Л. А. БАТАНОВ, И. В. МИШУКОВ

О КВАЗИОПТИМАЛЬНОМ РАСПОЛОЖЕНИИ ПРОГРАММЫ В ПАМЯТИ ЦИКЛИЧЕСКОГО ТИПА

Имеющееся противоречие между сравнительно невысокой стоимостью циклических запоминающих устройств и большим временем ожидания при обращении по произвольному адресу в значительной степени сглаживается оптимальным программированием. Однако в общем виде задача оптимального расположения программы в запоминающем устройстве циклического типа достаточно сложна и требует больших затрат времени у программиста [1].

Будем считать, что запись программы в память циклического типа производится схемным путем по алгоритму: каждая последующая команда по вероятности располагается на расстоянии, пропорциональном времени исполнения предыдущей команды. Это условие, накладываемое на запись программы, приводит к квазиоптимальному ее расположению в памяти циклического типа.

Произведем оценку основного параметра, характеризующего такое расположение программы, — времени ожидания при поиске команды, следующей за данной. Время ожидания при обращении к циклической памяти будем измерять числом кодовых мест, которое приходится пропускать до осуществления записи данной команды. Примем следующие обозначения:

m — число команд, расположенное на данной дорожке циклической памяти;

M — полное число кодовых мест на дорожке;

r — коэффициент параллелизма, определяемый числом дорожек, на которых располагается одно информационное слово;

R — число разрядов слова.

Предположим, что некоторая m-ая команда может с равной вероятностью занять любое из M кодовых мест на дорожке циклической памяти.

Рассмотрим обращение к фиксированной дорожке и будем считать, что обращение к следующей происходит лишь после того, как заполнена предыдущая дорожка. Если на дорожке, к которой происходит обращение на запись команды, уже имеется m команд, то $P=\frac{m}{M}$ есть вероятность того, что при некотором $(m+1)$-ом обращении данное кодовое место окажется занятым. Тогда вероятности ожидания 0, 1, 2,......, m кодовых мест соответственно выразятся следующим образом:

$$P(0) = 1-\frac{m}{M};$$

$$P(1) = \frac{m}{M}\left(1-\frac{m-1}{M-1}\right);$$

$$P(2) = \frac{m}{M}\frac{m-1}{M-1}\left(1-\frac{m-2}{M-2}\right);$$

$$\cdots\cdots\cdots\cdots\cdots\cdots$$

$$P(k) = \left(1-\frac{m-k}{M-k}\right)\prod_{i=0}^{i=k-1}\frac{m-i}{M-i}.$$

Время ожидания при расположении некоторой $(m+1)$-ой команды можно вычислить по формуле

$$t_{m+1} = \sum_{k=0}^{k=m} k\left(1-\frac{m-k}{M-k}\right)\prod_{i=0}^{i=k-1}\frac{m-i}{M-i}. \qquad (1)$$

Воспользовавшись равенством (0.151) из (2) и произведя упрощающие пребразования для времени ожидания при расположении в памяти некоторой $(m+1)$-ой команды, получим следующее выражение:

$$t_{m+1} = \frac{m}{M-m+1}. \qquad (2)$$

Таким образом, время ожидания при поиске следующей команды, выраженное числом пропущенных кодовых мест дорожки, является возрастающей функцией от порядкового номера команды на данной дорожке.

Найдем среднее значение числа пропуска кодовых мест, приходящееся на одну команду, при условии, что все кодовые места дорожки заняты:

$$\bar{t} = \frac{1}{M}\sum_{m=0}^{m=M-1}\frac{m}{M-m+1} = \frac{M+1}{M}\sum_{m=0}^{m=M-1}\frac{1}{M-m+1} - 1. \qquad (3)$$

Произведя замену переменной, преобразуем формулу (3) к виду:

$$\bar{t} = \frac{M+1}{M}\sum_{k=2}^{k=M+1}\frac{1}{k} - 1$$

Воспользовавшись равенством (0.131) из (2), получим окончательное выражение для искомой величины среднего числа пропуска кодовых мест

$$\bar{t} \simeq \ln(M+1) - 1{,}42, \tag{4}$$

где $M \gg 1$. Отметим, что формула 4 является лишь некоторым приближением к действительному характеру расположения команд по предложенному алгоритму. В реальных программах первые команды располагаются на дорожке циклической памяти точно в тех кодовых местах, которые следуют на расстоянии, соответствующем времени исполнения предыдущих команд. Расположение последующих команд уже связано с пропуском кодовых мест. Как следует из формулы (2), первые команды теоретически дают отличное от нулевого значение времени ожидания при поиске на дорожке. Однако при $M \gg 1$, что имеет место в реальных конструкциях циклических запоминающих устройств (исключая динамические регистры, для которых $M \neq 1$), первые команды дают близкое к нулю значение времени ожидания при их поиске на дорожке.

Информация в циклическом запоминающем устройстве может быть представлена в последовательной, последовательно-параллельной и параллельной формах. Пусть некоторое слово располагается на r параллельных дорожках по $\frac{R}{r}$ разрядов в каждой. Полагая суммарное число разрядов всех кодовых мест равным постоянной величине N, с помощью формулы (4) найдем отношение величины среднего времени ожидания при последовательном способе представления информации в циклической памяти к соответствующей величине для последовательно-параллельного способа:

$$\gamma = \frac{r\left[\ln\left(\frac{N}{R}+1\right)-1{,}42\right]}{\left[\ln\left(\frac{N}{R}r+1\right)-1{,}42\right]}. \tag{5}$$

Из формулы (5) видно, что коэффициент γ возрастает с увеличением r. Максимальным выигрыш в величине времени ожидания получается при $r = R$. Однако увеличение коэффициента r связано с пропорциональным возрастанием количества электронного оборудования, в частности, усилителей записи и чтения. Поэтому в каждом отдельном случае может быть найдено оптимальное значение этого коэффициента с учетом необходимого оборудования.

Отметим, что полученные зависимости справедливы для команд линейных участков программы, в которых последовательность их исполнения не нарушается. Команды типа условного перехода нарушают естественный ход программы и учитывать их влияние на время выполнения программы в целом достаточно сложно. С некоторыми допущениями время поиска команды, следующей после команды условного перехода, можно принять равным в среднем половине периода циркуляции информации в циклической памяти.

С целью сокращения времени исполнения программы отдельные часто повторяющиеся циклические участки разворачивают в линию (3). Осуществить подобное расположение программы схемным путем не представляется возможным, так как трудно не только выделить циклические участки, но и определить частоту их повторения. Поэтому для упрощения схемного решения приходится довольствоваться лишь основным алгоритмом оптимального расположения команд в линейных участках программы.

Формирование адреса схемным путем для команды, следующей за данной, требует анализа кода операции предыдущей команды. Простым случаем был бы такой, когда длительности операций были бы одинаковыми. Однако в большинстве случаев операции сильно различаются по длительности. В связи с этим возникает задача сокращения оборудования в блоке, фомирующем весовые коэффициенты операций. Для этого имеющуюся упорядоченную последовательность длительностей операций t_1, t_2, ..., t_n необходимо разбить на такое число групп $s<n$, чтобы

$$\sum_{i=1}^{i=k}(t_к - t_i) + \sum_{i=k+1}^{i=l}(t_l - t_i) + \cdots + \sum_{i=p+1}^{i=n}(t_n - t_i) = \min.$$

Объединив в группы операции с приблизительно одинаковыми длительностями и присвоив группам значения максимальных по величине членов, образующих их, получим оптимальное решение.

Таким образом, блок весовых коэффициентов, проанализировав код операции предыдущей команды, в соответствующий момент времени разрешает поиск свободного кодового места для расположения в нем следующей команды.

Рассмотренный способ квазиоптимального расположения программы в памяти циклического типа дает эффект близкий к тому, который получается при оптимальном программировании.

ЛИТЕРАТУРА

1. А. И. Китов, Н. А. Криницкий. Электронные цифровые машины и программирование. Физматгиз, М., 1959.

2. И. С. Градштейн, И. М. Рыжик. Таблицы интегралов, сумм, рядов и произведений, Гифмл, М., 1962.

3. З. Л. Рабинович. Вопросы построения малых вычислительных машин с запоминающим устройством на магнитном барабане. Сб. «Вопросы вычислительной техники». Киев, 1961.