

И - 208

82-89

На правах рукописи

Иванов Юрий Николаевич

РАЗРАБОТКА И РЕАЛИЗАЦИЯ
СРЕДСТВ ПРОГРАММИРОВАНИЯ ПРИКЛАДНЫХ ЗАДАЧ
В СИСТЕМАХ МАТЕМАТИЧЕСКОГО ОБЕСПЕЧЕНИЯ
СПЕЦИАЛИЗИРОВАННЫХ ДИАЛОГОВЫХ КОМПЛЕКСОВ

01.01.10 - математическое обеспечение
вычислительных машин и систем

А в т о р е ф е р а т
диссертации на соискание ученой степени
кандидата физико-математических наук

Дубна 1982

Работа выполнена в Институте физики высоких энергий (г.Серпухов).

Научный руководитель: кандидат технических наук Г.М.Максимов.

Официальные оппоненты: доктор физико-математических наук И.Н.Силин,
кандидат физико-математических наук О.Б.Арушанян.

Ведущее научное учреждение: Институт прикладной математики АН СССР
(г. Москва).

Защита диссертации состоится "25" декабря 1982 г. в 10.30
часов на заседании специализированного совета Д 047.01.04. при Лаборато-
рии вычислительной техники и автоматизации Объединенного института ядер-
ных исследований (г.Дубна Московской обл.).

С диссертацией можно ознакомиться в библиотеке ОИЯИ.

Автореферат разослан "23" ноября 1982 г.

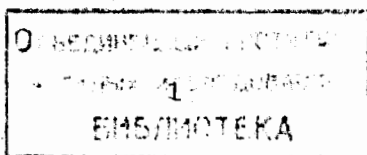
Ученый секретарь
специализированного совета
кандидат физ.-мат. наук

Шб.
З.М.Иванченко

Актуальность проблемы. Настоящая работа посвящена решению проблем разработки и реализации средств программирования прикладных задач в различных по технической оснащенности диалоговых комплексах, предназначенных для обработки фотографий с больших пузырьковых камер — уникальных физических установок, используемых в экспериментах по изучению субатомного строения материи. Такие комплексы включают парк специализированных диалоговых терминалов — просмотрно-измерительных проекторов, подключенных к центральной ЭВМ (ЦЭВМ), имеющей соответствующее математическое обеспечение (МО). МО подразделяется на системную и прикладную части. В прикладной части МО реализуются программы сопровождения операторов проекторов, охватывающие разнообразные алгоритмы обработки фильмовой информации, получаемой с различных камер и в разных экспериментах. Размеры таких программ и их данных достигают десятков К слов ЦЭВМ. При этом они должны обеспечивать время реакции на большинство посылок с терминала не более 1–2 с.

Выполнить объем обработки, достигающий сотен тысяч фотографий в год, можно на парке высокоавтоматизированных проекторов из 20–30 штук. В этом случае потребность в оперативной памяти ЦЭВМ может многократно превышать ее размер. Но недогрузка процессора и внешних устройств позволяет выполнять параллельно задания других пользователей. Отсюда следует, что МО реального времени на ЦЭВМ целесообразно создавать в виде специализированной системы разделения времени (СРВ), включающей специализированную операционную систему (СОС) и совокупность выполняемых под ее контролем прикладных программ.

В задачу СОС входит организация режима мультидоступа для обслуживаемых абонентов путем моделирования виртуальной вычислительной машины для каждого абонента. Такой подход позволяет оптимизировать распределение ресурсов ЦЭВМ при обслуживании терминалов в рамках задания, выполняемого под контролем штатной операционной системы (ОС), но по существу он закрывает возможность использования в неизмененном виде средств реализации языков программирования высокого уровня из штатного МО ЦЭВМ. Сложность программирования на языке ассемблера при этом возрастает из-за необходимости учета особенностей нестандартного терминала и структуры виртуальной машины абонента. Поэтому для выполнения необходимого объема программирования алгоритмов сопровождения, соблюдения сроков обработки



экспериментов, а также предоставления возможности параллельного использования ЦЭВМ в других программных проектах решение проблемы разработки и реализации средств программирования прикладных задач приобретает особую актуальность.

При разработке МО важное значение имеет проектирование системы в целом и отдельных ее компонентов. Вместе с тем, конкретные пути проектирования средств программирования для прикладных задач, выполняемых в специализированных СРВ, мало исследованы. Вопрос этот актуален и для создания новых алгоритмических языков, и для модернизации и реализации в СРВ существующих языков.

Обычно проектирование структуры СРВ производится в сжатые сроки (чтобы быстрее подключить прикладных программистов), что жестко ограничивает время проектирования и разработки средств программирования. Эта работа должна быть, в основном, завершена к моменту окончания разработки блок-схем первых прикладных программ. Актуальность решения проблемы создания средств программирования возрастает и в связи со сложностью оценки ресурсоемкости, времени реакции и других важных характеристик будущих прикладных программ (особенно в условиях новой для разработчиков вычислительной техники, изменяющегося числа обслуживаемых абонентов и т.п.).

Состояние вопроса. В настоящее время во многих физических центрах как в СССР, так и за рубежом функционируют комплексы обработки снимков на проекторах, работающих в линию с ЭВМ. Они отличаются аппаратной базой, мощностью и загруженностью используемых ЭВМ, общей структурой и построением МО, идеологией контроля и управления работой операторов. В качестве основных языков программирования обычно используют язык ассемблера и ФОРТРАН, хотя они не адекватны задачам создания программ сопровождения операторов. В частности, они не содержат средств программирования диалога между человеком и ЭВМ. Другие неудобства связаны с необходимостью учета при программировании возможности обслуживания одной программой нескольких абонентов и сложностью организации защиты данных, относящихся к разным абонентам. Язык ассемблера применяется на средних ЭВМ второго поколения (типа БЭСМ-4), а ФОРТРАН — на достаточно мощных ЭВМ третьего поколения.

Распределением ресурсов ЭВМ, необходимых для выполнения прикладных программ на ФОРТРАНе, занимается штатная ОС. Но ее универсальность не позволяет в полной мере учесть специфику прикладных программ, что приводит по мере увеличения числа обслуживаемых абонентов ко все более непроизводительному использованию ЦЭВМ.

Специализированные языки и версии ФОРТРАНа для программирования прикладных задач в аналогичных комплексах не применялись.

Цель работы. Диалоговые комплексы, рассматриваемые в диссертации, включают ЦЭВМ третьего поколения ICL-1903A и DEC-10. Использование таких ЭВМ, обладающих развитым матобеспечением, и специфика обработки снимков выдвинули ряд новых задач по разработке структуры эффектив-

ной и экономичной СРВ. К ним относилась и задача разработки и реализации средств автоматизации программирования прикладных программ (САП). Ее решение должно было способствовать созданию эффективной СРВ и обеспечить:

- возможность создания эффективных прикладных программ;
- простоту модернизации средств программирования для обеспечения их большей адекватности задачам создания прикладных программ;
- относительно низкую трудоемкость разработки и реализации САП, максимальное использование возможностей, предоставляемых штатным МО ЦЭВМ;
- достаточную степень удобства для прикладных программистов.

Разработка и реализация САП осуществлялись автором в системе МО FAUST на ЦЭВМ ICL-1903A и в системе МО COSDES на ЦЭВМ DEC-10. Создание данных систем предусматривалось планом организации в ИФВЭ мощного измерительно-вычислительного центра.

Научная новизна. Выработан подход к решению проблемы создания средств программирования прикладных задач в специализированных диалоговых комплексах, позволяющий на основе штатного математического обеспечения ЭВМ эффективно создавать адекватные этим задачам специализированные языковые средства для их программирования, средства трансляции и выполнения программ, написанных на этих языках.

Разработаны специализированные средства программирования прикладных задач обслуживания абонентов в просмотрово-измерительных комплексах ИФВЭ:

- автокод FAUCOD, ориентированный на программирование задач сопровождения операторов, для ЭВМ ICL-1903A;
- диалект языка ФОРТРАН, модифицированный для программирования задач обслуживания управляющих ЭВМ проекторов в многомашинном комплексе с ЦЭВМ DEC-10.

Обеспечена реализация языков программирования прикладных задач:

- на основе предложенного автором метода программных целей создана система трансляции с FAUCOD;

- решена задача трансляции программ на ФОРТРАНе в реентерабельные процедуры;

- разработаны и реализованы средства, позволяющие организовать работу набора реентерабельных программ на ФОРТРАНе в рамках одного штатного задания (программы) на ЭВМ DEC-10.

Поставлена и решена задача автоматического совмещения локальных переменных фортранных программ, исследована применимость методов решения к различным версиям ФОРТРАНа.

Реализация. Язык FAUCOD и система трансляции для него эксплуатируются в ИФВЭ с 1973 г. На FAUCOD написаны, отлажены и эксплуатируются свыше 40 модулей прикладных программ, модулей диспетчера СРВ и СОС, модулей проверки исправности аппаратуры комплекса. В совокупности с другими компонентами матобеспечения была обеспечена обработка свыше 450 000 событий с камеры "Мирабель".

Средства реализации ФОРТРАНа в системе COSDES созданы и эксплуатируются с 1978 г. Они позволили создать на ФОРТРАНе программы сопровождения для 15-футовой камеры FNAL в экспериментах E-180 и E-564 и обработать свыше 90 000 фотографий, а также для камеры BEBC в эксперименте pp, в котором на апрель 1982 г. обработано около 3 000 снимков.

В составе программного обеспечения, используемого для реализации FAUCOD, автором написано около 15 000 команд на языке ассемблера. При реализации средств, обеспечивших возможность программирования на ФОРТРАНе прикладных задач в системе COSDES, автором написано около 480 предложений на языке редактора TECO и 1 000 команд на языке ассемблера. Программы, реализующие автоматическое совмещение локальных переменных, включают 890 предложений языка TECO.

Практическая ценность. Результаты диссертации могут быть использованы при создании средств программирования прикладных задач в аналогичных комплексах обработки फिल्मовой информации, а также в других диалоговых и управляющих комплексах, где время и память являются критическими ресурсами применяемых ЭВМ. Метод программных цепей может быть применен в операционных системах общего назначения или в частных задачах, использующих значительные объемы промежуточных данных. Предложенный подход к получению разделяемых фортранных программ может быть реализован на машинах с достаточным количеством индексных регистров в компиляторах или постпроцессорах к ним. Его реализация позволяет получать преимущества от применения разделяемых программ и одновременно использовать преимущества программирования на языке высокого уровня. Метод совмещения локальных переменных в фортранных программах может быть реализован в компиляторах и редакторах связей в любой системе программирования на ФОРТРАНе.

Апробация и публикации. Основные результаты диссертации докладывались на Всесоюзных семинарах по обработке физической информации (Ереван, 1975, 1977), II Всесоюзном совещании по автоматизации научных исследований в ядерной физике (Алма-Ата, 1978), II Всесоюзном совещании "Диалоговые вычислительные комплексы" (Серпухов, 1979), на семинарах ИФВЭ, ЛВТА ОИЯИ и МГУ им. М.В.Ломоносова. Они также опубликованы в работах /1-8/.

Объем работы. Диссертация состоит из введения, пяти глав, заключения, библиографии и приложения. Объем работы составляют 112 страниц текста, 18 рисунков и 12 таблиц. Библиография включает 77 наименований.

СОДЕРЖАНИЕ ДИССЕРТАЦИИ

В первой главе рассматриваются основные проблемы создания языкового обеспечения как совокупности предоставляемых языков программирования. Они тесно связаны с общей проблемой создания МО специализированных диалоговых комплексов (СДК), что в свою очередь определяется аппаратной и

программной базой комплексов. Рассматриваются основные компоненты систем МО FAUST и COSDES, решаемые ими задачи в СДК на базе ЭВМ ICL-1903A и более мощной DEC-10.

Важнейшей частью матобеспечения СДК является система разделения времени. Основными принципами построения СРВ являются: модульное программирование прикладных программ, организация разделяемых (реентерабельных) модулей, отделение кодов от данных, организация очередей на обращение к внешним устройствам и т.д. Более подробно рассматриваются принципы, наиболее важные для разработки и реализации средств программирования. Структура СРВ является определяющей при решении проблемы разработки и реализации языкового обеспечения. Чисто языковая сторона проблемы заключается в предоставлении языков программирования, средств трансляции и организации выполнения программ на этих языках под контролем СОС. Другая сторона заключается в стратегии распределения задач между системной и прикладной частями МО и обусловлена схожестью блок-схем многих прикладных программ и общим набором присущих им рутинных операций. Так, программирование некоторых частей программ сопровождения можно формализовать обращением к специальным подпрограммам общего назначения или к модулям СОС. Часть задач общего характера допускает вариации в плане их распределения, решение этого вопроса определяется мощностью используемых ЦЭВМ и другими обстоятельствами.

Системы FAUST и COSDES дают примеры различного решения проблемы создания средств программирования. В FAUST для программирования прикладных программ разработан специализированный язык, а стратегия распределения задач заключается в максимальном упрощении программирования прикладных программ за счет расширения функций СОС и увеличения возлагаемых на нее задач. В системе COSDES для программирования применяются язык ассемблера и специально разработанный диалект языка ФОРТРАН, стратегия распределения задач определяется стремлением к максимальной экономии ресурсов (оперативной памяти, времени процессора и т.д.) за счет совместных усилий разработчиков СОС и прикладных программистов.

Однако решения проблемы имеют и общие черты. С языковой стороны это, например, разработка специальных средств программирования диалога, в распределении задач – возможность написания прикладной программы из расчета обслуживания только одного абонента. Мультиобслуживание с помощью таких программ обеспечивается СОС. При этом прикладной программист также автоматически освобождается от задачи защиты данных различных абонентов – она решается средствами реализации языкового обеспечения и СОС.

Вторая глава посвящена описанию автокода FAUCOD. Этот язык предназначен для решения следующих основных задач: организации модульного программирования прикладных программ; сегментной организации данных; организации диалога прикладной программы и оператора проектора; защиты данных и программ. Автокод выполнен в виде макрорасширения языка ассемблера ЦЭВМ, у которого унаследован синтаксис части предложений FAUCOD.

Основными понятиями FAUCOD являются понятия процесса, программы сопровождения, модуля, рабочего поля данных (сегмента данных), типа модуля, команды, директивы. Модулем является логически замкнутая, соответствующим образом оформленная часть программы сопровождения. Модули могут быть разделяемыми и неразделяемыми, в зависимости от этого они имеют разный тип.

Обращение к данным осуществляется указанием в операнде двумерного адреса, содержащего номер поля данных и смещение относительно начала поля. Команды обращения к данным разворачиваются транслятором с FAUCOD обычно в три команды языка ассемблера. Одна из них формирует смещение, другая представляет собой вызов специальной подпрограммы, проверяющей корректность двумерного адреса и наличие поля данных в оперативной памяти, третья команда выполняет соответствующие действия с указанной ячейкой.

Две специальные команды предназначены для приема данных от оператора и для вывода ответной информации на соответствующие устройства проектора. Виртуальная FAUCOD-машина (вычислительная машина, моделируемая с помощью СОС для выполнения предложений FAUCOD) выполняет эти команды, используя контрольный массив, содержащий информацию о текущем состоянии устройств проектора, адреса, областей ввода данных и т.д.

Команды передачи управления позволяют организовать переходы типа GO TO, CALL, RETURN как внутри модулей, так и между ними. Для обеспечения возможности ликвидации последствий сбоев ЭВМ и ошибок операторов введены команды ENER и EXER, позволяющие организовать передачу управления на начало определенного этапа обработки фотоснимка, пройденного оператором ранее.

Для повышения эффективности прикладных программ необходимо учитывать некоторые особенности программирования, связанные со структурой FAUCOD-машины и организацией мультиобслуживания абонентов на ICL-1903A. Например, части программ диалогового характера рекомендуется оформлять как разделяемые модули, а вычислительного — как неразделяемые. Рекомендуется реже осуществлять переходы из модуля в модуль, чаще использовать возможность блочной передачи данных между рабочими полями и т.д.

В третьей главе рассматривается реализация специализированного автокода FAUCOD на ЭВМ ICL-1903A. Она включает виртуальную машину языка и систему трансляции модулей прикладных программ.

Специфика выполнения команд FAUCOD заключается в том, что команды выполняются в условиях мультиобслуживания. Относительно часто используемые команды выполняются посредством резидентных частей СОС (организация диалога, загрузка полей в оперативную память и т.д.). Более громоздкие части FAUCOD-машины, выполняющие редкие команды (копирование и восстановление данных процесса, останов процесса), загружаются в оперативную память лишь при необходимости.

Файлы, используемые при реализации FAUCOD, находятся на устройствах прямого доступа (диск, барабан). Виртуальная FAUCOD-машина использует два файла. Один из них служит для освобождения места в оперативной памяти.

ти, поскольку суммарный объем данных всех процессов и модулей программ сопровождения многократно превышает ее размер. В другом файле размещаются копии атрибутов процессов, выполненные в указанных точках процесса по команде ENER и используемые при восстановлении их командой EXER. Система трансляции модулей использует файлы, содержащие библиотеки: двоичных форм модулей, перемещаемых форм подпрограмм, форматов блоков данных.

Трансляция модулей осуществляется набором из 5-7 программ: транслятор с FAUCOD, штатный ассемблер, штатный редактор связей, библиотекарь и т.д. Для получения двоичной формы модуля достаточно организовать последовательную прогонку этих программ, взаимодействующих между собой с помощью файлов промежуточных данных, т.е. файлов, в которые одни программы только пишут, а другие - только читают. В связи с необходимостью частого повторения и длительностью такой процедуры был разработан метод, позволивший эффективнее решать задачу трансляции модулей.

Уточним термин "последовательность программ" и введем понятия звена, цепи и матрицы программ. Совокупность программ

$$П_1, П_2, \dots, П_n \quad (n > 1) \quad (1)$$

назовем последовательностью программ, если часть выходных файлов программ $П_1, П_2, \dots, П_\ell$ относится к числу входных файлов программы $П_{\ell+1}$, $\ell = 1, \dots, n-1$. Протокол чтения программы $П$ из входных файлов Φ_1, \dots, Φ_m ($m \geq 1$) определяется как последовательность пар $\{(i_s, j_s)\}$ $m \geq i_s \geq 1, j_s \geq 1$, в которой зафиксировано, что s -ой командой чтения является команда чтения из файла Φ_{j_s} физической порции данных с номером i_s . Аналогично определяется протокол записи.

Последовательность из двух программ $П_1, П_2$ назовем звеном, если протокол записи программы $П_1$ в выходные файлы промежуточных данных совпадает с протоколом чтения этих файлов программой $П_2$. Целью назовем последовательность (1), в которой $П_\ell, П_{\ell+1}$ является звеном для каждого $\ell = 1, \dots, n-1$. В данной главе рассмотрен алгоритм работы цепи программ, при котором обмены с файлами промежуточных данных заменяются непосредственной передачей данных. Используемая в задаче трансляции модулей последовательность программ не является цепью, но допускает разбиение на цепи, которые удобно записать в виде строк матрицы:

$$\begin{pmatrix} П_1 & \dots & \dots \\ \dots & \dots & \dots \\ \dots & \dots & П_n \end{pmatrix}. \quad (2)$$

Матрица (2) формируется в зависимости от параметров трансляции и передается управляющей программе, которая организует работу указанных в ней цепей. Реализация этого метода не требует модификации штатных программ, используемых в системе трансляции, а с другой стороны дает новые возможности в программировании специальных программ системы трансляции.

В четвертой главе рассматривается решение проблемы реализации ФОРТРАНа в системе разделения времени COSDES на ЦЭВМ DEC-10. Эта система, как и CPB FAUST, работает на ЭВМ в рамках одного задания и

организует мультиобслуживание абонентов. При этом обязательно отделение данных от кодов и организация кодов в виде разделяемых процедур. Для размещения данных каждого абонента отводится одно рабочее поле, имеющее фиксированное начало в адресном пространстве CPB. Обращение к данным в прикладных программах на языке ассемблера производится путем использования специально зарезервированного (базового) индексного регистра, в который диспетчер CPB загружает перед запуском прикладной программы адрес начала (базу) рабочего поля текущего процесса.

Чтобы иметь возможность включения прикладных программ на ФОРТРАНе в CPB COSDES, необходимо было решить следующие задачи:

- организовать работу фортранной программы с данными посредством базы рабочего поля;
- разместить переменные подпрограмм фортранной программы в рабочем поле.

Анализ кодов, генерируемых штатным компилятором, показал, что они допускают несколько видов модификаций, позволяющих решить эти задачи. Суть модификаций заключается в изменении адресной части некоторых кодов и констант. Переменным присваиваются фиктивные адреса в некотором условном рабочем поле, а при работе программы ввиду выполненной модификации они связываются посредством базового регистра с адресами в рабочем поле нужного процесса. Постпроцессор, осуществляющий модификацию, использует создаваемый компилятором файл, содержащий текст кодов на языке ассемблера, эквивалентных соответствующим операторам ФОРТРАНа. В задачу постпроцессора входит также оформление кодов в виде синтаксически правильной программы на языке ассемблера.

В качестве языка написания постпроцессора использован язык редактора текстовых файлов ТЕСО. В связи с этим были разработаны методы программирования на ТЕСО, позволяющие повысить эффективность программ. В частности, их применение позволило примерно в два раза увеличить скорость работы постпроцессора по сравнению с первоначальным вариантом.

Способ реализации ФОРТРАНа в системе COSDES фактически определил диалект этого языка, используемый для программирования прикладных задач. Но он не только наложил некоторые ограничения на язык (впрочем, несущественные для данного класса задач), а и позволил внести в него добавления, учитывающие специфику таких задач. К ним относится возможность ввода/вывода на управляющую ЭВМ проектора стандартными операторами READ и WRITE, COMMON — блок с динамически изменяющимся размером занимаемой виртуальной памяти. Этот блок предусмотрен для формирования выходного массива программы сопровождения, увеличивающегося по мере обработки снимка и сокращаемого после сброса этого массива в дисковый файл.

В пятой главе рассматривается задача повышения эффективности использования памяти в программах на ФОРТРАНе за счет автоматического совмещения локальных переменных подпрограмм. Она возникла в связи с потребностью экономии памяти рабочего поля абонента в системе COSDES, но имеет и самостоятельное значение. Отметим, что задача экономии памяти в про-

граммах на ФОРТРАНе целиком возлагается на программиста, в то время как во многих распространенных языках она решается автоматически, с помощью специально предоставляемых средств. Подобным образом было бы желательно реализовать и локальную память в ФОРТРАНе, сохранив вместе с тем положительные стороны статической памяти этого языка — простоту ее реализации и быстрый доступ к значениям переменных, — обусловленные определением адресов переменных на этапе редакции связей программы.

Предлагаемый метод распределения памяти (именуемый адресно-статическим) основан на рассмотрении графа фортранной программы с весами на ребрах, равными количеству локальных переменных в вызывающих подпрограммах. В главе отдельно рассматриваются способы реализации адресно-статического распределения памяти для программ, использующих и не использующих оператор EXTERNAL. В первом случае удается в явном виде построить граф в виде слова в специальном алфавите и определить для каждой подпрограммы такие необходимые для дальнейшего данные как максимальная глубина ее вызова и множество всех вызывающих ее подпрограмм. Во втором случае для данной фортранной программы строится модель — специальная программа на ФОРТРАНе, имеющая граф, изоморфный графу моделируемой программы. Необходимые данные вычисляются при работе модели и используются затем специальной версией постпроцессора компилятора.

Адресно-статический метод может быть распространен и на реализацию общих переменных (COMMON-блоков) программ на ФОРТРАНе. При рассматриваемом подходе переменные COMMON-блока имеют силу одновременно с локальными переменными специально определенной подпрограммы-вершины графа программы, из которой достижимы все вершины, соответствующие подпрограммам, содержащим описание данного COMMON-блока. В конце главы рассматриваются особенности конкретной реализации адресно-статического распределения для фортранных программ, работающих в системе COSDES, и приводятся результаты его применения.

ЗАКЛЮЧЕНИЕ

Основными результатами диссертации являются разработка и реализация средств программирования прикладных задач в системах математического обеспечения двух просмотрово-измерительных диалоговых комплексов ИФЗ в системе FAUST, обслуживающей разнотипные приборы первой очереди на линии с центральной ЭВМ среднего класса ICL-1903A, и в системе COSDES, обслуживающей приборы второй очереди, оснащенные управляющими ЭВМ PDP-8/E, подключенными к большой ЭВМ DEC-10.

Участие автора в разработке и реализации диспетчера системы разделения времени, системы информационного обслуживания, прикладных программ и других компонентов матобеспечения специализированных диалоговых комплексов позволило с разных сторон проанализировать и решить проблему разработки и реализации средств программирования прикладных задач. Полученные результаты имеют самостоятельное научное и практическое значение.

1. Предложен способ создания машинно-ориентированного языка для программирования прикладных задач на ЭВМ среднего класса в виде входного языка специализированного макрогенератора языка ассемблера. При его реализации в системе FAUST:

- создан автокод FAUCOD для программирования прикладных задач, выполняемых под контролем специализированной операционной системы на ЭВМ ICL-1903A;
- написан транслятор FAUCOD - язык ассемблера;
- предложен способ разделения времени и управления памятью в виртуальной FAUCOD-машине, позволяющий осуществить выполнение предложений языка в условиях мультиобслуживания.

2. На основе разработанного метода программных цепей на ЭВМ ICL-1903A реализована система трансляции, преобразующая исходные тексты на FAUCOD в двоичную форму. В предложенной схеме могут работать без модификации штатные программы и программы системы трансляции, при этом за счет организации непосредственной передачи данных между программами вместо передачи данных через дисковые файлы примерно в два раза увеличилась скорость трансляции.

3. Решена задача организации разделяемых программ на ФОРТРАНе в рамках одного задания, работающего под контролем штатной операционной системы ЭВМ DEC-10. Предложен способ размещения кодов и данных прикладных программ в адресном пространстве системы разделения времени и организации защиты данных различных процессов. Созданы программные средства, обеспечивающие работу прикладных фортранных программ в системе математического обеспечения COSDES на ЭВМ DEC-10:

- постпроцессор, преобразующий исходные фортранные тексты в разделяемые коды;
- пакет подпрограмм, позволяющих осуществлять ввод/вывод на нестандартное устройство (управляющую ЭВМ проектора) с помощью операторов языка ФОРТРАН READ и WRITE.

Реализован COMMON-блок с динамически изменяемым размером, определяемым количеством записанных в него данных.

4. Разработан метод совмещения локальных переменных в фортранных программах, основанный на автоматическом построении графа программы. Предложены способы его реализации применительно к разным версиям ФОРТРАНа. Созданы средства совмещения локальных переменных для фортранных программ, работающих в системе COSDES, что дало в некоторых прикладных программах экономию памяти, необходимой для размещения локальных переменных, в размере около 50% (или до 0,7К ячеек). Общая экономия пропорциональна числу обслуживаемых программой абонентов.

5. Разработаны и реализованы средства отладки прикладных программ в диалоговых комплексах на ЭВМ ICL-1903A и DEC-10: специализированные языковые средства, средства распечатки данных процессов, средства автономной отладки прикладных программ (имитаторы проекторов, диспетчера системы разделения времени).

Л И Т Е Р А Т У Р А

1. В.Н.Бойцов, А.Л.Бонч-Осмоловский, Н.Г.Демидов, Т.С.Дурсенева, Ю.Н.Иванов, Е.Г.Сухов, А.Ю.Сычев, Е.Д.Щербаков. Математическое обеспечение просмотрово-измерительных проекторов, работающих на линии с ЭВМ ICL-1903A. Материалы семинара по обработке физической информации. Агверан, 1975. Ереван, АРУС, 1976, стр. 355.
2. Ю.Н.Иванов, А.Ю.Сычев, Е.Д.Щербаков. Трансляция модулей в системе FAUST. Препринт ИФВЭ 78-150, Серпухов, 1978.
3. В.Н.Бойцов, А.Л.Бонч-Осмоловский, Ю.Н.Иванов, А.Ю.Сычев, Е.Д.Щербаков. Программирование задач реального времени в системе FAUST. Препринт ИФВЭ 78-149, Серпухов, 1978.
4. Ю.Н.Иванов. Об одном методе организации работы последовательности программ на ЭВМ ICL-1903A. Препринт ИФВЭ 78-120, Серпухов, 1978.
5. Ю.Н.Иванов, Н.С.Иванова, А.Ф.Лукьянцев, Г.М.Максимов, Е.Д.Щербаков. Чистые процедуры на ФОРТРАНе в мультипрограммной системе реального времени. Препринт ИФВЭ 79-25, Серпухов, 1979.
6. Ю.Н.Иванов. О совмещении локальных переменных в программах на ФОРТРАНе. Препринт ИФВЭ 79-109, Серпухов, 1979.
7. Ю.Н.Иванов, Н.С.Иванова, В.Н.Ласковой, А.Ф.Лукьянцев, Г.М.Максимов, Р.А.Максимова. Организация мультиобслуживания под контролем операционной системы большой ЭВМ (DEC-10). Препринт ИФВЭ 81-27, Серпухов, 1981.
8. Ю.Н.Иванов, Н.С.Иванова, А.Ф.Лукьянцев, Г.М.Максимов. Средства разработки программ сопровождения в системе COSDES. Препринт ИФВЭ 81-148, Серпухов, 1981.

Рукопись поступила в издательскую группу
3 мая 1982 года.