptyset$.

В зависимости от структурных отношений между элементами в массиве и типа массива можно выделить следующие структуры:

– п о с л е д о в а т е л ь н ы е: элементы данных записываются в массив в порядке их поступления, начало нового элемента данных помечается специальным символом; нет обязательной связи между местом элемента в массиве и его содержимым; тип массива для последовательных структур безразличен;

– у п о р я д о ч е н н ы е н е а с с о ц и а т и в н ы е: нет явно выраженной функциональной зависимости между значением признака (ключа) элемента, по которому проведено упорядочение, и его местом; тип массива – частично-однородный; признак, взятый в качестве ключа, должен находиться в однородной части элементов (примером может служить упорядоченный по частоте встречаемости частотный словарь);

– а с с о ц и а т и в н ы е н е о д н о з н а ч н ы е (имеют смысл для однородных массивов): место элемента определяется пу-

тем выполнения некоторой операции над его признаком, который должен находиться в однородной части элементов*);

- а с с о ц и а т и в н ы е о д н о з н а ч н ы е: существует взаимно-однозначное соответствие между значением признака элемента и его местом (например, когда над признаком выполняются целочисленные арифметические операции; в частном случае значение признака может совпадать с номером элемента в массиве); тип массива однородный; признак, определяющий место элемента, может быть опущен, так как он может быть легко определен по номеру элемента в массиве;

- л и н е й н ы е с п и с к о в ы е;

- д р е в о в и д н ы е.

В линейных списковых и древовидных структурах элементы данных записываются в массив в порядке их поступления. Но,в отличие от последовательных структур,для указания места следующего элемента выделяется специальный отсылочный адрес. Кроме того, с помощью отсылочных адресов массив может быть упорядочен по некото - рым признакам.В линейных списковых структурах каждый элемент имеет не более одного отсылочного адреса на каждый признак. Для древовидных структур или ориентированных графов с выделенной вершиной таких адресов может быть несколько. Нужный элемент выбирается по признаку.Признаки и адреса должны находиться в однородных частях. элементов. Тип массива - частично однородный.

Стековая память может работать со всеми типами массивов с последовательной организацией,регистровая - с однородными по размерам и однородными с последовательной или упорядоченной неассоциативной организацией и т.д.

4. Основными особенностями и достоинствами обобщенной памяти над массивами данных являются:

- сочетание последовательного и прямого доступа к п р о и з в о л ь н ы м поименованным единицам информации;

---

*) В литературе эту операцию называют по-разному: хеш-кодированием (hash-coding) или хешированием [6], неоднозначным ассоциативным кодированием [7], вычислением функции расстановки [8]. Один и тот же хеш-код может соответствовать разным значениям признаков элементов (но каждое значение признака элемента однозначно определяет хеш-код), т.е. для структур данного типа, кроме операции хеширования, должен указываться способ устранения неоднозначности.

- описание структуры данных - массива (как правило, описать структуру данных проще, чем подогнать их под какую-либо память);

- возможность изменения структуры (трафарета обработки) массива в ходе решения задачи;

- возможность динамического наращивания массивов, обеспечиваемая страничной организацией памяти ЭВМ;

- возможность распределения отдельных элементов или их группы на разные машины при решении задачи в системе машин.

5. Массивы удобно представлять в виде последовательности элементов, расположенных на потенциально-бесконечной в обе стороны ленте (рис. I). При этом различаются крайние левый и правый элементы данных. Им соответствуют левая ($\overline{\uparrow}$) и правая ($\overline{\downarrow}$) стрелки. Может также выделяться доступный для данной операции текущий элемент, ему соответствует текущая стрелка ($\overline{\uparrow}$).

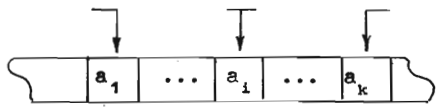

Рис. I

5.I. Рассмотрим операции общего вида над массивами.

- Приписать элемент данных справа-слева. Стрелка правая/левая сдвигается на один шаг вправо/влево, после чего осуществляется запись в правый/левый элемент.

- Считать элемент данных справа/слева без их разрушения. Доступным для чтения является правый/левый элемент данных. После чтения положение стрелок не изменяется, информация не разрушается.

- Считать элемент данных справа/слева с разрушением. Информация считывается из правого/левого элемента, доступного для чтения. Стрелки сдвигаются на один элемент влево/вправо.

- Поиск по образу данных слева направо/справа налево. Последовательно просматриваются все элементы данных, начиная с левого/правого и сравниваются с заданным образом до первого совпадения. При совпадении текущая стрелка устанавливается на этом элементе, и он становится доступным для чтения/записи. Если элемент данных не найден, то доступным объявляется очередной левый/правый элемент.

- Читать адрес элемента. Адрес доступного элемента, указанного текущей стрелкой, считывается.

- Поиск по адресу. Текущая стрелка устанавливается на элементе данных, заданном адресом. Этот элемент становится доступным для чтения/записи.

- Записать элемент данных. Информация записывается в доступный элемент, текущая стрелка при этом не передвигается. Длина записываемой информации должна соответствовать длине элемента. В противном случае информация усекается, либо дополняется нулями или пробелами (в зависимости от формата).

- Считать элемент данных без разрушения. Из доступного элемента считывается информация без разрушения, текущая стрелка не передвигается.

- Считать элемент данных с разрушением. Информация считывается из доступного элемента. Элемент обнуляется, текущая стрелка не передвигается.

- Считать элемент данных с разрушением. Информация считывается из доступного элемента. Элемент обнуляется, текущая стрелка не передвигается.

- Арифметические и логические операции над элементами данных.

- Сбор мусора.

5.2. Операциями специального вида для ассоциативных структур являются операции хеш-поиск и дихотомический поиск; для ассоциативных однозначных структур - ассоциативный поиск; для линейных списковых структур - линейный поиск, включение элемента данных, исключение элемента данных; для древовидных структур - поиск по дереву, включение элемента данных, исключение элемента данных.

Операциями над структурами являются

- изменение структуры массива (в последующих операциях работа ведется с массивом новой структуры);

- изменение структуры элемента, поля и т.д.

Приведенный набор операций является объединением операций, определенных в соответствующих памятях. Этот набор открыт, т.е. может пополняться как операциями общего вида, так и операциями специального вида с введением новых типов массивов.

6. Информация о структуре массива содержится в специальном массиве-дескрипторе (табл.2). Он заполняется и изменяется по мере того, как становятся известными и меняются характеристики массива. В любой момент времени дескриптор отражает текущее состояние массива. В нем хранятся адреса очередных элементов справа и слева, текущая длина массива в элементах, адреса текущей и первой страниц, номера начального и последнего элементов, обрабатываемых на данной машине в системе машин, информация о типе массива и типе элемента массива (фиксированной длины), длине элемента, длине страницы.

Таблица 2

Адрес очередного элемента справа		
Адрес очередного элемента слева		
Адрес текущей страницы		
Адрес первой страницы		
Длина страницы		Текущая длина массива
Тип элемента	Тип массива	Длина элемента
Номер начального элемента		Номер последнего элемента
Адрес основного описания		
Адрес активного описания		

Дескриптор содержит адрес размещения описания структуры массива (адрес основного описания). Если же информация о структуре массива задается динамически или изменяется во время решения задачи, то формируется новое описание структуры массива, адрес которого помещается в поле адреса активного описания.

NAME	TYPE	FORMAT	LENGHT	Адрес по совпадению	Адрес по несовпадению	Текущий адрес

Рис. 2

Описание представляет собой списковую структуру (рис.2), в которой описываются структурные отношения между элементами массива, полями в элементе и т.д.

В поле NAME помещается идентификатор, именующий структурную единицу.

В поле TYPE помещается одно из значений:

H - если единица является образом хеш-поиска;

T - если единица является образом древовидного поиска;

L - если единица является образом линейного поиска;

M - если единица имеет структуру массива.

В поле FORMAT описывается формат данных, соответствующих структурной единице с именем из NAME . Это могут быть двоичные с

фиксированной запятой, символьные, упакованные десятичные и другие данные. Длина структурной единицы помещается в поле LENGHT в формате двоичного числа с фиксированной запятой.

Адрес по несовпадению указывает на строку в описании массива, которая содержит информацию о структурной единице этого же уровня, адрес по совпадению - на структурную единицу уровнем ниже. Текущий адрес указывает на размещение соответствующих данных в памяти машины.

Заметим, что в данной работе рассматриваются лишь массивы, создаваемые в оперативной памяти.

7. Рассмотрим задачу создания модели стека на регистре в рамках существующей технологии [I]. Необходимо выполнить следующие процедурные операции:

I) NUL (R1,C) - обнуление вспомогательного регистра и счетчика;

2) R#LENR (C,R) - чтение адреса текущего элемента регистра R в счетчик C ;

3) C = C-1 - установление адреса последней записи;

4) R#SUBSTR (R1, C, 1, R) - чтение последнего элемента длиной в один символ из регистра R в R1, что соответствует 82 машинным командам; в обобщенной памяти чтение из стека достигается выполнением 8 машинных команд.

Предложенная обобщенная память расширяет возможности каждой из памятей в отдельности. Сочетание последовательного и прямого доступа к произвольным единицам информации значительно упрощает программирование задач обработки данных со сложной структурой.

Литература

I. ГЛУШКОВ В.М., ВЕЛЬБИЦКИЙ И.В. Технология программирования и проблемы ее автоматизации.-"Управляющие системы и машины",1976, № 6, с. 75-93.

2. ВЕЛЬБИЦКИЙ И.В. Метаязык R-грамматик.-"Кибернетика",1973, № 3. с. 47-63.

3. ГЛУШКОВ В.М., ВЕЛЬБИЦКИЙ И.В., СТОГНИЙ А.А. Об одном подходе к построению системного математического обеспечения современных вычислительных машин.-"Кибернетика", № 3, 1972. с. 25-35.

4. КОСАРЕВ Ю.Г., ЧУХАНОВА Н.А. Автоматический синтез алгоритмов по динамическим входным данным. -В кн.: Методы обработки информации. (Вычислительные системы, вып. 74.) Новосибирск, 1978, с. 52-63.

5. КОСАРЕВ Ю.Г., МОСКВИТИН А.А. Система широкого применения для автоматизации редакционно-издательских работ. -Там же, с.3-20.

6. КНУТ Д. Искусство программирования для ЭВМ. М., "Мир", 1976.

7. ВЕЛИЧКО В.М., ГУСЕВ В.Д., КОСАРЕВ Ю.Г., ЛОЗОВСКИЙ В.С., ТИТКОВА Т.Н. Ассоциативное кодирование: реализация и применение. -В кн.: Вычислительные системы. Вып. 62. Ассоциативное кодирование. Новосибирск, 1975, с.3-37.

8. ЛАВРОВ С.С., ГОНЧАРОВА Л.И. Автоматическая обработка данных. Хранение информации в памяти ЭВМ. М., "Наука", 1971.

Поступила в ред.-изд.отд.
25 октября 1978 года