


КИЕВСКИЙ ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ



На правах рукописи

Саад Кадум Маджид
Ирак

УДК 681.3.06

АВТОМАТИЗИРОВАННАЯ СИСТЕМА ИНКРЕМЕНТНОЙ РАЗРАБОТКИ
КРУПНЫХ ПРОГРАММ ДЛЯ МИНИ- И МИКРО-ЭЕМ

Специальность 05.13.11 - Математическое и програм-
мное обеспечение вычислительных машин, комплексов,
систем и сетей

А В Т О Р Е Ф Е Р А Т
диссертации на соискание ученой степени
кандидата технических наук

КИЕВ-1993

№ 27095

ЛННБ України ім.В.Стефаніка



00814236 (0)

робота виконана на кафедрі прикладної математики
Київського політехнічного інститута

Научний керівник: кандидат технічних наук, доцент
Россошнінський Д.А.

Офіційні опоненти: член-корреспондент АН УРСР,
доктор фіз. мат. наук, професор
Редько В.Н.

кандидат технічних наук,
Сергєєв А.І.

Ведущая организация: Інститут кібернетики ім.В.М.ГЛУШКОВА
АН УРСР

Защита состоится "17 05" 1993 г. в 14³⁰ час.
на засіданні спеціалізованого Ради Д 068.14.09 в Київському
політехнічному інституті, 252056, Київ-56, проспект Перемоги, 37.

С дисертацією можна ознайомитися в бібліотеці Київського
політехнічного інститута.

Автореферат розісланий "16.04." 1993 г.

Учений секретар
спеціалізованого Ради, *Бузовий*
доктор технічних наук, професор О.В.Бузовський

ЛННБ ім. В. Стефаніка
АН України

А Н Н О Т А Ц И Я

Целью диссертационной работы является исследование и реализация системы инкрементной разработки крупных программ для мини- и микро-ЭВМ (СИРКП). Система обеспечивает повышение эффективности труда программиста за счет уменьшения непроизводительных затрат времени на этапах компиляции и редактирования.

В соответствии с поставленной целью решаются следующие задачи:

- исследование существующих методов и технологии разработки больших программ;

- разработка метода разбиения больших программных систем на сегменты;

- анализ существующих СУБД для мини- и микро-ЭВМ и выбор подходящей системы управления с целью создания и сопровождения БД, обеспечивающих хранение текстов программ, объектных и загрузочных модулей, системы перекрестных связей;

- анализ характеристик наиболее известных диалоговых редакторов для мини- и микро-ЭВМ и разработка диалогового редактора, обеспечивающего текстовое редактирование и управление функциями по сборке модулей из сегментов или их разборке;

- исследование методов хранения и доступа к данным и обоснование выбора оптимального метода в рамках реализации системы;

- разработка функциональной структуры СИРКП, исследование принципов построения компонент системы.

Автор выносит на защиту следующие основные положения:

1. Функциональная структура СИРКП.
2. Методика определения оптимального размера сегмента.
3. Алгоритмы работы программы разбиения текста программы на сегменты.
4. Модель данных в СИРКП.
5. Функциональная структура диалогового редактора.
6. Расчет оптимальных по объему памяти физических блоков для последовательных и хеш-структур, используемых в предлагаемой СУБД.
7. Методика диагностических ошибок, обнаруженных компилятором на языке ассемблера и способ автоматического исправления некоторых их видов.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы. Появление высокопроизводительных и сложных вычислительных систем сделало человека и, в первую очередь, специалиста, не являющегося профессиональным программистом, самым узким звеном автоматизированной технологии решения многих практических задач. Современная тенденция развития вычислительной техники и ее математического обеспечения связана с передачей человеком машине все более "интеллектуальных" функций, что еще острее ставит вопрос о производительности труда человека, работающего во взаимодействии с ЭВМ. Значительные и еще не до конца использованные резервы повышения производительности лежат на пути облегчения отладки программ, эффективной организации диалогового взаимодействия, автоматизации процессов хранения и сопровождения документации на программные продукты. Роль перечисленных элементов усиливается с ростом объема и сложности решаемых задач, когда требуется отладка и стыковка с большим числом операторов.

Повсеместное использование персональных ЭВМ породило новые подходы к созданию математического обеспечения. Одним из них является ориентация на разработку интегрированных систем программирования, построенных на базе компонентов, имеющих возможность автономного функционирования. Анализ сложившейся ситуации в области создания системного математического обеспечения для мини- и микро-ЭВМ показывает, что эффективность труда программиста может быть значительно увеличена за счет разработки интегрированных систем и, в частности, систем инкрементной компиляции и редактирования, которые к настоящему моменту не созданы для мини- и микро-ЭВМ. Настоящая работа посвящена решению этой важной научной и практической задачи. Конечной целью работы является создание системы, которая будет обеспечивать разбиение крупных программ на слабосвязанные сегменты. Полученное представление может быть преобразовано в автономные независимо компилируемые модули, что позволяет впоследствии при внесении изменений в исходный текст программы выполнять перекompиляцию только одного или нескольких сегментов из набора. Для хранения сегментов и системы перекрестных связей необходимо создание соответствующей базы данных и средств автоматизации для работы с ней. Соответст-

вующие реализующие компоненты СИРКП, следовательно, должны включать СУБД, развитые средства редактирования, автоматизации обработки текстовой информации и данных соответствующих объектным модулям. Для обеспечения взаимосвязи между пользователем и системой предлагается создать диалоговый редактор, содержащий базовые функции для обслуживающей системы управления базами данных. Перечисленные вопросы составляют основное содержание работы.

Методы исследования базировались на использовании, технологии системного и структурного анализа, модульного программирования, теории графов, теории множеств и аппарате реляционной алгебры.

Научная новизна работы состоит в следующем:

- разработан алгоритм автоматического разделения исходного текста программы на слабосвязанные сегменты, которые могут быть преобразованы в автономные независимо компилируемые модули;
- разработан макрогенератор универсального типа, базирующийся на построении системы шаблонов с сопоставлением каждому шаблону действий, связанных с обработкой текстов программ, предназначенной для реализации алгоритма в системе;
- предложена модель специализированной базы данных реляционного типа, обеспечивающей автоматизацию процесса разработки программ на всех этапах, включая компиляцию, редактирование и отладку;
- разработаны методики для определения оптимального размера сегментов для структур хранения текстов в БД;
- разработана структура диалогового редактора, включающая операторы двух уровней.

Практическая ценность работы:

- разработана функциональная структура СИРКП, состоящая из четырех подсистем;
- в результате исследования большого числа программ определен оптимальный размер сегментов программ с целью оптимизации временных затрат, связанных с перекомпиляцией модулей в результате обнаружения ошибок для класса мини- и микро-ЭВМ;
- разработана физическая структура, обеспечивающая оптимальное хранение и доступ к данным;
- на основе анализа диагностических ошибок, обнаруженных компилятором на языке макроассемблера для класса мини-ЭВМ пред-

ложены алгоритмы для автоматического исправления некоторых видов ошибок.

Реализация результатов работы. Компоненты, разработанной СИРКП могут быть использованы автономно или комплексно. Так, автономно использованы макрогенератор и текстовый редактор. Например, один из макрогенераторов был использован для реализации кросс-ассемблера для микро-ЭВМ "Электроника С5". Редактор текстовой информации используется в системе программирования комплексов автоматизации сварочных процессов, разработанной на базе ОС РАТОС. Экспериментально проверена работоспособность алгоритма разделения программ на сегменты.

Апробация и публикации. Основные научные результаты, выполненные в диссертационной работе, докладывались и обсуждались на научно-технических семинарах кафедры прикладной математики в 1990-1991 г.г. и на научном семинаре в институте кибернетики им. В.М. Глушкова в 1992 г. По материалам диссертационной работы опубликованы две научные статьи.

Структура и объем диссертации. Диссертационная работа состоит из введения, четырех глав, заключения, списка литературы из 156 наименований и приложения. Основная часть работы изложена на 188 страницах машинописного текста, содержит 35 рисунков и 14 таблиц.

Во введении обоснована актуальность темы диссертационной работы, сформулированы задачи исследований и основные положения, выносимые автором на защиту.

В первой главе проведен анализ прогрессивных технологий разработки математического обеспечения и средств, которые могут быть использованы для построения технологических линий на основе инкрементной компиляции и редактирования.

Во второй главе предложена структура для рассматриваемых систем; метод и средства разделения программ на слабо связанные сегменты; исследованы модели данных в СИРКП; модель управления и фазы редактирования, компиляции и компоновки.

Третья глава посвящена исследованию методов хранения и доступа к данным и обоснованию выбора методов проектирования программ.

Четвертая глава содержит описание реализации основных компонент системы таких как - универсальный макрогенератор, редак-

тор текстовой информации и манипулирования данными; описание физической структуры файлов.

В заключении сформулированы основные результаты работы.

В приложении приведены: результаты изучения программ с целью определения размеров сегментов; результаты ошибок в программах на языке ассемблера; результаты анализа синтаксических ошибок и способы исправления некоторых из них.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Эффективность использования вычислительной техники во всех областях включает создание новых и совершенствование старых методов управления, упрощение и повышение надежности документооборота процесса программирования и, в первую очередь, определяется решением проблемы быстрой разработки качественных программ. Для решения этих проблем создаются промышленные технологии с оптимальными способами проведения процесса программирования, которые обеспечивают необходимые средства для спецификации проектирования и написания математического обеспечения ЭВМ. Анализ основных методов и средств их реализации, таких как структурный анализ, модульное программирование, композиционное программирование и др., а также технологических комплексов программирования (Р-технология, ПРИЗ, ТКП, ПРОМЕТЕЙ и др.) позволяет определить их общую направленность: расчленение процесса обработки информации на стандартные элементы и представление программы в виде иерархической многоуровневой структуры, выявление взаимосвязей между элементами, анализ потоков данных и создание диаграмм потоков данных для различных уровней детализации, определение технических характеристик программного обеспечения на разных этапах разработки программ, их тестирования и отладки. Анализ в области средств поддержки требований и спецификаций в разработке программ показывает, что управление базами данных более унифицировано. Функции управления более ограничены, хотя сами базы данных имеют больший объем за счет содержания информации в символьной и графической форме. Анализ состояния дел в области организации процессов компиляции и редактирования показывает, что существующие подходы не обеспечивают повышение эффективности труда программистов в полной мере, хотя и повышают эффективность использования средств ВТ. Каждый раз при обнаружении ошибки требуется перекомпиляция всего тела программы, либо формирова-

ние необходимого контекста для компиляции некоторой единицы, где обнаружена ошибка. Целью настоящей работы является исследование и разработка средств инкрементной компиляции и редактирования. Функциональная структура СИРКП содержит четыре относительно автономных подсистемы (рис.1): проверки и выделения сегментов (ПРВД); базы данных (БД); управления базой данных (УБД); редактирования, компиляции и компоновки (РДК). В первой подсистеме решается задача разбиения крупных программ на сегменты с целью уменьшения непроизводительных потерь времени, возникающих при обмене данными между внешними запоминающими устройствами (ВЗУ) и оперативной памятью (ОП) в процессе разработки программы; обеспечения автономного доступа к каждому сегменту; обеспечения автономной модификации модулей, при которой изменение модуля можно отобразить на составляющий его сегмент и несколько связанных с ним. Содержательная постановка задачи формируется следующим образом. Процесс разработки программы на ЭВМ организован так, что в ОП размещается не вся программа, а лишь некоторая ее активная часть, которая размещается в фиксированном сегменте заданного объема. Остальная часть программы размещается на ВЗУ также в виде сегментов. Процедура разбиения крупной программы на сегменты решается в два этапа. На первом этапе определяется минимальный объем программы для того, чтобы было выгодно разделить на сегменты. На втором шаге определяется оптимальный размер сегмента. На первом шаге используется пара параметров:

$$\langle T_k, T_i \rangle, \quad (1)$$

где $T_k = \{T_{k1}, T_{k2}, \dots, T_{kn}\}$ - конечное не пустое множество времени

для разных программ в обычном методе;

$T_i = \{T_{i1}, T_{i2}, \dots, T_{in}\}$ - конечное не пустое множество времени

компиляции в методе инкрементной разработки программы.

На рис.2 показан график зависимости времени компиляции двух указанных выше параметров от количества операторов в программе. Исследования для 40 программ имеют разные количества операторов, показывающих, что время компиляции для двух параметров увеличи-

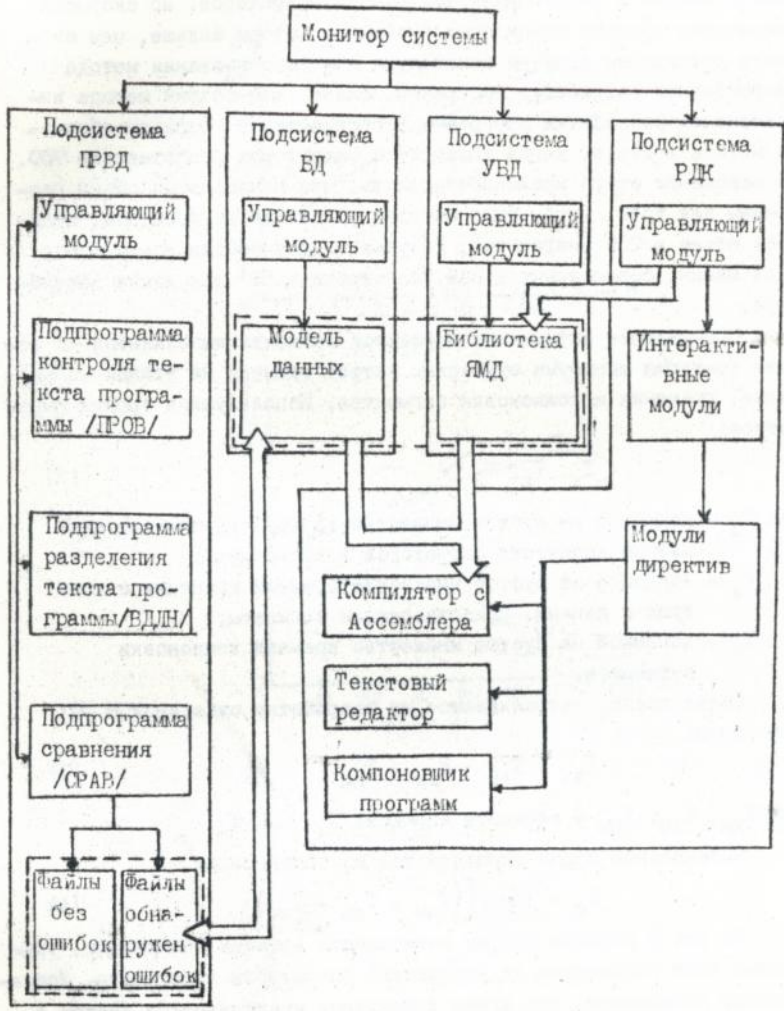


Рис.1 Функциональная структура СИККП

ваются вместе с увеличением количества операторов, но скорость увеличения времени компиляции в обычном методе больше, чем скорость увеличения времени компиляции при использовании метода инкрементной разработки программы. Видно, что график метода инкрементной разработки программы пересекается с графиком обычного метода в точке, когда количество операторов представляет 700. На основании этого можно опеределить, что минимальный объем программы для того, чтобы было выгодно разделить на сегменты, является объем в 700 операторов. Результат получен для конкретного ряда машин, совместимых в ЭВМ "Электроника 60" для языка ассемблера.

Определение оптимального размера сегмента выполняется на основе критерия минимума суммарных затрат времени на этапах компиляции, хранения и компоновки сегментов. Используемая тройка параметров:

$$\langle T_K, T_X, T_H \rangle \quad (2)$$

где T_K - конечное не пустое множество времени компиляции для разных количеств операторов в программах;

T_X - конечное не пустое множество времени хранения и доступа к данным, представляющим сегменты;

T_H - конечное не пустое множество времени компоновки сегментов.

Общее время, затрачиваемое на разработку отдельной M -й программы:

$$T_{CM} = T_{KM} + T_{XM} + T_{HM} \quad (3)$$

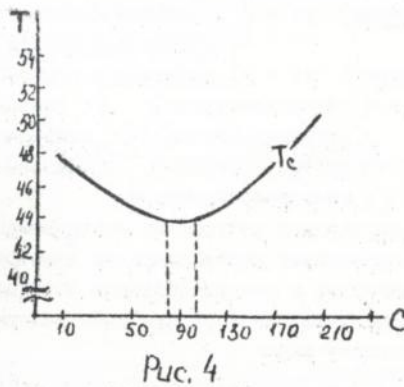
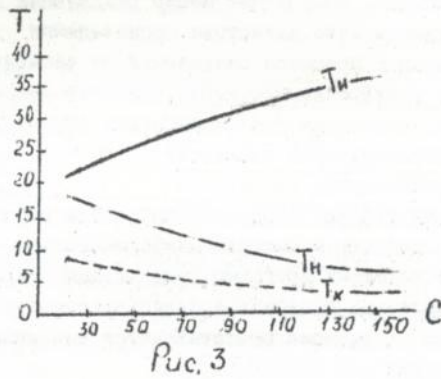
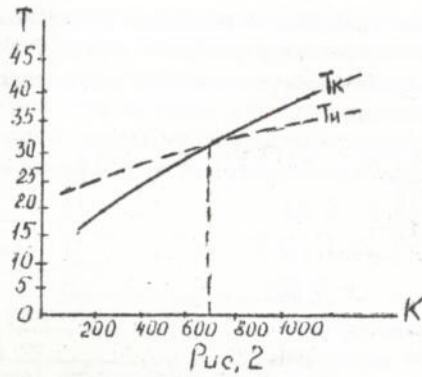
где T_{KM} , T_{XM} , T_{HM} - элементы множеств T_K , T_X , T_H .

Оптимальное время определяется из соотношения

$$T_C = \text{Min} \{ \{ T_{KM} + T_{XM} + T_{HM} \} \} \quad (4)$$

На рис.3 показан график зависимости времени реализации, указанных выше параметров от количества операторов в сегменте. Исследование показывает, что время компиляции увеличивается вместе с увеличением количества операторов в сегменте, а времена хранения и компоновки уменьшаются. Оптимальный размер сегмента составляет 85-95 операторов (рис.4) для указанных ранее классов ЭВМ и программ.

Алгоритм разбиения крупной программы на сегменты сводится



к решению комбинаторной задачи разбиения управляющего графа программы. Суть задачи состоит в разбиении графа $T(X, U)$ на конечное число непересекающихся множеств, где каждое X_i имеет вес C_i так, что

$$X = \bigcup_{i=1}^k X_i, \quad X_i \cap X_j = \emptyset, \quad i \neq j$$

$$\sum_{i=1}^k C_i \leq M$$

$$95 \geq C_i \geq 85$$

$$M \geq 700$$

где C_i - размер сегмента;
 M - объем программы;
 X_i - сегмент, полученный после разделения программы;
 U - подмножество декартова произведения

Для реализации процесса разделения на сегменты разработан макрогенератор универсального типа, ориентированный на шаблоны линейного вида. Основными информационными структурами, обрабатываемыми макрогенератором являются:

- файл с шаблонами,
- командная строка, определяющая режимы работы,
- рабочие таблицы и массивы макрогенератора,
- файлы с текстами программ, подлежащими обработке.

Файл с шаблонами содержит последовательность пар шаблон, номер-процедуры, которые представляются следующими синтаксическими правилами:

$$\langle \text{файл с шаблонами} \rangle ::= \left\{ \begin{array}{l} \langle \text{шаблон} \rangle \langle \text{номер-процедуры} \rangle \\ \langle \text{строка-пробелов} \rangle \end{array} \right\}$$

$$\langle \text{номер-процедуры} \rangle ::= \langle \text{десятичное-целое-без-знака} \rangle$$

$$\langle \text{шаблон} \rangle ::= \left\{ \begin{array}{l} \langle \text{литерная-строка} \rangle \left\{ \begin{array}{l} \langle \text{символ-аргумента} \rangle \\ \langle \text{литерная-строка} \rangle \left[\langle \text{символ-аргумента} \rangle \right] \\ \langle \text{литерная-строка} \rangle \left[\langle \text{символ-аргумента} \rangle \right] \\ \left[\langle \text{символ-аргумента} \rangle \right] \end{array} \right. \end{array} \right\}$$

где $\left\{ \dots \right\}$ - определяет рекурсивно повторяемую конструкцию,
 $\left[\dots \right]$ - определяет необязательную конструкцию.

Действия, связанные с каждым шаблоном определяются либо процедурой, программируемой на фортране либо текстами, содержащими последовательность вида:

$\langle \text{метка} \rangle \langle \text{предикат} \rangle / \langle \text{последовательность-операций} \rangle \langle \text{метка-перехода} \rangle$

Результаты выполнения действий и обработанные части текстов программы используются СИРКП для размещения в БД информации о командах передачи управления и метках в тексте программы. Полученная информация используется как справочник для модулей при реализации алгоритма разбиения текста программы на сегменты. Алгоритм работы универсального макрогенератора показан на рис.5.

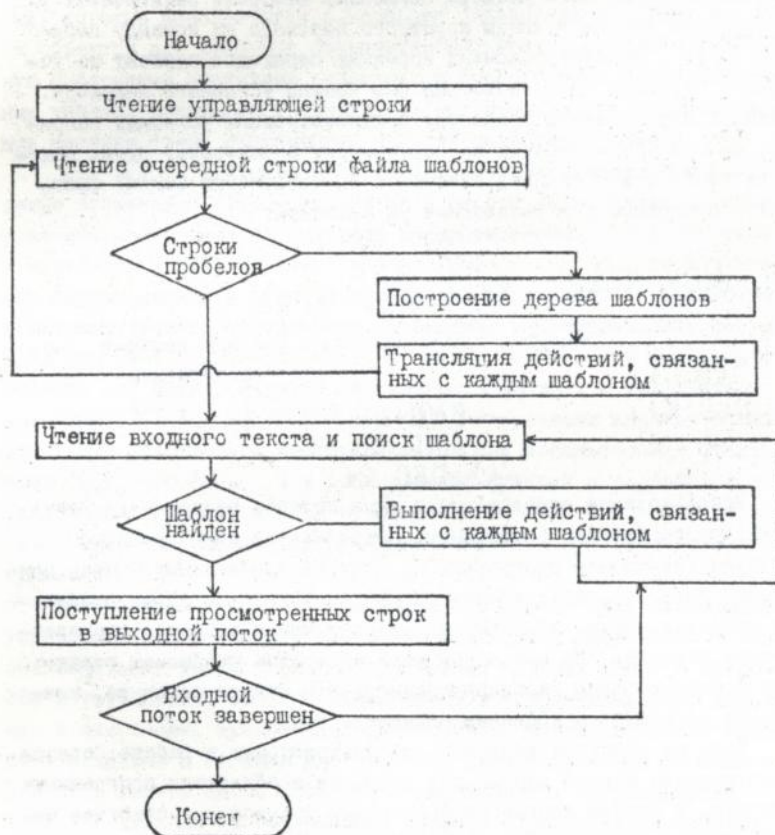
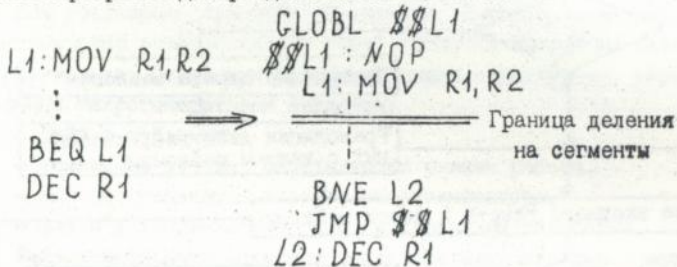


Рис.5.

На основе информационных структур, обрабатываемых макрогенератором, можно построить таблицу связи, которая используется в

процессе разбиения текста программы на сегменты. Алгоритм разбиения включает в себя анализ адресов перехода команд коротких переходов, определение типа перехода вперед или назад и формирование разбиения текста программы на сегменты на основании определенного размера сегмента $95 \geq C \geq 85$. В алгоритме предложен способ исключения полностью возможности передачи управления за пределы определенного размера сегмента. Контроль заключается в определении замены команды короткого перехода на команду перехода JMP. Множество команд коротких переходов состоит из условного перехода BR и восьми пар команд условного перехода. В каждой паре команды реализуют противоположные условия, например, BNE и BEQ - переход по неравенству и равенству нулю. Исходя из этого можно сделать следующую эквивалентную замену фрагментов программы для разделения на сегменты:



Использование предложенного выше способа позволит улучшить процедуру разбиения на сегменты. Недостатками данного способа являются увеличение программы, за счет дополнительных команд при наличии большого числа команд перехода, появление новых глобальных объектов метки, что приводит к увеличению размеров словарей внешних объектов. На последнем шаге алгоритма разбиения реализуется формирование глобальных символов в каждом сегменте, именования сегментов и хранения сегментов в БД.

Один из основных вопросов, рассматриваемых в работе, связан с построением модели данных для хранения и обработки программных структур. В основе модели иерархические взаимосвязи объектов четырех типов: ПОЛЬЗОВАТЕЛЬ, ПРОГРАММА, МОДУЛЬ, СТРОКА. При отбражении на соответствующие реализующие подсистемы на самом верхнем уровне находится библиотека пользователей, содержащая информацию о всех пользователях системы. На нижних уровнях рассматриваются программы пользователей, соответствующие модули которых

получаются после разделения программ, и строки в каждом модуле. Каждый объект имеет локальную информационную структуру, включая их сущность и связь, которые удовлетворяют всем требованиям к данным соответствующего приложения. Локальные информационные структуры всех объектов связаны между собой с помощью идентификаторов объектов, имеющих специальную структуру. Информация, входящая в состав структуры, реализуется с помощью словарей данных, которые используются в процессе управления данными, как база объектов для конечных пользователей. Модель содержит глобальную информацию структуру, включающую совокупность вспомогательных таблиц, связанных с операциями над отношениями и определяющих правила связи программных модулей по информации.

Для выбора подходящей системы управления базой данных с целью создания и сопровождения БД в работе предприятия попытка сравнительного анализа наиболее распространенных СУБД для мини- и микро-ЭВМ и ПЭВМ отмечено значительное место средств управления базами данных в области технологии программного обеспечения и математического обеспечения. В анализе рассматривались пять СУБД для СМ-4 (СЕТОР-СМ, КВАНТ-М, ВАРС, ФОРАН, INGRES), две системы для Электроники-60 (РДБ-микро, RDBMS) и три системы для ПЭВМ (MDBS, DB-VISTA, FOXPRO). Результаты анализа показывают, что большинство систем строится без использования стандартных СУБД, что приводит к дублированию функций и разработок в различных системах; в СУБД современных мини- и микро-ЭВМ отсутствуют средства реализации принципов инкрементной разработки программ.

На основании результатов анализа в работе предложена модель управления базой данных, представляющая собой расширение функциональных возможностей систем управления данными мини- и микро-ЭВМ и устраняющая основные недостатки в современных СУБД для мини- и микро-ЭВМ. Функциональная структура модели включает средства создания и ведения БД, средства манипулирования данными и средства защиты БД. В СУБД СИРКП используется непроцедурный язык, основанный на реляционной алгебре и предназначенный для обеспечения связи пользователя и администратора БД с системой управления данными. Команды языка разбиваются на четыре группы, показанные в таблице I.

В состав функции защиты БД входят три класса средств, обес-

Команды манипулирования данными в СИРКП

Табл. I

Команда	Назначение
	<u>Команды создания и ведения БД</u>
СБ	Создание схемы БД
УБ	Уничтожение схемы БД
СО	Создание схемы отношения ЛИС
УО	Уничтожение схемы отношения ЛИС
ОТ	Объединение отношения
РТ	Разделение отношения
СГ	Создание схемы отношения ГИС
УГ	Уничтожение схемы отношения ГИС
	<u>Команды модификации данных</u> <u>внутри отношения</u>
IK	Добавление нового кортежа
KK	Уничтожение кортежа
СК	Изменение содержания кортежа
OK	Объединение двух кортежа
PK	Разделение одного кортежа
	<u>Команды выборки и просмотра данных</u>
ВЫБ	Выборки данных в отношении
СМР	Просмотр данных текущего кортежа
ЛСТ	Просмотр всех выбранных строк отношения
	<u>Сервисные команды</u>
ДИР	Распечатка каталога отношений
КО	Копирование отношения
ВСТ	Вставка
СБ	Уплотнение данных отношения

печивающих целостность, которые свяжутся к защите от параллельных модификаций с помощью механизма полной блокировки всей базы данных и запрету доступа любому процессу во время модификационного процесса; защите от ошибок, обнаруживаемых трансляторов в программе; защите от разрушения основного носителя.

В подсистеме РДК содержится текстовый редактор, реализуемый по принципу непосредственного редактирования и использующий макрогенератор специального назначения, включающий операторы двух уровней: пользовательского и базового. Пользовательские операторы предназначены для выполнения текстового редактирования, а базовые - для обслуживания СУБД реляционного типа и выполнения управляющих функций по сборке модулей из сегментов или их разборке.

В работе проведен анализ диагностических ошибок, обнаруживаемых компилятором на языке макроассемблера и предложены алгоритмы автоматического исправления некоторых видов ошибок.

Еще один раздел связан с вопросом поиска оптимального варианта хранения и доступа к данным на физическом уровне. Процесс исследования включает в себя пять методов, обеспечивающих поиск и манипулирование сегментами и записями каждого сегмента, хранямыми в файлах БД. В состав этих методов входят последовательное распределение памяти, упорядоченное и связанное, хеширование идентификатора, хеш-последовательная организация, B*-дерево поиска и произвольная обработка. В процессе исследования были проанализированы способы использования каждого метода в СИКР, его преимущества и недостатки. На основании этих исследований был выбран метод связанного распределения памяти для хранения записей о пользователях и программах в соответствии со следующими критериями: корректировка файлов БД не требует существенных затрат времени и памяти ЭВМ; использование указателя обеспечивает достаточно высокую скорость доступа и эффективность поисковых операций.

Для хранения записей о модулях и строках был выбран хеш-последовательный метод, преимущества которого состоят в эффективной выборке и обновлении отдельных записей, корректировка файлов БД не вызывает существенных трудностей, не требуется хранение дополнительной адресной информации.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ РАБОТЫ

1. Разработана функциональная структура СИРКП, состоящая из четырех подсистем. В основе построения лежат методы структурного анализа и модульного программирования.

2. Разработан алгоритм и средства реализации для автоматического разделения исходного текста программы на слабосвязанные сегменты. Сегментное представление не влияет на исходный текст программы и используется только на этапах компиляции и связывания и для представления программы в базе данных СИРКП.

3. Предложена модель управления базой данных, обеспечивающей автоматизацию процесса разработки программ на всех этапах, включая компиляцию, редактирование и отладку на основе сегментного представления модулей с возможностью автоматической выборки зависимых сегментов, передаваемых на перекомпиляцию в случае внесения изменений в исходные тексты программ.

4. Разработаны принципы построения и реализован редактор для СИРКП, выполняющий обработку команд двух типов: для передачи СУБД реляционного типа и выполнения редактирования пользовательских программ.

5. Разработана методика использования диагностики ошибок, обнаруженных компилятором и способ автоматического исправления некоторых их видов.

Результаты диссертации опубликованы в следующих работах:

1. Россошинский Да7, Саад К.М. Метод реализации инкрементной компиляции и редактирования для мини- и микро-ЭВМ. - К.: Киев. политехн. ин-т, 1988, 6 с. Рук. деп. в УкрНИИТИ 8.04.88, № 882-Ук88.

2. Саад К.М. Система инкрементной компиляции и редактирования для мини- и микро-ЭВМ. - К.: Киев. политехн. ин-т, 1988, 6с. Рук. деп. в УкрНИИТИ 8.04.88, № 881-Ук88.

Подписано в печать 14.04.93г формат 60x84/16
Бумага писчая. Усл.печ.л. 1,0. Тираж 100 экз. Заказ № 676
Отпечатано ЦУОП ГНПП "Плодвинконсерв" г.Киев,Сакаганского,1

AB 27.095