

**СИСТЕМА
МОДУЛЬНОГО ПРОГРАММИРОВАНИЯ
ДЛЯ ЭВМ «МИНСК-22»**

Общее описание

Научно-исследовательский и проектно-технологический
институт систем планирования и управления
в электропромышленности

Тыгу Э.Х.

СИСТЕМА МОДУЛЬНОГО ПРОГРАММИРОВАНИЯ
ЛЯ ЭВМ "МИНСК-22"

Общее описание

Таллин 1970

Tartu Ülikooli Raamatukogu
ARHIIVKOGU

i22287997x

I. ОБЩЕЕ

I.1. Введение

Система модульного программирования (СМП) предназначена для программирования сложных инженерных и экономических задач. Наме- рение разработать систему программирования возникло у авторов посте- пенно, в ходе работы над большими программами инженерных расчетов. Авторам приходилось в течение ряда лет программировать заново одни и те же алгоритмы с незначительными изменениями. Отсюда возникло желание создать такую систему программирования, которая сохраняла бы в ЦВМ возможно большую часть однажды описанных соотношений и алгоритмов в виде, пригодном для повторного использования внутри любой новой программы.

Другая проблема, которая стояла перед разработчиками СМП, - организация эффективной совместной работы людей разных специаль- ностей и квалификации. В частности, такая проблема возникает при разработке математического обеспечения ЦВМ. Здесь требуются совме- стные усилия высококвалифицированных работников по прикладным спе- циальностям (экономистов, инженеров), программистов с высокой ква- лификацией, а для выполнения большого объема работ непосредственно по программированию (копированию программ) и отладке желательно привлечь программистов средней и низкой квалификации.

Первый вариант системы СМП на ЦВМ "Минск-22" был в 1966 году использован для организации работ по составлению больших программ технологических расчетов. Использование системы СМП для специали- зированной ЦВМ СТЭМ-2 позволила приблизительно за один год соста- вить и пустить программы объемом 24 000 команд.

Описываемая здесь система СМП значительно отличается от ее первых вариантов, так как за прошедшее время интересы авторов не- сколько изменились, и теперь они смотрят на СМП как на средство составления проблемно-ориентированных систем программ, в том числе и трансляторов для специальных (инженерных) языков. В этом отноше- нии надо отметить, что интересными для разработчиков СМП являются работы по созданию интегрированной системы программирования инже- нерных задач JCES [1] и системы автоматизации инженерного про-

ектирования АЕД [2] :

Настоящее описание предназначено для предварительного ознакомления с СМП и не может рассматриваться как инструкция по использованию системы. В первой главе дается общая характеристика СМП, основные понятия, приводятся общие сведения о работе обслуживающих программ и использования СМП для решения разных классов задач. Вторая глава целиком посвящена основной структурной единице СМП - модулю. В третьей главе рассматриваются вычислительные модели, а в четвертой - вычисления на моделях. Подробные правила организации работ, описания алгоритмов и программирования приведены в инструкциях [3].

В конце этого пункта приводятся некоторые общие сведения, характеризующие СМП.

Динамическое распределение памяти обеспечивает возможность составления программ, значительно превышающих объем оперативного ЗУ, позволяет вносить изменения в программы, наращивать их и стыковать между собой программы, составленные ранее для решения отдельных задач.

Для обработки данных имеются стандартные программы ввода-вывода и редактирования, используются стандартные средства описания и интерпретации форматов.

Для работы с массивами документов применяются экстракоды для поиска, считывания и записи документов. Обработка массивов ведется по страницам.

Отладка систем разрабатываемых программ производится в СМП отладочными программами. При этом отладку возможно осуществить с разной степенью подробности:

- полная прокрутка программ,
- печать значений и названий переменных,
- печать названий переменных и подпрограмм,
- печать сообщений о режимах работы программ,
- печать сообщений о динамических адресах переменных и программ,
- печать полного распределения памяти.

Основным понятием в СМП является вычислительная модель, которая состоит из двух типов объектов, весьма условно называемых переменными и отношениями. Переменными могут быть числовые, тексто-

вые и логические переменные, документы, отдельные реквизиты в документах, массивы документов и любые записи в памяти ЦВМ, занимающие определенное количество ячеек. Отношения описываются модулями, которыми являются таблицы, программы, форматы, модели и уравнения.

Хранимые на магнитных лентах вычислительные модели образуют проблемно-ориентированные средства программирования. Названия переменных, модулей и отношений образуют словарь, который используется для описания задач и алгоритмов в СМП.

Практика использования СМП показывает, что она создает благоприятные условия для организации совместной работы инженеров и программистов различной квалификации. Разработка большой системы программ в СМП происходит аналогично проектированию инженерных систем (машин, знаний и т.п.). Для описания общих принципиальных решений и для детального описания отдельных частей программы или данных используются разные средства. Все описания совершенно точно увязываются между собой. Всегда можно от общей схемы или описания объектов перейти к более детальным описаниям отдельных частей объекта, а для каждого детального описания определено, в какие более общие схемы оно вписывается. *

1.2. Возможности использования СМП

С точки зрения использования СМП возможно деление решаемых задач на три группы.

I. Небольшие, однократно решаемые задачи, при решении которых не используются ранее введенные в машину данные и программы. Эти задачи программируются целиком одним программистом и могут решаться в СМП или вне системы с одинаковым успехом.

В СМП программа такой задачи составляется в виде одного модуля, в котором нет вычислений на моделях. СМП накладывает ряд ограничений на программирование, что окупается для небольших задач лишь в том случае, если их программы в дальнейшем будут использованы для описания отношений в моделях. Сюда можно отнести программы, содержащие до 500 новых команд, программирование которых одним человеком занимает не более 1-2 месяцев.

* Работа выполнена сотрудниками НИПТИ Варьяс А., Вельтсон Х., Кабер М., Муст В., Мяннисалу М., Тинн К., Тьугу Э., Унт М.

2. Небольшие, но близкие друг к другу задачи, решаемые в большом количестве. Например, расчеты, относящиеся к одному и тому же объекту; экономические расчеты определенного характера (плановые, бухгалтерские); обработка экспериментальных данных и т.п.

Программы решения таких однотипных задач целесообразно составлять из небольших модулей, выражающих общие для данных задач отношения. Отношения связывают типичные, повторяющиеся в разных задачах переменные.

Эти переменные и отношения объединяются в модели, которые используются при решении каждой отдельной задачи как источник сведений, однажды введенных в машину. По мере решения новых задач модели в машине дополняются и накапливается все больше данных о проблеме.

Текущее состояние моделей фиксируется в документации, оформляемой по строго определенным правилам. Основное содержание модели выражается схемой модели, точные сведения содержатся в описаниях переменных и отношений.

Перед решением очередной задачи по схеме модели определяется, какие уже имеющиеся отношения и переменные могут быть использованы при решении. По схеме модели составляется псевдопрограмма или формируется задача автоматических вычислений на модели. Алгоритм решения задачи записывается в виде модуля, который является управляющей программой для вычислений на модели.

3. Сложные задачи или классы задач, программы которых составляются группой людей в течение длительного времени и содержат тысячи или десятки тысяч команд. Разработка таких программ требует четкой организации труда: строго определенной системы документации по алгоритмам и программам, организации взаимодействия программистов разной квалификации и людей разных специальностей. Примерами здесь могут служить проблемно-ориентированные системы математического обеспечения, программы управления сложными объектами, программы обработки данных в системе управления производством и т.п.

В этом случае СМП используется как основа для организации работ по следующим этапам:

- 1) составление схемы общей модели и алгоритма;
- 2) детальное описание частей алгоритма и составление детальных схем моделей;

- 3) составление модулей;
- 4) отладка модулей;
- 5) отладка частей программы;
- 6) отладка программы целиком;
- 7) решение серии контрольных задач и окончательное оформление документации.

Со 2-го по 5-й этап возможно практически сколь угодно мелкое дробление задачи на части. Каждая часть, описываемая отдельным модулем, может в свою очередь делиться на более мелкие модули.

Этапы 3, 4 и 5 могут выполняться программистами низкой квалификации. Первые два этапа выполняются специалистами по объекту (не программистами) совместно с консультантом-программистом. Лишь 6-й этап выполняется системным программистом высокой квалификации.

1.3. Основные понятия

Общим названием всех изменяющихся записей в СМП является переменная. С каждой переменной связывается:

- 1) множество значений переменной;
- 2) форма записи значений переменной;
- 3) идентификатор, обозначающий данную переменную.

При вычислениях на ЦВМ каждой переменной выделяется в памяти ЦВМ место, куда записано ее значение. В любой момент времени переменная может иметь не более одного значения. Значение переменной может быть также неопределенным, как, например, значение результата в самом начале вычислений.

Идентификатором переменной может быть любая комбинация цифр и букв, начинающаяся с буквы и содержащая не более пяти символов.

Перечислим некоторые возможные типы переменных:

- 1) переменная с целочисленными значениями,
- 2) переменная, принимающая в качестве значений практически любые числа (числа с плавающей запятой),
- 3) логическая переменная (принимает только значения "ложь" или "истина"),
- 4) переменная, значениями которой являются тексты определенной длины, т.е. любые последовательности алфавитно-цифровых символов, содержащие заданное число символов,

5) переменная, имеющая в качестве значений любой двоичный код определенной длины,

6) документ, содержащий в качестве реквизитов переменные любого из перечисленных выше типов,

7) массив документов,

8) таблица,

9) массив числовых или логических переменных.

Отношение выражает связь между переменными и используется для вычисления значений переменных. Например, отношение $\alpha + \beta + \gamma = \pi$ выражает связь между углами α , β и γ треугольника. По этому отношению значение любого угла может быть найдено по заданным значениям остальных двух углов.

Переменная, значение которой должно быть задано (вычисляется) при любом использовании некоторого отношения, называется входной (выходной) переменной данного отношения. Переменная, являющаяся одновременно для некоторого отношения входной и выходной, называется сильносвязанной. Переменные, значения которых могут задаваться или вычисляться при вычислении по отношению, но которые не являются ни входными, ни выходными, называются слабосвязанными. Входные, выходные, сильносвязанные и слабосвязанные переменные образуют множество связанных переменных отношения.

Каждое отношение имеет идентификатор, обозначающий название отношения.

Любое отношение в СМП описывается модулем. Существуют следующие типы модулей:

- программа ССК,
- таблица,
- формат,
- модель,
- уравнение.

Возможно расширение номенклатуры типов модулей. Например, при включении в СМП некоторого нового языка X может потребоваться новый тип модуля "X-программа". Такая "X-программа" при вычислениях будет интерпретироваться отлично от существующего модуля "программа".

Модули, записанные в ЯСК, приводятся при трансляции к типу "программа" или "уравнение". Это - программы в относительных адресах, состоящие из переадресуемой и непереадресуемой части.

Модули типа "таблица" и "формат" представлены в машине в виде таблиц, которые не меняются при сдвиге в ОЗУ. Первый выражает таблично заданное функциональное отношение. Второй - отношения "состоять из" и "является частью", связывающие документ и реквизиты.

Модуль типа "модель" является носителем сведений о переменных и связях между переменными. Там указано, какие отношения связывают каждую переменную, и имеются ссылки на модули, являющиеся описаниями этих отношений.

Модуль выражает один или несколько алгоритмов вычисления значений переменных и он аналогичен описанию процедуры в алгоритмических языках. Однако при использовании модуля в каждом конкретном случае необходимо указать, какие внешние переменные являются исходными данными и какие - результатами вычислений.

При использовании СМП все программы составляются только в виде модулей, что обеспечивает возможность стыковки любых программ и динамического распределения памяти. Каждый модуль полностью описывается на одном языке, но модуль может являться управляющей программой для расчета на некоторой модели, которая содержит описания на других языках.

Каждый модуль имеет название, выраженное идентификатором. При этом название отношения не обязательно совпадает с названием того модуля, которым отношение описано. Один и тот же модуль может описывать несколько отношений.

Поясним связь между модулем и отношением на примере. Пусть имеются переменные $J_1, U_1, R_1, J_2, U_2, R_2, J_3, U_3, R_3$, которые принимают числовые значения, и два модуля А и В. Модуль А имеет внешние переменные J, U, R и описывает следующие три алгоритма:

1) $J = U/R;$

2) $U = J \times R;$

3) $R = U/J.$

Модуль В имеет внешние переменные J_A, J_B, J_C и описывает алгоритмы:

- 1) $J_A: = - J_B - J_C;$
- 2) $J_B: = - J_A - J_C;$
- 3) $J_C: = - J_A - J_B.$

Очевидно, любое отношение, выраженное модулем А, можно короче выразить уравнением $U' = J' \times R'$, если под U' , J' и R' подразумевать переменные, которые данное отношение связывает. Точно также отношение, выраженное модулем В, можно выразить одним уравнением $J_A' + J_B' + J_C' = 0$.

На рис. 1а точками изображены переменные и кружочками - модули. Каждый модуль имеет отростки, соответствующие его внешним переменным.

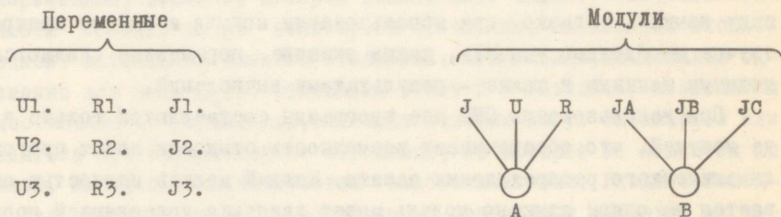


Рис. 1а

Опишем теперь модулем А три отношения $C1, C2, C3$, первое из которых связывает переменные $J1, U1, R1$, второе $J2, U2, R2$ и третье $J3, U3, R3$. Установим при этом между внешними переменными модулей и переменными, связанными отношениями $C1, C2, C3$, следующие соответствия:

$C1: U \dots U1$	$C2: U \dots U2$	$C3: U \dots U3$
$J \dots J1$	$J \dots J2$	$J \dots J3$
$R \dots R1$	$R \dots R2$	$R \dots R3$

Модулем В опишем отношение $C4$, связанные переменные которого соответствуют внешним переменным модуля следующим образом:

$C4: J_A \dots J1$
$J_B \dots J2$
$J_C \dots J3$

Опишем отношения графически также кружочками, соединив их линиями с теми переменными (точками), которые связаны с отношением. Получим схему, приведенную на рис. 1б.

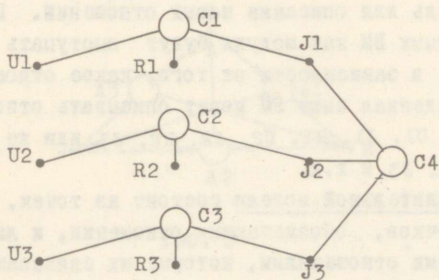


Рис. 1б

Эта схема описывает фрагмент электрической цепи, приведенной на рис. 1в.

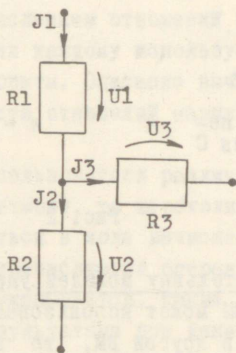


Рис. 1в

Вычислительная модель (ВМ) состоит из множества переменных и множества отношений, которые связывают эти переменные (и не связывают никаких других переменных). Вычислительная модель имеет название, выраженное индентификатором.

Переменные $J_1, J_2, J_3, U_1, U_2, U_3, R_1, R_2, R_3$ и отношения C_1, C_2, C_3, C_4 можно рассматривать как некоторую вычислительную модель. Интересным свойством ВМ является то, что она может использоваться как модуль для описания новых отношений. Причем в качестве внешних переменных ВМ как модуля будут выступать то одни, то другие переменные, в зависимости от того, какое отношение описывается. Например, приведенная выше ВМ может описывать отношение, связывающее переменные $U_1, J_1, R_1, U_2, J_2, R_2, J_3$, или же отношение, связывающее J_1, J_2, J_3 и т.д.

Схема вычислительной модели состоит из точек, обозначающих переменные, кружочков, обозначающих отношения, и линий, соединяющих переменные с теми отношениями, которые их связывают. Примером схемы ВМ является рис. 1б.

На схеме ВМ переменные связываются с отношением линиями, как показано на рис. 2.

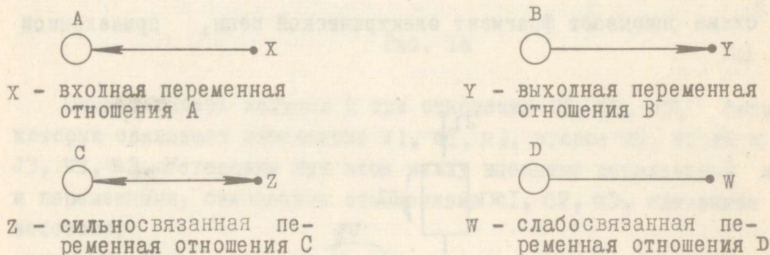


Рис. 2

Вложение вычислительных моделей упрощает описание сложных ВМ. Поскольку каждая ВМ может использоваться как модуль, описывающий некоторое отношение в другой ВМ, то возможно укрупнять структуру сложных ВМ. Для этого несколько отношений и с ними связанные переменные объединяются в новую ВМ, которая является вложенной в ис-

ходную, и описывает там одно новое и более общее отношение. Наоборот, если некоторое отношение трудно описать точно на одном уровне, то целесообразно представить его в виде ВМ, которая состоит из нескольких более простых отношений.

В рассмотренном выше примере модуль А, описывающий отношения С1, С2 и С3, можно представить в виде ВМ из трех отношений, схема которой приведена на рис. 3.

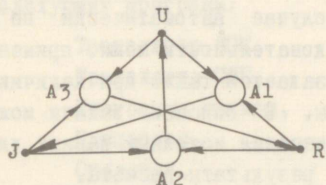


Рис. 3

В данном случае все же проще использовать модуль типа "уравнение": $J - U/R = 0$. Практически полезные вычислительные модели могут оказаться значительно сложнее и содержать десятки переменных и отношений.

Вычислительным процессом на ВМ является последовательное применение отношений для вычисления значений переменных. Вычислительный процесс можно описать перечислением отношений в порядке их применения. Этого достаточно, если каждому используемому отношению соответствует только один алгоритм. Описание вычислительного процесса в виде последовательности отношений называется вычислительной схемой на ВМ.

Если отношение может использоваться различным образом (ему соответствует несколько алгоритмов), то необходимо указать, как это отношение должно использоваться в ходе вычислений. Для этого вместе с названием отношения перечисляются операнды - переменные, которые связаны отношением, и указывается, какие из них являются исходными данными, какие - результатами при данном применении отношения. Название отношения вместе с указанием операндов образует макрокоманду.

Описание вычислительного процесса в виде последовательности макрокоманд называется макропрограммой. Макропрограмма — операционная схема алгоритма.

Некоторые вычисления на ВМ можно выполнять в СМП без описания вычислительного процесса. В этом случае задается только описание задачи, которое состоит из списка исходных переменных и списка результатов. Описание вычислительного процесса в виде макропрограммы составляется в этом случае автоматически по схеме ВМ, из которой видно, в какой последовательности можно применить отношения.

Описание задачи задается также при наличии вычислительной схемы или макропрограммы. В описании задачи можно указать, из каких ячеек памяти взять значения исходных данных для вычислений на модели, и куда заслать результаты расчета.

Описание задачи вместе с вычислительной схемой или макропрограммой (если они имеются) задаются в модуле, который называется управляющей программой. Управляющая программа содержит описание алгоритма использования ВМ для решения конкретной задачи.

Управляющая программа может содержать операторы, выполняющие вычисления вне модели, логические операторы, управляющие ходом вычислений, и операторы обращения к модели. Последние состоят из описания задачи, дополненного в некоторых случаях вычислительной схемой и макропрограммой.

1.4. Общие сведения о работе обслуживающих программ

Основной структурной единицей в СМП является модуль. Модули несут сведения об алгоритмах, переменных, связях между переменными и алгоритмами. Каждому отношению и каждой модели соответствуют модули, которые вызываются в ОЗУ при вычислениях по отношению и модели. Модули могут располагаться в произвольном месте ОЗУ и копироваться в нескольких экземплярах, чем обеспечивается рекурсивность обращения к отношениям. Исходные данные и результаты вычислений по модулю являются значениями его внешних переменных. Значения внешних переменных (или их адреса для длинных переменных) находятся во время вычислений в части ОЗУ, называемой рабочим полем. Значения остальных переменных модели хранятся в ОЗУ в поле переменных.

Обслуживающие программы СМП образуют малую операционную систему - МОС, выполняющую задания на вычислительной машине. Задание задается в виде последовательности системных команд ПУСК, ТРАНСЛИРОВАТЬ, ОТЛАДКА, ИЗЛАТЬ и т.п. перфоленты через старт-стопный фототех.

Работой МОС управляет программа Монитор, которая считывает системную команду, анализирует ее название и параметры и передает управления одной из следующих программ:

Транслятор ЯСК,
Транслятор СМП,
Программа архива,
Процессор,
Отладчик.

Если работа, заданная системной командой, выполнена успешно, то управление передается опять Монитору, который печатает на телетайпе сообщение о выполненной работе и выдает следующую системную команду.

Алгоритм вычислений для СМП задается макропрограммой, которая содержит:

- макрокоманды, определяющие вычисление по некоторому модулю,
- макрокоманды, требующие выполнения некоторых экстракодов системы, обеспечивающих ввод-вывод, работу с массивами (выборка, считывание и запись документов) и т.п.

Макропрограмма может задаваться программистом при обращении к системе. Но для достаточно хорошо определенных задач макропрограмму может составлять сама система. В последнем случае задается описание задачи, которое в словах выражается следующим образом: "Вычислить на модели M значения переменных v_1, v_2, \dots, v_m по значениям $\alpha_1, \alpha_2, \dots, \alpha_n$ переменных u_1, u_2, \dots, u_n ".

Вычислениями в СМП управляет Процессор, состоящий из Загрузчика, Интерпретатора и Планирующей программы. Загрузчик обеспечивает динамическое распределение памяти при отладке и при выполнении программ, не помещающихся целиком в оперативной памяти.

При вычислениях в СМП ОЗУ делится на рабочее поле, поле переменных, поле модулей и место для обслуживающих программ (см.рис.4).

Рабочее поле
Обслуживающие программы
Поле переменных
Модули

Рис. 4

Загрузчик распределяет место в поле переменных для тех переменных модели, которые указаны в макрокомандах выполняемой макропрограммы. Он загружает в ОЗУ также все модули, необходимые для выполнения макропрограммы. Алгоритм для распределения памяти для модулей такой, что наиболее долго в памяти находившиеся модули затираются, если вычисления по ним не приостановлены. Все модули с приостановленными вычислениями сохраняются в ОЗУ до конца вычислений по ним.

Интерпретатором осуществляется связь между модулями. На время работы модуля интерпретатор засылает значения или адреса внешних переменных в рабочее поле. Интерпретатор сохраняет также значения переменных тех моделей, вычисления на которых были приостановлены выполнением следующего оператора обращения к моделям. Если модуль — программа, то передается управление модулю, если таблица или формат, то используются подпрограммы выборки из таблиц или работы с форматом. Если модуль, к которому обращаются в макрокоманде, является моделью, то по макрокоманде формулируется новая задача вычисления на модели, решение "старой" задачи приостанавливается.

Новые задачи могут решаться на прежней модели, на новой модели или на новом экземпляре модели с приостановленными вычислениями. На рис. 5 приведена схема хода вычислений на модели M_1 по макропрограмме $a_{11}, a_{12}, a_{13}, a_{14}$, в которой макрокоманда a_{13} — вычисление на модели M_2 . Последнее выполняется по макропрограмме a_{21}, a_{22}, a_{23} , где в макрокомандах a_{21} и a_{23} внутри модулей формируются но-

вые задачи на новом экземпляре модели M_1 и на модели M_3 .

Планирующая программа используется в системе для автоматического составления макропрограмм.

Отладчиком пользуются при отладке отдельного модуля. В этом случае создается специальная отладочная среда и исключаются обращения к другим модулям при выполнении отлаживаемого модуля.

Планирующая программа использует схему модели, на которой задача должна решаться. По этой схеме составляется последовательность отношений, которые могут быть применены при заданных исходных данных задачи, из этой последовательности исключаются те отношения, которые не нужны для решения данной задачи, и полученную последовательность преобразуют в форму макропрограммы.

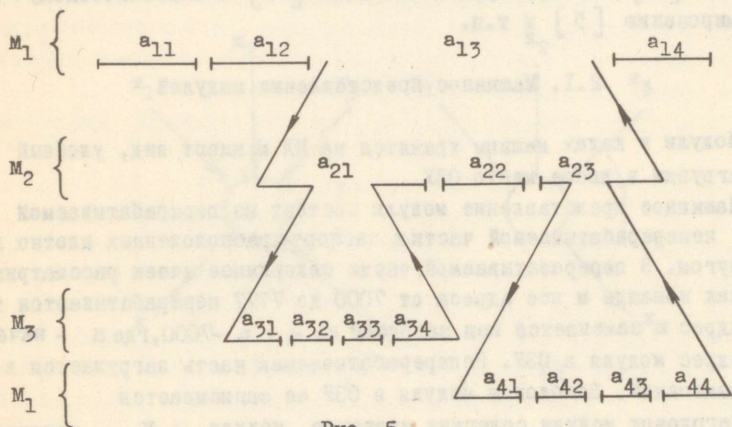


Рис. 5

2. МОДУЛИ

Одно и то же отношение можно задать разными модулями. Как правило, выражаемое любым модулем отношение может быть описано программой. Выбор типа модуля для описания отношения в большой мере зависит от программиста. Для облегчения выбора в данной главе приводится краткая характеристика тех типов модулей, которые применимы в СМП.

При программировании, т.е. при действительном составлении модулей, следует, кроме данной главы, руководствоваться теми инструкциями, которые относятся к соответствующему способу программирования [3], к структурным таблицам [4] и символическому программированию [5] и т.п.

2.1. Машинное представление модулей

Модули в кодах машины хранятся на МЛ и имеют вид, удобный для их загрузки в любое место ОЗУ.

Машинное представление модуля состоит из перерабатываемой части, неперерабатываемой части паспорта, расположенных плотно друг за другом. В перерабатываемой части содержимое ячеек рассматривается как команды и все адреса от 7000 до 7777 перерабатываются так, что адрес a заменяется при загрузке на $a + a - 7000$, где a - начальный адрес модуля в ОЗУ. Неперерабатываемая часть загружается в ОЗУ без изменений. Заголовок модуля в ОЗУ не записывается.

Заголовок модуля содержит название модуля - N_m , дополнение контрольной суммы - $k \sum$, количество ячеек в перерабатываемой части модуля - n_1 и общее количество ячеек, занимаемых перерабатываемой и неперерабатываемой частью модуля - n_2 .

Заголовок модуля имеет следующую форму:

$k)$	$+$	00	N_m	
		6	30	дв.разрядов
$k + 1)$			$k \sum$	
			37	дв.разрядов
$k + 2)$	$+$	n_1	n_2	0000
		12	12	12
				дв.разрядсв

При вычислениях модуль интерпретируется, в зависимости от своего типа, по-разному. Так, например, модуль, являющийся по существу программой вычисления значения функции $f(x_1, x_2, \dots, x_m)$, может интерпретироваться как оператор присваивания с правой частью $f(x_1, x_2, \dots, x_m)$ или как уравнение $f(x_1, x_2, \dots, x_m) = 0$.

В первом случае он имеет тип ССК и описывает отношение, которое имеет схему рис. 6а. Во втором случае это — модуль типа УРАВНЕНИЕ, описывающий отношение со схемой рис. 6б.

В некоторых модулях отсутствует перерабатываемая или неперерабатываемая часть. Табл. I показывает состав различных типов модулей.

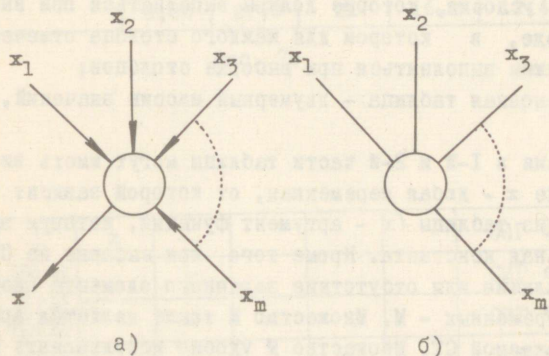


Рис. 6

Таблица I

Тип отношения	Таблица	Формат	Модель	Уравнение	Программа (ССК)
Перерабатываемая часть	Нет	Нет	Нет	Есть	Есть
Неперерабатываемая часть	Есть	Есть	Есть	Может быть	Может быть

2.2. Таблицы

Таблицы задаются в СМП в форме структурных таблиц (СТ). Они используются для представления конечнозначных функций нескольких переменных. Практически любая "человеком для человека" составленная таблица приводима без существенных изменений к виду структурной таблицы. Пример приведен на рис. 7а и б. Обе таблицы выражают одно и то же отношение, вторая из них записана в форме СТ.

В полной структурной таблице четко выделяются следующие части:

- 1) условия выборки строк;
- 2) условия выборки столбцов;
- 3) поле, в котором для каждой строки единицами (крестиками) отмечены те условия, которые должны выполняться при выборке строки;
- 4) поле, в котором для каждого столбца отмечены условия, которые должны выполняться при выборке столбцов;
- 5) основная таблица - двумерный массив значений, выбираемых из СТ.

Условия в 1-й и 2-й части таблицы могут иметь вид $x = a$, $x \neq a$, $x \leq a$, где x - любая переменная, от которой зависит значение, выбираемое из таблицы (x - аргумент функции, которую задает СТ), a - произвольная константа. Кроме того, при выборке из СТ может учитываться наличие или отсутствие заданного элемента (константы) в множестве переменных - M . Множество M также является аргументом функции, выражаемой СТ. Множество M удобно использовать как множество примечаний, которые заляются только при некоторых обращениях к таблице.

Множество M используется в условиях вида $a \in M$ и $a \notin M$, где a - константа.

При обращении к СТ проверяется выполнение всех условий выборки строки и столбца и выбирается значение из первой допустимой строки и первого допустимого столбца (допустимых в том смысле, что выполнены все условия, помеченные для данной строки или столбца единицами в поле 3 или 4).

Перечислим здесь некоторые возможности использования СТ.

- 1) Представление алгоритмов принятия решений. Например, в виде

Вид резьбы (ВО)	ШАГ до	Предел прочности (СИГ) до				
		70,00	80,00	93,00	107,00	125,00
Резьба наружная	3,00	41,0	32	24,6	18,9	14,7
	4,00	35,8	28	21,6	16,3	12,9
	5,00	29,4	23	17,7	13,6	10,6
	6,00	27,0	21	16,2	12,4	9,7
Резьба внутренняя упорная	3,00	35,8	28	21,6	16,3	12,9
	4,00	33,3	26	20,0	15,3	12,0
	5,00	28,2	22	16,9	13,0	10,0
	6,00	27,0	21	16,2	12,4	9,7

Рис. 7а

ШАГ $\leq 3,00$	ШАГ $\leq 4,00$	ШАГ $\leq 5,00$	ШАГ $\leq 6,00$	ВО = РНАПРО	ВО = РВНУЛО	x				СИГ $\leq 70,00$
						x				СИГ $\leq 80,00$
							x			СИГ $\leq 93,00$
								x		СИГ $\leq 107,00$
									x	СИГ $\leq 125,00$
x				x		41,0	32	24,6	18,9	14,7
	x			x		35,8	28	21,6	16,3	12,9
		x		x		29,4	23	17,7	13,6	10,6
			x	x		27,0	21	16,2	12,4	9,7
x					x	35,8	28	21,6	16,3	12,9
	x				x	33,3	26	20,0	15,3	12,0
		x			x	28,2	22	16,9	13,0	10,0
			x		x	27,0	21	16,2	12,4	9,7

Рис. 7б

(ТАБЛИЦА: ВВ2/СЖАТИЕ: 0/ТИП: 0001/СТРОК: 8/СТОЛБЦОВ: 5/76/

(41/32/24,6/18,9/14,7/)

(35,8/28/21,6/16,3/12,9/)

(29,4/23/17,7/13,6/10,6/)

(27,0/21/16,2/12,4/9,7/)

(35,8/28/21,6/16,3/12,9/)

(33,3/26/20,0/15,3/12,0/)

(28,2/22/16,9/13,0/10,0/)

(27,0/21/16,2/12,4/9,7/)

(СТРОКИ/

(ШАГ/МР/Д/3/)

(4*4/)

(4*5/)

(4*6/)

(ВО/РА/Т/РНАПРО/)

(4*РВНУПО/))

((42/)

(22/)

(12/)

(06/)

(41/)

(21/)

(11/)

(05/))

(СТОЛБЦЫ/

(СИГ/МР/Д/70/)

(4*80/)

(4*93/)

(4*107/)

(4*125/))КОНЕЦ

Рис. 7в

СТ можно задать алгоритм игры в "крестики и нолики", алгоритм выбора технологического варианта обработки детали.

2) Представление функций многих переменных, например, стоимости обработки детали в зависимости от размеров, конфигурации и материала детали, от способа обработки и применяемой оснастки; нормативных материалов, используемых при управлении производством - нормативов расхода материалов и рабочего времени на изготовление продукции, характеризуемой многими параметрами.

Структурная таблица является весьма компактной и наиболее наглядной формой представления отношения. Но выборка из СТ занимает, как правило, больше времени, чем вычисления по соответствующей программе. Поэтому СТ не рекомендуется применять для описания отношений, используемых внутри многократно повторяемых циклов, если не обходима экономия времени счета.

В СМП структурные таблицы записываются на бланках стандартной формы и перфорируются на телетайпе. Пример копии перфорации СТ приведен на рис. 7в. Транслятор СМП производит контрольную печать СТ на АЦПУ. Пример контрольной печати дан на рис. 7г.

2.3. Программы

Любую программу желательно в СМП представить в виде модуля, так как в этом случае она применима наиболее общим образом. Исходные данные и результаты программы описываются как внешние переменные модуля. Ввод, редактирование и печать желательно отделить от основной вычисляющей программы, так как данные, передаваемые через устройство фотоввода и печатающие устройства, теряются для системы, и не могут быть внешними переменными.

Если длина программы превышает 512 ячеек, то она делится на несколько модулей. Один из них, содержащий начало и конец программы, становится управляющей программой, в которой другие модули используются посредством операторов обращения к модели.

В любой программе-модуле допускается обращение ко всем модулям, описывающим отношения в моделях. Таким образом организуется взаимодействие между программами.

Программы-модули записываются в виде символических программ (СИМП).

Внешние переменные программы-модуля или их адреса располагаются в рабочем поле, начинающемся с адреса $0100_{(8)}$. Причем, если переменная занимает менее 4 ячеек, то в рабочем поле находится ее значение, если же переменная занимает больше места, то в рабочем поле находится ее адрес в виде $+n - I 0000$ а, где n - число ячеек, занятых переменной, а - адрес первой ячейки переменной. Обращение к библиотечным стандартным программам производится через ячейку $4000_{(8)} + i$, где i - номер СП. Программа кончается всегда переходом к стандартной ячейке возврата, адрес которой - $0177_{(8)}$.

Пример.

```

010/ +ПРИМЕР МОДУЛЯ
020// НАЧ / 7000В
030/ ВХОЛ / П / I77В, ТПП
040/ У / ЗНАЧ / I00В
050/ X // I01В
060/ З // I02В
070// П / X, 40В
080// ИЗАП / 4026В, I7В
090// УПЗ / П2, 40В
I00// УП / ПЗ, 3
I10// СПР / 40В/
I20// СПВ / 41В, У
I30/ ТПП / ОСТ /
I40/ П2 ЧП / 2
I50/ ПЗ // 6
I60// КОН / ПРИММО

```

Этот модуль имеет название ПРИММО и вычисляет y по формуле $y = 2x + e^x + 6z$, причем внешние переменные расположены следующим образом:

```

0100) y
0101) x
0102) z

```

Для каждой внешней переменной можно проверить, является ли она входной или выходной при данном обращении к модулю. Это указано при интерпретации модуля для i -й внешней переменной признаком t_i в виде

$$5305 + 1) + \underbrace{\quad}_{0-12} \underbrace{t_1}_{13-15} \underbrace{\quad}_{16-36} \quad \text{дв.разр.,}$$

где

$$t_1 = \begin{cases} 0 & \text{- переменная не используется} \\ 2 & \text{- выходная переменная} \\ 4 & \text{- входная переменная} \\ 6 & \text{- входная и выходная переменная} \end{cases}$$

Отсюда следует, что программой можно описать отношение со слабосвязанными переменными. Например, если программа вычисляет любую из m внешних переменных x_1, x_2, \dots, x_m по заданным значениям остальных $m-1$ переменных, то программа имеет логическую структуру, показанную на рис. 8.

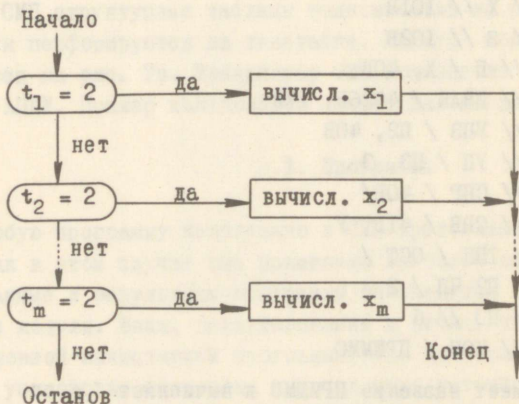


Рис. 8

2.4. Форматы

Формат выражает отношение между документом и реквизитами, из которых документ состоит (документ и реквизиты совпадают по смыслу с понятиями записи и поля записи.) Формат содержит сведения о том, в какой последовательности реквизиты входят в документ и какова длина (количество двоичных символов) каждого из них. Длиной документа является сумма длин реквизитов.

Формат описывается в следующем виде:

(ФОРМАТ <название формата>
РЕКВИЗИТЫ: (< назв.реквизита > <тип> <кол.символов>)
.....
(< назв.реквизита > <тип> <кол.символов>)

Названия формата и реквизитов - идентификаторы. Для реквизита указывается тип одним из следующих слов:

ДЕС - десятичный,
ВОСЬМ - восьмеричный;
СИМВ - символы телеграфного кода,
ДВ - двоичный.

Длина реквизита в битах получается умножением количества символов на 4, 3, 6 или 1 соответственно типу ДЕС, ВОСЬМ, СИМВ или ДВ.

Пример.

(ФОРМАТ ФI23
РЕКВИЗИТЫ: (НОМ ДЕС 4
(ДАТА СИМВ 6
(СУММА ДЕС 5)).

Отношение, заданное форматом, можно использовать для: 1) составления значения документа по заданным значениям всех его реквизитов, 2) определения значения одного или нескольких реквизитов по заданному значению документа, 3) изменения значения документа заменой значений некоторых реквизитов на новые заданные значения.

Форматы позволяют представлять вычисление всех реквизитов вычислительными моделями, которые имеют схему, аналогичную приведенной на рис. 9. В такой модели "нестандартной частью" являются только отношения между реквизитами - нижний уровень на рис. 9.

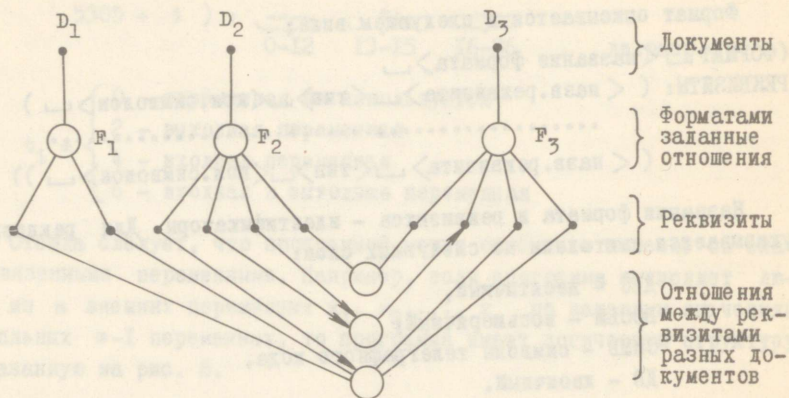


Рис. 9

2.5. Уравнения

Весьма распространено описание отношений уравнениями, которые в СМП должны иметь вид

$$f(x_1, x_2, \dots, x_m) = 0,$$

где x_1, x_2, \dots, x_m - действительные переменные,
 f - функция переменных x_1, x_2, \dots, x_m .

Модуль, описывающий уравнение $f(x_1, x_2, \dots, x_m) = 0$, - программа, вычисляющая значение $f(x_1, x_2, \dots, x_m)$ по заданным значениям переменных x_1, x_2, \dots, x_m . Этим модулем заданное отношение может использоваться для вычисления любой переменной x_i ($i = 1, 2, \dots, m$) при заданных значениях переменных $x_1, x_2, \dots, x_{i-1}, x_{i+1}, \dots, x_m$ и начальном значении переменной x_i при условии, что подстановкой значений этих переменных в $f(x_1, x_2, \dots, x_m) = 0$ получается уравнение $\varphi(x_i) = 0$, разрешимое методом хорд при заданном начальном

значении переменной x_i . В случае, когда начальное значение переменной x_i неизвестно, оно принимается равным 0. В частности, такой способ позволяет методом хорд найти правильное решение уравнения, линейного относительно x_i .

Модули-уравнения записываются в символических кодах так же, как модули-программы. В конце работы программы вычисленное значение функции $f(x_1, x_2, \dots, x_m)$ должно оставаться в сумматоре.

2.6. Модули связи

Вычислительная модель является в СМП модулем, однако она используется для описания отношений отлично от других модулей. Чтобы модель M_1 использовать для описания отношения в модели M_2 , необходимо создать дополнительно модуль связи, в котором указано соответствие между переменными модели M_1 и M_2 .

Модуль связи является таблицей из двух столбцов. В первом столбце указаны переменные модели, в которой описывается отношение, во втором столбце указаны соответствующие им переменные модели, описывающей отношение. В заголовке таблицы указывается модель, которая задает отношение.

Модуль связи формируется в ЦВМ автоматически по описанию отношения (см. п. 3.3).

Например, по описанию

```
(ОТНОШЕНИЕ_С1_МОДЕЛЬ_А_И_И_СЛ_И_
                У_СЛ_У_
                Р_СЛ_Р_)
```

составляется модуль связи

Модель: А	
Связанная переменная	Ей соответствует в модели А переменная
И	И
У	У
Р	Р

Модули связи соединяют вычислительные модели в единую структуру моделей.

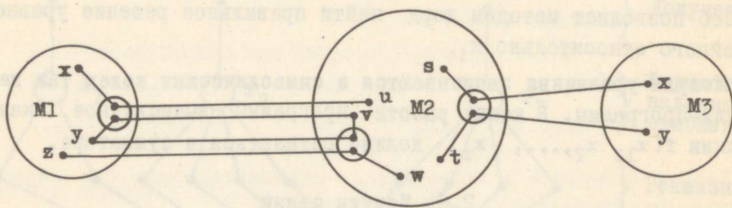


Рис. 10

На рис. 10 показаны три модели M1, M2 и M3, соединенные между собой модулями связи так, что создается следующее соответствие переменных разных моделей

M1 M2

x → u

y → v

M2 M3

s → x

t → y

M2 M1

v → y

w → z

3. СОСТАВЛЕНИЕ МОДЕЛЕЙ

3.1. Описание модели

В данной главе рассматриваются текстовые описания моделей, которые формально точно определяют каждую вычислительную модель.

Описание модели содержит:

1. Определитель модели, который выражен одним из слов:

НОВЫЙ,
НОВАЯ,
НОВОЕ,
ПРОДОЛЖЕНИЕ.

2. Идентификатор, который становится названием модели.

3. Шифр магнитной ленты, на которой хранится модель. Шифром может быть любой идентификатор.

4. Продолжение описания. Продолжение описания может быть пустым (может отсутствовать).

Описание вычислительной модели без продолжения описания имеет следующую структуру:

(< определитель модели > _ < идентификатор > _ < шифр МЛ > _).

Пример:

(НОВЫЙ _ СКЛАД _ А072 _).

Слова НОВЫЙ, НОВАЯ или НОВОЕ указывают на то, что описывается новая ВМ, после ввода такого описания в ВМ выделяется место для новой модели и эта модель заносится в каталог модулей и в каталог моделей. С этого момента она может использоваться как любой другой модуль для составления моделей и как ВМ для решения задач на модели.

Каждая модель может описываться несколькими описаниями, которые учитываются последовательно в порядке их введения. Слово ПРОДОЛЖЕНИЕ указывает, что описывается уже ранее определенная модель.

Все сведения о модели, кроме ее названия, задаются в виде описаний. Описания бывают следующие:

- 1) описание переменных,
- 2) описание отношений,
- 3) описание удаления.

Последовательно друг за другом записанные описания образуют продолжение описания. Описание вычислительной модели имеет в общем случае следующую структуру:

(< определитель модели > ⊔ < идентификатор > ⊔
 < шифр МЛ > ⊔ < продолжение описания >).

Пример:

(НОВАЯ ⊔ СХЕМА ⊔ Т11 ⊔
 (ПЕРЕМЕННАЯ ⊔ X1 ⊔ I ⊔)
 (ПЕРЕМЕННАЯ ⊔ X2 ⊔ I ⊔)
 (ПЕРЕМЕННАЯ ⊔ X3 ⊔ I ⊔)
 (ОТНОШЕНИЕ ⊔ C4 ⊔ ССК ⊔ C ⊔ I ⊔
 X1 ⊔ VX ⊔ : ⊔
 X2 ⊔ VX ⊔ : ⊔
 X3 ⊔ VHX ⊔ : ⊔)).

В данном примере список изменений содержит три описания переменной и одно описание отношения. С описаниями переменных и отношений можно подробнее ознакомиться в следующих пунктах.

3.2. Описание переменной

Описание переменной содержит:

1. Одно из слов ПЕРЕМЕННАЯ, ПЕРЕМЕННЫЕ, ПЕР.
2. Идентификатор, который становится названием переменной.
3. Длину переменной.

Длина переменной равна числу ячеек памяти, необходимых для записи значения переменной. Если переменная не занимает целое число ячеек, то длина переменной получается округлением до большего целого.

Описание переменной имеет следующую структуру:

(ПЕРЕМЕННАЯ ⊔ < идентификатор > ⊔ < длина переменной > ⊔).

Пример:

(ПЕРЕМЕННАЯ \lfloor XI \lfloor 2 \lfloor).

Несколько описаний переменных может быть объединено и записано более коротко, например:

(ПЕР \lfloor XI \lfloor I \lfloor X2 \lfloor I \lfloor XU \lfloor 4 \lfloor).

3.3. Описание отношения

Описание отношения содержит:

1. Слово ОТНОШЕНИЕ
2. Идентификатор, который становится названием отношения.
3. Тип модуля, выраженный одним из следующих слов:

МОДЕЛЬ,
ТАБЛИЦА,
ФОРМАТ,
ССК,
УРАВНЕНИЕ.

4. Название модуля.

5. Степень функциональности - число, указывающее, сколько переменных вычисляется по данному отношению.

6. Список связанных переменных, в котором для каждой связанной переменной указано:

- а) название переменной,
- б) тип связи, выраженный одним из следующих слов:

VX,
VYX,
СИЛ,
СЛ;

в) название соответствующей внешней переменной модели или ":". Название переменной задается только для модулей типа "модель", для остальных типов модулей задается ":".

Пример списка связанных переменных для модуля типа "модель":

ВЕС \lfloor VX \lfloor Г \lfloor ЦЕНА \lfloor VX \lfloor Ц \lfloor СУММА \lfloor VYX \lfloor С \lfloor .

Пример списка связанных переменных для модуля другого типа:

НАПР ВХ : СОПР ВХ : ТОК ВЫХ : .

Описание отношения в целом имеет следующую форму:

(ОТНОШЕНИЕ <идентификатор> <название типа модуля>
 <название модуля> <степень функциональности>
 <список связанных переменных >).

В качестве примера приведем описание отношения СИ, с использованием модуля А (см. п. 1.3), который имеет тип "модель":

(ОТНОШЕНИЕ СИ МОДЕЛЬ А I ИI СЛ И
UI СЛ У
PI СЛ Р).

3.4. Описание удаления

Переменная может исключаться из модели, если ее не связывает ни одно отношение. В этом случае задается описание следующего вида:

(ИСКЛ.ПЕР. <название переменной>).

Например, изменение:

(ИСКЛ.ПЕР. СУММА)

исключает переменную СУММА из той модели, в описание которой это изменение включено. Если такой переменной в модели нет, или она связана некоторым отношением, то это изменение не выполняется и печатается сообщение об ошибке в описании модели.

Из модели может исключаться любое имеющееся там отношение. Исключение отношения производится описанием следующего вида:

(ИСКЛ.ОТН. <название отношения>).

Например, изменение

(ИСКЛ.ОТН. ТОКИ)

исключает отношение ТОКИ из той модели, в описание которой это изменение включено. Если такого отношения в модели нет, то печатается сообщение об ошибке в описании модели.

3.5. Программное генерирование изменений к моделям

Вычислительные модели сами могут рассматриваться как значения, которые присваиваются переменным соответствующего типа. Это позволяет составлять и изменять ВМ не только транслятором СМП, но и всеми другими средствами СМП. В частности, для ВМ можно определить ряд операций, позволяющих строить новые ВМ по уже существующим.

Рассмотрим два макроса, параметрами которых являются вычислительные модели, их переменные и отношения. Эти макросы предназначены для изменения некоторой ранее выделенной ВМ, называемой вычислительной моделью задачи.

1. (ВВЕСТИ : УСИЛИТЕЛЬ / ПОД НАЗВАНИЕМ : У1 /
ПЕРЕИМЕНОВАВ : ВХОД1 / НА : СКОРОСТЬ / ВХОД2 / НА : КООРД /)

Этот макрос дополняет ВМ задачи копией ВМ УСИЛИТЕЛЬ, заменив предварительно названия переменных ВХОД1 и ВХОД2 на СКОРОСТЬ и КООРД. При этом все остальные вводимые переменные получают также новые названия. Переменная Вых получит название У1.Вых и т.д. Кроме того, соблюдается правило, что в ВМ задачи не включают двух переменных под одним названием. То есть, если ВМ задачи содержала ранее переменную СКОРОСТЬ или КООРД, то эта переменная будет связана с соответствующими отношениями из У1 (У1 - название созданной копии УСИЛИТЕЛЯ).

2. (УРАВНЕНИЕ / У4.Вых - 2 x У2.Вых - 0.1 x УСКОР = 0)

Этот макрос вводит в ВМ задачи новое отношение, связывающее переменные У4.Вых, У2.Вых и УСКОР. Эти переменные должны иметься в ВМ задачи.

Используя такие макросы, можно по набору имеющихся ВМ строить для некоторых задач такие ВМ, на которых составляется автоматически схема процесса решения. В этом случае на основе рассмотренных типов макросов и подходящего набора ВМ получается декларативный язык описания задач.

Такой принцип реализован в небольшом Универсальном Трансляторе Описаний Тривиальных задач - УТОПИСТ.

В этом языке имеются макросы приведенных выше типов. При описании уравнения арифметическое выражение записывается в обратной польской записи и последние символы, т.е. "=" опускаются. Разрешается только вводить идентификаторы, состоящие из одной или двух букв или буквы с цифрой.

Кроме рассмотренных выше макросов, описание задачи для программы УТОПИСТ содержит предложения следующего вида:

3. (ЗАДАЧА / НА/ТРЕУГ /)

где ТРЕУГ - название проблемы, по которой задача решается, определяющее исходный набор ВМ, выбираемых из архива.

4. (ЗАДАНО / X / I.0 / T1.Y1 / + I.5 /)

Здесь перечислены исходные данные задачи, т.е. переменные, которые известны, и их значения: $X = I.0$ и $T1.Y1 = + I.5$.

5. (ВЫЧИСЛИТЬ / ПП / T2.Y2 /)

является перечнем искомым переменных. В данном случае требуется вычислить ПП и T2.Y2.

В качестве примера рассмотрим задачу по тригонометрии, в которой используются стандартные обозначения ТРЕУГОЛЬНИК - модель треугольника, $U1, U2$ и $U3$ - углы, $C1, C2$ и $C3$ - стороны, ПП - площадь треугольника.

Задача. Найти площадь заштрихованного треугольника, если:

$$U : \Phi = 1 : 2;$$

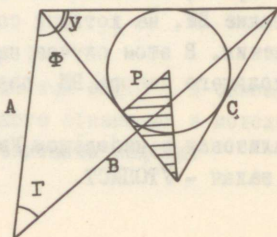
сторона заштрихованного треугольника Р равна радиусу вписанной во второй треугольник окружности;

$$A = 3,0;$$

$$B = 6,0;$$

$$C = 4,796;$$

$$\Gamma = 1,047.$$



Описание задачи для УТОПИСТА^{*})

(ЗАДАЧА / НА / ТРЕУГ /)

(ТРЕУГОЛЬНИК / Т1 / С1 / А / С2 / В / У2 / Ф / У3 / Г /)

(ТРЕУГОЛЬНИК / Т2 / С1 / С / У1 / У /)

(ТРЕУГОЛЬНИК / Т3 / С1 / Р /)

(УРАВНЕНИЕ / Е1 / 2 / У / ? / Ф / - /)

(УРАВНЕНИЕ / Е2 / Т1.С3 / Т2.С2 / - /)

(УРАВНЕНИЕ / Е3 / Т2.РА / - /)

(УРАВНЕНИЕ / Е4 / 2 / Т3.У1 / ? / Т2.У2 / - /)

(УРАВНЕНИЕ / Е5 / Т3.У2 / 1,571 / - /)

(ЗАДАНО / А / 3,0 / В / 6,0 / С / 4,796 / Г / 1,047 /)

(ВЫЧИСЛИТЬ / Т3.ПЛ /)

(СТОП /) КОНЕЦ

3.6. Синтаксис описания модели

1. < буква > :: = А | Б | В |

Г | Д | Е |

Ж | З | И |

Й | К | Л |

М | Н | О |

П | Р | С |

Т | У | Ф |

Х | Ц | Ч |

Ш | Щ | Ы |

Ю | Я | Ъ

2. < цифра > :: = 0 | 1 |

2 | 3 |

4 | 5 |

6 | 7 |

8 | 9

3. < идентификатор > :: = < буква > | < идентификатор >

< буква > | < идентификатор >

< цифра >

^{*}) В описании задачи опущен пояснительный текст перед двоеточиями и сами двоеточия, так как они транслятору не нужны (но и не помешали бы). Знак "?" обозначает умножение.

4. $\langle \text{целое без знака} \rangle ::= \langle \text{цифра} \rangle \mid \langle \text{целое без знака} \rangle \langle \text{цифра} \rangle$
5. $\langle \text{длина переменной} \rangle ::= \langle \text{целое без знака} \rangle$
6. $\langle \text{название переменной} \rangle ::= \langle \text{идентификатор} \rangle$
7. $\langle \text{описание переменной} \rangle ::=$
 $(\text{ПЕРЕМЕННАЯ} \mid \langle \text{название переменной} \rangle \mid$
 $\langle \text{длина переменной} \rangle)$
8. $\langle \text{тип связи} \rangle ::= \text{ВХ} \mid \text{ВЫХ} \mid \text{СЛ} \mid \text{СИЛ}$
9. $\langle \text{указатель внешней переменной} \rangle ::=$
 $\mid \langle \text{идентификатор} \rangle$
10. $\langle \text{описание связи} \rangle ::= \langle \text{название переменной} \rangle \mid$
 $\langle \text{тип связи} \rangle \mid \langle \text{указатель внешней переменной} \rangle$
11. $\langle \text{список связанных переменных} \rangle ::= \langle \text{описание}$
 $\text{связи} \rangle \mid \langle \text{список связанных переменных} \rangle$
 $\langle \text{описание связи} \rangle$
12. $\langle \text{тип модуля} \rangle ::= \text{МОДЕЛЬ} \mid \text{ТАБЛИЦА} \mid \text{ФОРМАТ ССК}$
 УРАВНЕНИЕ
13. $\langle \text{степень функциональности} \rangle ::= \langle \text{целое без знака} \rangle$
14. $\langle \text{название модуля} \rangle ::= \langle \text{идентификатор} \rangle$
15. $\langle \text{название отношения} \rangle ::= \langle \text{идентификатор} \rangle$
16. $\langle \text{описание отношения} \rangle ::= (\text{ОТНОШЕНИЕ} \mid \langle \text{название}$
 $\text{отношения} \rangle \mid \langle \text{тип модуля} \rangle \mid \langle \text{название модуля} \rangle \mid$
 $\langle \text{степень функциональности} \rangle \mid \langle \text{список связанных}$
 $\text{переменных} \rangle)$
17. $\langle \text{Описание удаления} \rangle ::=$
 $(\text{ИСКЛ.ПЕР.} \mid \langle \text{название переменной} \rangle \mid$
 $(\text{ИСКЛ.ОТН.} \mid \langle \text{название отношения} \rangle))$
18. $\langle \text{Описание} \rangle ::= \langle \text{описание переменной} \rangle \mid$
 $\langle \text{описание отношения} \rangle \mid$
 $\langle \text{описание удаления} \rangle$
19. $\langle \text{Продолжение описания} \rangle ::= \langle \text{описание} \rangle \mid$
 $\langle \text{продолжение описания} \rangle \mid \langle \text{описание} \rangle$
20. $\langle \text{определитель модели} \rangle ::= \text{НОВЫЙ} \mid \text{НОВАЯ} \mid$
 $\text{НОВОЕ} \mid \text{ПРОДОЛЖЕНИЕ}$

21. <название модели> : : = <идентификатор>
22. <шифр МЛ> : : = <идентификатор>
23. <описание модели> : : = (<определитель модели>
└ <название модели> └ <шифр МЛ> └ <продолжение
описания >).

4. ВЫЧИСЛЕНИЯ НА МОДЕЛЯХ

4.1. Макрокоманды

Отношение, содержащееся в ВМ, может использоваться для вычисления значений переменных, связанных этим отношением. Одно и то же отношение может использоваться в разных случаях по-разному. Например, отношение CI из модели, описанной в п. 1.3, может использоваться для вычисления по любому из алгоритмов:

$$U1 := J1 \times R1,$$

$$J1 := U1 / R1,$$

$$R1 := U1 / J1.$$

Вычисление по отношению задается в СМП всегда макрокомандой, так же как операции сложения, умножения и др. задаются вычислительной машине командами этой машины.

Макрокоманда состоит из кода операции и адресов. Кодом операции является название отношения. Адреса указывают, какие переменные являются исходными данными и результатами при выполнении макрокоманды.

Выполнение макрокоманды состоит в применении одного из алгоритмов, определенных названным отношением, к переменным, которые являются исходными данными макрокоманды. Макрокоманда выполняется, если могут быть вычислены значения всех ее результатов, т.е. если существует алгоритм, соответствующий данному отношению, который по исходным данным макрокоманды позволяет вычислять все ее результаты.

Адресами макрокоманды являются все названия переменных, связанных отношением этой макрокоманды. Названия переменных в адресах макрокоманды записываются в том порядке, в каком они заданы в описании отношения. Каждый адрес макрокоманды снабжен признаком, принимающим значения +, -, C или L.

Эти значения признаков обозначают, что переменная является соответственно входной, выходной, сильносвязанной или слабосвязанной для данной макрокоманды.

Входная переменная входит только в исходные данные макрокоманды;

выходная переменная является только результатом вычислений по макрокоманде;

сильносвязанная переменная входит в исходные данные и в результаты макрокоманды;

слабосвязанная переменная в вычислениях по макрокоманде не используется.

Макрокоманда имеет структуру:

(K \square t₁ \square A₁ \square t₂ \square A₂ \square ... t_l \square A_l \square)

Код операции

Признак 1-го адреса

1-й адрес

Признак 2-го адреса

2-й адрес

Признак l-го адреса

l-й адрес

Пример:

(A1 \square + \square I1 \square + \square T1 \square - \square P1 \square).

4.2. Макропрограмма

Макропрограмма является последовательностью макрокоманд и определяет вычислительный процесс на ВМ, заключающийся в выполнении вычислений по макрокомандам. Макрокоманды выполняются в порядке, в котором они записаны в макропрограмме. В конце макропрограммы записывается пустая макрокоманда (\square).

Пример:

(C1 \square + \square X1 \square + \square X2 \square - \square X3 \square)

(C2 \square + \square X3 \square + \square X4 \square - \square X5 \square)

(\square)

В программах, записанных на языке символического кодирования (в модулях типа ССК) макропрограмма записывается в виде последо-

вательности русских констант в коде М-2.

Пример:

```
090//КТРМ/(С1 +  
100//Х1 + Х  
110//2 - Х3  
120// )С2  
130//+ Х3 +  
140// Х4 -  
150//Х5 )(  
160//)
```

4.3. Вычислительная схема

Если в вычислениях на модели используются отношения, модули которых имеют тип ССК, или ТАБЛИЦА, то вычислительный процесс на ВМ можно описать просто перечислением отношений в таком порядке, в каком они должны использоваться. Последовательность названий отношений называется вычислительной схемой.

Модули перечисленных выше типов определяют один алгоритм, который определен уже названием отношения. Входные, выходные и сильно связанные переменные для каждого применения отношения в этом случае определены описанием модели, и макропрограмма составляется автоматически заменой каждого отношения в вычислительной схеме соответствующей макрокомандой.

Вычислительная схема задается в виде:

$$(A_1 \sqsubset A_2 \sqsubset \dots, A_K \sqsubset),$$

где A_1, A_2, \dots, A_K - названия отношений.

Пример:

$$(C1 \sqsubset C2 \sqsubset C3 \sqsubset).$$

В программах, записанных на языке символического кодирования, вычислительная схема записывается в виде последовательности русских констант в коде М-2.

Пример:

```
220 // КТРМ / (С1 С2  
230 /// С3 )
```

4.4. Задачи на ВМ

Каждое обращение к вычислительной модели в СМП называется задачей на ВМ. Задача на ВМ может состоять из 4-х частей:

1. Присвоение значений переменных в модели.
2. Составление макропрограммы.
3. Вычисление по макропрограмме.
4. Считывание значений переменных модели и присвоение этих значений переменным управляющей программы.

В разных задачах некоторые из этих частей могут отсутствовать. В частности, возможно производить только запись или считывание значений, выполнять только вычисление по макропрограмме и т.д.

По способу описания вычислительного процесса задачи делятся на 3 вида:

- 1) вычисление по готовой макропрограмме;
- 2) вычисление по вычислительной схеме;
- 3) задачи с автоматическим планированием вычислительного процесса по заданным названиям исходных данных и результатов расчета.

Первые два вида задач различаются только по подробности описания вычислительного процесса на ВМ. Вычислительная схема и макропрограмма определяют однозначно вычислительный процесс на ВМ, в то время как для одной и той же задачи третьего вида могут быть составлены разные макропрограммы.

Рассмотрим подробнее задачу с автоматическим планированием решения на примере модели, которая имеет схему рис. II.

Любую задачу такого вида можно по существу описать, перечислив исходные переменные и переменные, которые следует вычислить. Пусть исходными будут переменные X1, X2, X3 и вычислить необходимо X6. Эту задачу можно решить при помощи любого из четырех вычислительных процессов, которые описываются вычислительными схемами:

$$(C2 \text{---} C6 \text{---} C4 \text{---} C5 \text{---}),$$

$$(C6 \text{---} C2 \text{---} C4 \text{---} C5 \text{---}),$$

$$(C1 \text{---} C2 \text{---} C3 \text{---}),$$

$$(C2 \text{---} C1 \text{---} C3 \text{---}).$$

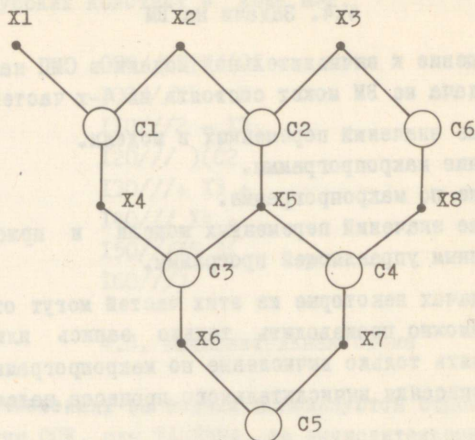


Рис. II

Первые два перечисленных процесса дадут обязательно одинаковый результат. Третий и четвертый вычислительные процессы могут дать результат, отличающийся от первых. При автоматическом планировании решения автоматически составляется и выполняется один из вычислительных процессов, который решает задачу.

4.5. Описание задачи

Описание задачи содержит:

- 1) тип задачи, указывающий, как производятся вычисления;
- 2) управление средой, образованной моделями;
- 3) название модели, на которой задача решается;
- 4) список переменных модели, являющихся исходными данными или результатами расчета;
- 5) список переменных модуля, в котором задача описывается;
- 6) название вычислительной схемы или макропрограммы.

Описание задачи задается оператором обращения к модели и состоит из 2 частей: тело оператора и текст алгоритма, которые мо-

гут записываться раздельно, в разных местах символической программы. В некоторых случаях текст алгоритма отсутствует.

Тело оператора имеет вид:

```
( < тип задачи > ┘ < управление средой > ┘  
  < название модели > ┘ < список переменных модели > )  
  < список внутренних переменных модуля >  
  < метка алгоритма >
```

или

```
(КОМ ┘ < управление средой > ┘ < название модели > ┘  
  < макрокоманда > ┘ )
```

Тип задачи:

- ВЫП - на указанной модели решается задача с автоматическим планированием решения, алгоритм отсутствует,
- ОПС - решается задача по заданной вычислительной схеме,
- ПР - решается задача по заданной макропрограмме;
- ЗАП - выполняется передача значений переменных между моделью и модулем, ничего не вычисляется,
- ВЫТР - планирование решения происходит во время трансляции аналогично типу ВЫП, задача решается по составленной при трансляции макропрограмме,
- ОПТР - во время трансляции по заданной вычислительной схеме составляется макропрограмма, задача решается по этой макропрограмме.

Оператор обращения КОМ означает выполнение одной макрокоманды, заданной в самом операторе обращения.

Управление средой производится словами ПРВ, ПРД, ЗМК, которые определяют, какая модель вызывается для выполнения оператора обращения к модели и какая остается там после его завершения.

Модель, находящаяся в оперативной памяти, непосредственно доступна для выполнения вычислений, на ее отношения и переменные можно ссылаться в макрокомандах и вычислительной схеме. Такая модель называется активной.

ПРВ означает, что делается активным новый экземпляр модели, названной в операторе обращения к модели. После выполнения оператора остается активной эта же модель.

ПРД выполняет оператор обращения на вызванной заранее модели, и активность модели не изменяется.

ЗМК делает активной новую модель, выполняет оператор обращения к модели, а после выполнения активизирует прежнюю модель.

Передача данных из модуля в модель и обратно задается списком переменных модели и списком переменных модуля.

Список переменных модели имеет вид:

VX \lfloor <название переменной модели> или

VYX \lfloor <название переменной модели>

VX означает присваивание значения модельным переменным до выполнения оператора обращения, а VYX - присваивание значения переменным модуля после выполнения оператора обращения.

Переменные модуля задаются в виде списка этикеток (в ССК).

Метка алгоритма также - этикетка ССК.

Пример:

```
010 // КТМ / (ВЫП $\lfloor$ П
020 /// РД $\lfloor$ ММ1
030 ///  $\lfloor$  VX $\lfloor$  X1
040 ///  $\lfloor$  VX $\lfloor$  X2
050 ///  $\lfloor$  VYX $\lfloor$  Y
060 /// I $\lfloor$  VYX
070 /// Y2 $\lfloor$ )
080 // ПСК / ОСН1, ОСН2, ОСН3
090 /// ОСН4
```

Здесь описана задача с автоматическим планированием решения, в которой по переменным X1 и X2 необходимо вычислить Y1 и Y2. Между этими переменными и переменными основной программы задано следующее соответствие:

```
X1 ... ОСН1,
X2 ... ОСН2,
Y1 ... ОСН3,
Y2 ... ОСН4.
```

4.6. Управляющая программа

При решении задач на моделях управляющей программой называется та программа, в которой содержатся описания этих задач. Управляющая программа, как правило, сама является модулем, записанным на языке символического кодирования.

Каждой задаче на ВМ соответствует в управляющей программе один оператор обращения к модели. В рационально составленной управляющей программе все основные вычисления выполняются на вычислительной модели. Блок-схема такой программы содержит блоки ввода-вывода, логические блоки и блоки вычислений на модели.

При составлении управляющей программы желательно иметь в виду, что однажды автоматически составленная макропрограмма становится доступной управляющей программе, если составление происходит во время трансляции.

Динамическое распределение памяти позволяет использовать оперативную память последовательно для разных программ и массивов. Этим обеспечивается возможность решения задач на разных моделях, а также на разных экземплярах одной и той же модели по одной управляющей программе.

Обслуживающие программы СМП позволяют прерывать вычисления на одной модели вычислениями на других моделях и потом, восстановив содержимое оперативной памяти, продолжать прерванные вычисления.

Желательно избегать лишнего переписывания переменных из основной программы в модель и обратно. Особенно надо избегать частой смены моделей, так как это связано с многократным обращением к магнитным лентам. Поэтому управляющую программу полезно составлять так, чтобы вычисления на одной модели наиболее редко прерывались вычислениями на других моделях.

При составлении управляющей программы необходимо также иметь в виду, что внешние переменные модуля не могут внутри данного модуля использоваться в описаниях задач на модели.

Л и т е р а т у р а

1. Integrated Civil Engineering System, MTI, 1965, AFIPS Comp. Conf.
2. Ross D.T., The AED Approach to Generalised Computer-Aided Design. Proc. ACM 22nd Natl. Conf. 1967, pp. 367-385.
3. Система модульного программирования для ЭВМ "Минск-22". Инструкция использования. НИПТИ, Таллин, 1970 (ротап rint).
4. Система табличного программирования. Научно-технический отчет ПТНИИ по теме 72-129-2. Таллин, 1969 (рукопись).
5. Столяр Г.К., Ковалевич Э.В., Курусь З.С. Система символического кодирования для ЭВМ "Минск-2 (22)", часть I, II, III. Минск, 1967.

СОДЕРЖАНИЕ

	Стр.
I. ОБЩЕЕ	3
I.1. Введение	3
I.2. Возможности использования СМП	5
I.3. Основные понятия	7
I.4. Общие сведения о работе обслуживающих программ	14
2. МОДУЛИ	18
2.1. Машинное представление модулей	18
2.2. Таблицы	20
2.3. Программы	24
2.4. Форматы	26
2.5. Уравнения	28
2.6. Модули связи	29
3. СОСТАВЛЕНИЕ МОДЕЛЕЙ	31
3.1. Описание модели	31
3.2. Описание переменной	32
3.3. Описание отношения	33
3.4. Описание удаления	34
3.5. Программное генерирование изменений к моделям	35
3.6. Синтаксис описания модели	37
4. ВЫЧИСЛЕНИЯ НА МОДЕЛЯХ	38
4.1. Макрокоманды	38
4.2. Макропрограмма	41
4.3. Вычислительная схема	42
4.4. Задачи на ВМ	43
4.5. Описание задачи	44
4.6. Управляющая программа	46
Л и т е р а т у р а	48

ОПЕЧАТКИ.

Стр.	Строка	Напечатано	Следует читать
17	2 снизу	M_1	M_1'
18	14 снизу	$\alpha + \alpha - 7000$, где $\alpha -$	$\alpha + \alpha_H - 7000$, где $\alpha_H -$
26	2 сверху	0-12	1-12
27	10 сверху	(СУММА ДЕС - 5)	(СУММА ДЕС - 4)
38	16 сверху	= МОДЕЛЬ ТАБЛИЦА ФОРМАТ ССК	= МОДЕЛЬ ТАБЛИЦА ФОРМАТ ССК
46	12 снизу	060 /// 1 - Вых	060 /// 1 - Вых

