

## ОБ ОДНОМ СПОСОБЕ ОРГАНИЗАЦИИ РАБОТЫ СИСТЕМЫ ПРОГРАММ

Ю. Н. ЕФИМОВ, Ю. Г. ЮРЕНКОВ

(Представлена научным семинаром УВЛ ТПИ)

Программы, организующие решение задач на ЭЦВМ, обычно рассчитаны на то, что алгоритм решения задачи представлен в виде линейной последовательности программ. При решении многих задач (планирование, управление и т. п.) систему программ удобнее представлять в виде графа нелинейной структуры. В данной статье рассматривается один из возможных путей построения программы, организующей работу систем программ указанного выше типа.

Будем считать, что совокупность программ задачи — это ориентированный граф без контуров (программный граф), где каждая вершина графа есть программный граф или отдельная программа, направление дуги графа указывает на преемственность работы вершин; если вершина — программный граф (подграф), то в свою очередь его вершины есть подграфы или отдельные программы. Отсюда можно ввести следующую иерархию среди набора программных и числовых массивов совокупности задач, решаемых на ЭЦВМ: массивы, общие для всех задач, — массивы ранга 0; массивы, общие для одной задачи, — массивы ранга 1; массивы, общие для части задачи, реализуемой подграфом (вершиной общего программного графа) — массивы ранга 2 и т. д.

Каждая задача перед началом работы должна передать программе, организующей счет (программа-диспетчер), полный набор данных о своих программных и числовых массивах (информационный массив задачи) и задание на работу системы программ задачи. В задании должны быть перечислены выходные вершины программного графа задачи, которые необходимо выполнить.

Введем обозначения для объектов, рассматриваемых в статье, и символы отношений между ними:

$k$  — номер ранга ( $0 \leq k \leq q$ );

$m_k$  — номер программного графа ( $1 \leq m_k \leq n_k$ );

$l$  — номер числового массива ( $1 \leq l \leq p$ ),

где  $p$  — граничное значение номера числового массива,

$q$  — граничное значение номера ранга,

$n_k$  — граничное значение номера программы графа ранга  $k$ . Символы отношений между объектами установлены следующим образом: для объектов  $A$  и  $B$ .

« $A(B)$ » — означает, что  $B \in A$ ;

« $A, B$ » — означает, что  $A$  и  $B$  объекты одного ранга;

« $A:B$ » — означает, что дуга графа направлена от вершины  $A$  к вершине  $B$ .

Так, запись « $A (B:C, D, E:C)$ » означает, что из вершины  $B$  графа  $A$  выходят две дуги, направленные к вершинам  $C$  и  $D$ , и из вершины  $E$  этого же графа выходит дуга, направленная к вершине  $C$ . Ниже верхний индекс « $A^{(j)}$ » означает, что из всех объектов типа  $A$  выбран один с номером  $j$ .

При этих обозначениях любой программный и числовой массив однозначно определяется заданием последовательности:

$$m_0(m_1(m_2(\dots(m_q(l))\dots))). \quad (1)$$

Если  $l=0$ , то (1) есть идентификатор программы; если  $l \neq 0$ , то (1) есть идентификатор числового массива, необходимого для работы программы  $m_0(m_1(\dots(m_q)\dots))$ . Если все  $m_k$  в (1), начиная с некоторого  $k > 0$ , равны 0, то при  $l \neq 0$  (1) задает числовой массив, общий для подграфа ранга  $(k-1)$ .

Информационный массив задачи состоит из 2-х частей: программно-го графа и списка числовых массивов задачи. Программный граф можно определить как объединение идентификаторов всех программ задачи с указанием их связей. Аналогично список числовых массивов есть объединение идентификаторов всех числовых массивов данного графа.

При принятых обозначениях подграф ранга  $k$  можно записать в виде следующей последовательности символов:

$$m^{(i)}_k(m^{(i_1)}_{k+1} : m^{(j_2)}_{k+1}, m^{(j_3)}_{k+1}, \dots, m^{(j)}_{k+1}). \quad (2)$$

Программный граф задачи задается совокупностью выражений (2) с общим числом членов

$$n = n_1 + n_2 + \dots + n_{q-1}. \quad (3)$$

Список числовых массивов, необходимых для работы подграфа  $m_k^{(i)}$  с указанием их применимости в вершинах подграфа, можно записать в виде

$$m^{(i)}_k(l_1, l_2, \dots, l_{i_1}), m^{(j_1)}_{k+1}(l_1, \dots, l_1), \dots \quad (4)$$

Список числовых массивов всей задачи полностью задается совокупностью выражений (4) с общим числом членов (3).

Выше задание было определено как перечисление некоторого числа выходных вершин программного графа задачи, т. е. задание представляет из себя список идентификаторов выходных программ системы. Этот список удобнее представить в виде, аналогичном (4),

$$m_1(m^{(i_1)}_2(m^{(j_1)}_3(m^{(\dots)}_4, \dots)), m^{(j_2)}_3(\dots), \dots). \quad (5)$$

Целью программы-диспетчера является: после получения от задачи задания на работу определить по программному графу последовательность программ, необходимых для выполнения задания, и выполнить их.

Программа-диспетчер такого типа разработана для ЭЦВМ «Урал-14 Д» в качестве управляющей программы автоматизированной системы планирования и управления производством «АСУП-Томск». «АСУП-Томск» применяется для планирования и управления производством нескольких предприятий. Совокупность программ «АСУП-Томск» представляет собой довольно сложную систему, которую удобнее всего представить нелинейным графом. Данная система программ применяется для различных целей планирования и управления, причем в каждом конкретном случае для работы нужны не все программы, входящие в систему. Программа-диспетчер рассчитана на число рангов массивов  $k=3$ . В приложении к системе программ «АСУП-Томск» введенные выше понятия программного графа и подграфа означают: программный граф задачи — вся система программ для полного расчета задач, решаемых

мых «АСУП-Томск» для одного предприятия; подграф — часть системы, решающая частную задачу в рамках всей системы.

Программа-диспетчер предназначена для последовательного выполнения заданий предприятий по планированию и управлению производством. Так как схема работы программы-диспетчера аналогична другим программам такого же типа, то ниже приводится только общее описание программы-диспетчера с указанием черт, отличающих ее от аналогичных программ. Программа-диспетчер последовательно выполняет задания предприятий, причем для ускорения работы в программе предусмотрено 5 параллельных потоков счета: решение задач, ввод информации с перфокарт, ввод информации с перфоленты, печать информации на алфавитно-цифровом печатающем устройстве (АЦПУ), выдача информации на перфоленту.

Программные и числовые массивы, как исходные, так и формируемые в процессе счета, хранятся в накопителе на магнитной ленте (НМЛ) и вызываются в оперативную память (НФ) только для работы. Данные об их местонахождении на НМЛ (адрес НМЛ) хранятся в управляющем массиве программы-диспетчера. Этот массив состоит из подмассивов, соответствующих рангам массивов системы. В первом подмассиве хранятся адреса НМЛ массивов ранга 1 и подмассивов управляющего массива с данными о массивах ранга 2 и т. д. Адреса НМЛ в подмассиве управляющего массива расположены в порядке возрастания номеров соответствующих массивов. Адрес НМЛ в управляющем массиве занимает 48 двоичных разрядов и содержит номер блока НМЛ, номер механизма НМЛ, номер зоны НМЛ и количество зон, занимаемых данным массивом. Относительные адреса в подмассиве управляющего массива определяются по следующей формуле:

$$A_0(m) = AN + 2m, \quad (6)$$

где  $m$  — номер массива,

$A_0(m)$  — относительный адрес адреса НМЛ от начала подмассива,

$AN$  — относительный адрес начала данных о массивах типа  $m$ .

Ввод информации в ЭЦВМ происходит по указанию оператора, задаваемому с пульта. Исходные массивы, вводимые в ЭЦВМ, по мере их ввода переписываются на НМЛ, данные с них заносятся в управляющий массив. По этим данным определяются адреса НМЛ массивов, формируемых в процессе решения (распределение НМЛ). Массивы, вызываемые из ЭЦВМ, формируются программами системы и записываются на НМЛ, после чего программа дает указание программе-диспетчеру выдать их на печать или перфоленту, а сама продолжает работу. Распределение оперативной памяти происходит следующим образом: программе отводится часть НФ определенной длины (постоянной для всех программ), и программист при составлении программы распределяет эту часть НФ по своему усмотрению; список массивов НФ перед началом работы программы передается программе-диспетчеру, которая по списку массивов программы формирует таблицу массивов ЭЦВМ. Причем в процессе работы программа может перераспределить свою часть НФ, о чем сообщает программе-диспетчеру, которая в свою очередь вносит изменения в таблицу массивов.

Программа-диспетчер начинает работу с начального ввода информации. После того, как в ЭЦВМ введены информационный массив задачи и задание, по программному графу задачи (ранг 1) определяется последовательность подграфов ранга 2, соответствующих заданию, и строится цикл на количество этих подграфов. Внутри этого цикла для каждого подграфа определяется линейная последовательность программ (ранг 3), соответствующих заданию, и эти программы выполняются для каж-

дого подграфа. В процессе вычисления программа-диспетчер проверяет наличие массивов, необходимых для работы. Если массивы отсутствуют, то программа-диспетчер выдает оператору список отсутствующих массивов и переходит к подграфу, независимому от выполняемого, или к другой задаче. После окончания цикла на число подграфов программа-диспетчер осуществляет переход к другой задаче.

Годовой опыт эксплуатации описанной программы-диспетчера для ЭЦВМ «Урал-14Д» показал ее достаточно высокую эффективность.

---