

$\gamma = 190$ . Окончательное определение коэффициентов модели П осуществлялось после нахождения  $\omega$  теоретическим методом. Кривая ( $\gamma$ ) на рисунке рассчитана по результатам проведенной идентификации.

С. В. Смирнов

## ОРГАНИЗАЦИЯ СПИСКОВ С ДВУМЯ СВЯЗЯМИ И РАБОТА С НИМИ ПРИ ПРОГРАММИРОВАНИИ НА ЯЗЫКЕ ПЛ/I ДОС ЕС

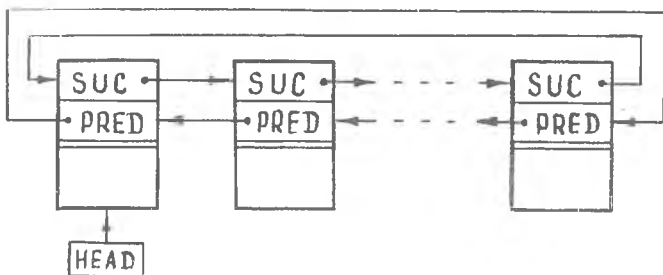
(К у й б ы ш е в)

Необходимость учета в машинном представлении данных их структурных отношений наиболее часто возникает при решении невычислительных задач, связанных с организацией систем дискретного моделирования, компиляции, информационного поиска, обработки свободной информации и т.п. Проблемам, связанным со структурами данных или информационными структурами, посвящена обширная литература, выполнено большое количество теоретических и практических работ.

В настоящем сообщении рассматривается весьма узкий из указанного круга проблем вопрос, сформулированный в заголовке, и обобщается опыт, накопленный авторами при разработке набора моделирующих подпрограмм *CREATE* [1]. Стимулом к изложению этого опыта послужили, с одной стороны, все более обнаруживающая себя тенденция написания системного математического обеспечения (чаще какой-либо его части) на языках высокого уровня (в данном случае — на языке ПЛ/I), а с другой стороны, тот факт, что программы на ПЛ/I могут рассматриваться как спецификация системных программ для последующего написания их на машинно-ориентированном языке [2].

Наибольшая гибкость в работе с линейными списками данных достигается при использовании циклических списков с двумя связями (рис. 1). В общем случае такие списки требуют большого объема памяти, однако, помимо очевидного достоинства, заключающегося в возможности продвигаться по списку как вперед, так и назад, они обладают тем преимуществом, что для исключения любого элемента из середины списка достаточно знать адрес этого элемента [3].

Универсальный алгоритмический язык ПЛ/I [4] располагает возможностями для весьма сложной организации данных, в частности,



Р и с. I.

линейных списков [5], однако, его подмножеству, входящему в наиболее распространенную операционную систему ЕС ЭВМ - ДОС (версии 1.3 и 2.0 [6]), - свойственны существенные в этом плане ограничения.

Организацию и обработку линейных списков в этом подмножестве ПЛ/1 можно воспроизвести с помощью массивов. При этом следует выделить три основных способа организации списка:

А. Память выделяется в виде единого массива, так что элемент списка представляет собой совокупность смежных элементов этого массива, каждый по порядку из которых несет строго определенную функцию (см. строки элементов списка на рис. I). Описатели массива, в котором размещается список, должны выбираться из условия сохранения семантики записей в элементах списка, имеющих в общем случае различные описатели. (Например, если все записи представлены в двоичной форме и лишь одна в десятичной, то массив должен иметь атрибут *DECIMAL*). Глобальные указатели списка, свойственные конкретным задачам, следует описывать с атрибутами *BINARY FIXED*.

В. Память выделяется в виде группы массивов, по одному для каждой разновидности строк элементов списка. При этом описатели массивов не зависят друг от друга и выбираются из удобства представления записей, и каждый элемент списка представляется группой из соответствующих элементов исходных массивов - по одному элементу с заданным индексом из каждого. Глобальные указатели, также массивы *SUC* и *PRED*, рационально описать с атрибутами *BINARY FIXED*.

С. Память выделяется в виде единого массива, описатели по

того согласованы с описателями данных лишь в части длины занимаемых участков памяти во внутреннем представлении (длина внутреннего представления элемента массива должна допускать размещение записи или ее семантического эквивалента, имеющей наибольшее внутримашинное представление), организация списка и работа с ним осуществляется с помощью продвижения по массиву специальной маски, представляющей собой базированную структуру. Глобальные указатели в этом случае целесообразно описывать с атрибутом *POINTER*.

Пр и м е р. *DECLARE 1 MASKA BASED(P),*  
*2 (SUC, PRED) BINARY FIXED,*  
*2 ZAPIS DECIMAL FLOAT,*  
*IM (9φ), (P, HEAD) POINTER.*

Здесь в массиве *IM*, каждый элемент которого занимает половину памяти, может быть размещено до 30-ти элементов списка со строками *SUC*, *PRED* и *ZAPIS*. Доступ к содержимому элемента списка осуществляется путем установки указателя *P* по адресу трех смежных элементов массива *IM*, представляющих интересующий нас элемент списка, и производится с помощью встроенной функции *ADDR(x)*. Например, для доступа к строке *ZAPIS* элемента списка, идентифицированного указателем *HEAD*, достаточно выполнить *P=HEAD*; и, если сразу же после действия с переменной *ZAPIS* этого элемента необходимо произвести операции с той же переменной следующего элемента списка, то маска сдвигается, используя локальный указатель *SUC*: *P=ADDR(IM(SUC))*. Рациональный выбор того или иного способа организации списка определяется конкретными условиями задачи, при этом способ организации определяет построение алгоритмов работы со списком.

С целью определения быстродействия основных алгоритмов работы со списком с двумя связями [3] для каждого из перечисленных способов организации таких списков в ПЛ/И ДЭС ЕС (версия 1.3) были проведены численные эксперименты на ЭВМ М-4030 с использованием машинного таймера. Результаты исследований позволяют сделать вывод о возрастании быстродействия соответствующих алгоритмов от А к С (например, алгоритмы включения в произвольное место списка нового элемента в условиях примера, приведенного в комментарии к способу С, отнимает соответственно 0,89; 0,75 и 0,69 мс).

По-видимому, здесь следует сделать общий вывод о том, что

механизм доступа к переменным с индексом в данной реализации ПЛ/И менее эффективен, чем доступ через указатели адресов переменных в памяти.

Таким образом, несмотря на кажущуюся тяжеловесность языковых конструкций третьего способа он в целом эффективнее двух других как ввиду большого быстродействия, так и скромных требований к согласованию описаний записей элементов списка и выделяемого в виде массива поля памяти. Именно этот способ реализован в системе *CREATE [1]*.

В заключение отметим следующее достоинство способа С. В общем случае в программе присутствуют несколько списков с различной внутренней организации и заранее неопределенной длиной [3]. Каждый список, организованный на основе индивидуальной базированной структуры, может черпать и отдавать необходимое число слов из общего списка свободных слов, и все эти списки размещаются в поле памяти, определенном в языке высокого уровня в виде единого массива. Ясно, что в подобной ситуации такое рациональное использование памяти для способов А и В невозможно.

## Л и т е р а т у р а

1. Будячевский И.А., Кораблин М.А., Смирнов С.В. Моделирование дискретных систем в рамках ПЛ/И (ДОС ЕС). - В кн: Автоматизация экспериментальных исследований. Вып. 9, Куйбышевский авиационный институт, 1976, с. 24-29.
2. Виноградов В.С., Калинин Л.А., Рывкин В.М., Смирнова Л.В., Шербин В.М. Использование *PL / I* для описания алгоритмов системных программ. В кн.: Языки системного программирования и методы их реализации. Киев, ИК АН УССР, 1974, с. 37-46.
3. Кнут Д. Искусство программирования для ЭВМ. Т. I. Основные алгоритмы. М., "Мир", 1976, с. 736.

1. Универсальный язык программирования ПЛ/1. М., "Мир", 1968, с. 352.
2. Х о о р К. Обработка записей. В кн.: Языки программирования. М., "Мир", 1972, с. 278-343.
3. В а с ю ч к о в а Т.Д., З а г у з о в а Л.К., И т - к и н а О.Г., С а в ч е н к о Т.А. Языки программирования ДЭС ЕС ЭВМ. Краткий справочник. М., "Статистика", 1977, 152 с.