



ВСЕСОЮЗНЫЙ  
НАУЧНО-ИССЛЕДОВАТЕЛЬСКИЙ ИНСТИТУТ  
ИНФОРМАЦИИ И ЭКОНОМИКИ МИНПРИБОРА СССР  
(ИНФОРМПРИБОР)

КАТАЛОГ

# ТЕХНИЧЕСКИЕ СРЕДСТВА АСУТП

Средства вычислительной техники

СИСТЕМНЫЕ ПРОГРАММНЫЕ СРЕДСТВА  
СМ ЭВМ

# ПСИ

Москва 1989



И. М. Аберемков, О. Г. Галата,  
А. Н. Жуков, Е. А. Исаичкина,  
Н. Н. Киселев, О. М. Мешков,  
А. Х. Морено-Пальи, Ф. М. Мучник,  
В. С. Нахмансон, Н. М. Новичихин,  
Н. П. Селезнева, Э. И. Станкевич,  
Н. И. Федченко

ИНФОРМПРИБОР выпускает каталог «Комплексы технических средств» (каталог ГСП) под общей редакцией канд. техн. наук В. А. Рухадзе.

Каталог издается в виде отдельных выпусков, содержащих описание технических средств, объединенных по отдельным измеряемым физическим величинам или выполняемым функциям в составе АСУТП.

В каталог включены сведения по системным программным средствам СМ ЭВМ Межотраслевого фонда системных программных средств воронежского СКТБ «Системпрограмм», московского НПО «Электронмаш», разработанным в 1987—1988 гг., поставка и внедрение которых осуществляются по прямым договорам непосредственно специалистами-разработчиками.

Приведенные в каталоге сведения содержат краткое описание программных компонент языков и систем программирования для 32-разрядной ЭВМ СМ 1700, в том числе языков «новой волны» — Пролог и Модула-2, отечественных СУБД реляционного типа БАРС для СМ-4 и СМ 1420, ИНТЕРЕА.Т для СМ 1420, СМ 1425, СМ 1810 и СМ 1820, БАРД для СМ-2М и СМ 1210, а также мобильной ПИХ-подобной операционной системы ДЕМОС и диалоговой системы разделения времени ДИАМС. Особый интерес для разработчиков прикладных программ представляют инструментальные программные средства для СМ 1810 ИНТЕРПРО и ТУРБО ПАСКАЛЬ. Предлагаемые системные программные средства не требуют особой конфигурации технических средств СМ ЭВМ.

Изложенные в каталоге сведения предназначены для проектировщиков АСУ различного назначения в качестве рекламного материала для последующего заказа программных изделий. За справками по приобретению просим обращаться по адресу: 394006, Воронеж, ул. Моисеева, 5, Воронежское СКТБ «Системпрограмм», тел. 57-52-99, телетайп 153476 «Олень».

По всем вопросам, касающимся издания каталога, просим обращаться по адресу: 125877, ГСП, А-252, Москва, Чапаевский пер., 14, ИНФОРМПРИБОР.

*Ответственный за выпуск И. Н. Морозова*



ИНФОРМПРИБОР

Технические средства АСУТП  
Отраслевой каталог

Средства вычислительной техники  
**СИСТЕМНЫЕ ПРОГРАММНЫЕ СРЕДСТВА  
СМ ЭВМ**

ГОСУДАРСТВЕННАЯ СИСТЕМА ПРОМЫШЛЕННЫХ ПРИБОРОВ  
И СРЕДСТВ АВТОМАТИЗАЦИИ

Москва 1989

## ВВЕДЕНИЕ

Системное программное обеспечение является необходимой составной частью ЭВМ, в состав которого входят операционные системы (ОС), языки и системы программирования, инструментальные и сервисные программы. Системные программные средства, дополняя аппаратные средства любого компьютера, являются инструментарием прикладных специалистов, с помощью которого осуществляются разработка прикладных программ и организация вычислительного процесса.

Система малых ЭВМ (СМ ЭВМ) имеет широкую номенклатуру системных программных средств, часть которых является базовым программным обеспечением (ПО), поставляемым непосредственно заводом-изготовителем вместе с вычислительным комплексом (ВК). Как правило, номенклатура базового ПО ограничивается одной из ОС и набором тестово-диагностических программ или систем типа ТМОС (тест-мониторная операционная система), ТЕДОС (тестово-диагностическая операционная система) и т. д.

Однако практическое применение ЭВМ конкретным пользователем часто вызывает необходи-

мость использования системных программных средств, отличных от тех, что поступили с ЭВМ. В этом случае пользователь вынужден обращаться к разработчикам программной продукции и приобретать ее дополнительно. Настоящий каталог и предназначен для того, чтобы дать представление пользователю о имеющихся дополнительных системных программных средствах для определения возможности их приобретения и использования.

В каталоге описаны ОС, не поставляемые с ЭВМ. Прежде всего, это диалоговая единая мобильная ОС (ДЕМОС) для ЭВМ СМ 1700, СМ 1702 и СМ 1820. ДЕМОС является UNIX, подобной универсальной ОС, предназначенной для решения широкого круга научно-технических, экономических и управленческих задач. Система реализуется на языке структурного программирования СИ, включающего большинство возможностей языков высокого уровня и одновременно обеспечивающего доступ к архитектурным элементам конкретной ЭВМ. В состав ОС ДЕМОС входят компоненты, обеспечивающие интерфейс оператора, программиста, подготовку и отладку новых программ, управ-



ление файловой системой и вводом-выводом. ДЕМОС обеспечивает многопользовательский, мультипрограммный режим работы, независимость программ от внешних устройств и способов доступа к информации.

Разрабатываемая специалистами воронежского СКТБ «Системпрограмм» московского НПО «Электронмаш» ОС охватывает всю номенклатуру СМ ЭВМ, что особенно эффективно с точки зрения пользователя и особенно при условии наличия в ОС ДЕМОС такого важного свойства, как мобильность. Семейство ОС ДЕМОС имеет набор языков и систем программирования как традиционных Фортран, Паскаль, СИ, Ассемблер, так и языков «новой волны» Пролог и Модула-2. В каталоге дается описание указанных языков и систем программирования и особенностей их использования в многофункциональной ОС с виртуальной организацией памяти (МОС ВП) для ЭВМ СМ 1700.

Кроме ДЕМОС представляет интерес пользователям диалоговая многотерминальная система ДИАМС для малых ЭВМ типа СМ и «Электроника», предназначенная для создания и ведения данных большого объема для решения информационно-логических задач. ДИАМС эффективно используется для организации данных и решения статистических расчетов, ведения делопроизводственной документации и документации архивов, оперативной обработки постоянно меняющейся информации. В каталоге дано описание последней расширенной и дополненной рядом сервисных компонент версии ДИАМС 3.3.

ОС ДЕМОС и ДИАМС не поставляются в комплекте ни с одним ВК. Их приобретение возможно только через межотраслевой фонд системных программных средств СМ ЭВМ (МОФАП).

Инструментальные средства разработки программ представлены в каталоге двумя программными изделиями: системой программирования ТУРБО ПАСКАЛЬ для СМ 1810 и средствами автоматизации разработки ПО для СМ 1810. Эта

категория системных программных средств наряду с языками и системами программирования является совершенно необходимой принадлежностью рабочего места прикладного программиста, без которых невозможно говорить об эффективной организации труда этой категории специалистов.

Системы управления базами данных (СУБД) и интегрированные средства обработки данных также следует отнести к системному программированию. Операционная среда, набор языковых интерфейсов, организация управления вычислительным процессом — все это атрибуты, определяющие отношение СУБД к системному ПО. В каталоге представлены пять СУБД реляционного типа для СМ ЭВМ типа СМ-4, СМ 1420, СМ 1425, СМ 1810, СМ 1820, СМ-2М и СМ 1210. Все пять СУБД являются оригинальными отечественными разработками, по своим характеристикам не уступающими адаптированным зарубежным аналогам. Это и СУБД БАРС в ОС РАФОС для СМ-4, СМ 1420, СУБД ИНТЕРЕАЛ для СМ-4, СМ 1420, СМ 1425, СМ 1810 и СМ 1820.

Для ЭВМ типа СМ 1810 и СМ 1820 в каталоге также представлены интегрированные средства обработки данных ИНТЕРСОД и система управления реляционными БД ДАТАРЕАЛ, обеспечивающие обработку крупноформатных электронных таблиц и деловую графику, что, несомненно, найдет интерес у пользователя. Для систем реального времени, реализуемых на ЭВМ типа СМ-2М и СМ 1210, будет интересно ознакомиться с основными характеристиками и возможностями СУБД БАРД, функционирующей под управлением ДОС АСПО СМ ЭВМ.

Приведенные в каталоге данные основаны на программной и эксплуатационной документации на представленные программные изделия с использованием сформировавшейся в нашей стране профессиональной терминологии в области программирования и автоматизации управления с применением ЭВМ.

## ОПЕРАЦИОННЫЕ СИСТЕМЫ

### Диалоговая единая мобильная операционная система для 32-разрядных малых ЭВМ СМ 1700 (ДЕМОС 32.2)

ДЕМОС для 32-разрядных малых ЭВМ СМ 1700 (ДЕМОС 32.2) — это универсальная ОС для решения широкого круга научно-технических, экономических и управленческих задач. ДЕМОС 32.2 является дальнейшим развитием ОС ДЕМОС 32 и предназначена для широкого применения в различных системах управления и обработки данных.

ДЕМОС 32.2 обеспечивает мультипрограммный, многопользовательский режим, а также имеет

развитые инструментальные средства для разработки программ.

Данная версия ОС ДЕМОС удовлетворяет принципам единого интерфейса пользователя, реализация которого позволяет решить ряд принципиальных проблем ПО: обеспечить возможность создания мобильных пакетов программ для ЭВМ различных типов; предоставить средства для создания унифицированной среды пользователя для разнотипных ЭВМ; снизить затраты на переобучение пользователей и обслуживающего персонала при переходе с одной вычислительной установки на другую; использовать программные средства, разработанные в среде ДЕМОС 32.2, для других ЭВМ.

ДЕМОС 32.2 входит в семейство ОС, совместимых с ОС UNIX, интерфейс которой постепенно становится международным стандартом среды пользователя. Широкое распространение систем этого типа объясняется не только их простотой,



логичностью, преимуществами использования унифицированной среды, но и рядом удачных конструктивных решений, позволяющих им успешно конкурировать с оригинальными ОС для конкретных ЭВМ.

Функциональным назначением ДЕМОС 32.2 является обеспечение пользователя развитыми средствами подготовки и отладки программ на языках высокого уровня, а также средствами управления программами на стадии их выполнения.

ДЕМОС 32.2 позволяет использовать технические средства СМ ЭВМ и предназначены для работы на одно- и многомашинных комплексах, в системах коллективного пользования.

Эксплуатационное назначение ДЕМОС 32.2 — обеспечение работы ЭВМ семейства СМ 1700 в режиме разделения времени.

ДЕМОС 32.2 состоит из набора программных модулей, предоставляющих пользователю следующие возможности: управление ОС с помощью специального языка управления;

редактирование и форматирование текстовых файлов, подготовленных как средствами ДЕМОС 32.2, так и средствами ОС РВ, РАФОС и МОС ВП; сервисные средства разработки прикладных программ; создание, отладку и выполнение программ на языках программирования высокого уровня, необходимых в инженерных, научных, экономических и других применениях: СИ, Фортран, Паскаль; управление вводом-выводом; унифицированный механизм обработки регулярных выражений; контролируемый доступ к системе; организацию и управление файлами на внешних запоминающих устройствах; генерацию системы под конкретный состав технических средств.

Функциональные возможности ОС ДЕМОС 32.2 прежде всего определяются ее целевой направлен-

ностью на применение в коллективных системах обработки информации. В состав ОС ДЕМОС 32.2 включены более 200 команд. Среди них имеются такие средства разработки программ, как компиляторы, отладчики, редакторы текстов, системы подготовки документации и обучения, информационные и почтовые службы.

Концепция «фильтров», свойства командного языка и файловой системы позволяют организовать сложные системы обработки информации, используя уже имеющийся набор команд. Набор функций управления терминалом позволяет разрабатывать интерактивные компоненты, практически независимые от типа устройства.

В ОС ДЕМОС 32.2 включены развитые средства программирования и создания сложных программных комплексов. В нее входят компиляторы с языков СИ и Фортран, компиляторы LEX и УАСС, система MAKE; имеются развитые средства отладки и другие инструментальные средства.

ОС ДЕМОС 32.2, СИ-компилятор и, по существу, все прикладные программы системы написаны на языке СИ. СИ — это универсальный язык программирования, для которого характерны экономичность выражения, широкий набор операторов и типов данных. Преимуществом языка является то, что он не связан с какой-либо одной ОС или машиной и может использоваться для написания любых больших вычислительных программ, программ для обработки текстов, СУБД и т. д.

Выбор интерпретатора командного языка для взаимодействия с системой делается каждым пользователем индивидуально, и в качестве такого интерпретатора может применяться любая интерактивная программа. Подобный подход позволяет одновременно существовать нескольким командным языкам; в данной версии ОС ДЕМОС их два: SHELL и CSHELL. Оба командных языка выступают и как языки управления заданиями, обеспечивающие взаимодействие с ОС, и как достаточно мощные и гибкие языки программирования, позволяющие строить сложные командные файлы. Оба языка позволяют организовывать циклы и проверки различных условий. Синтаксис языка CSHELL во многом совпадает с синтаксисом языка СИ. В данной версии системы CSHELL не является основным командным языком, но его популярность растет, и к настоящему времени язык CSHELL стал стандартным компонентом для наиболее развитых систем семейства UNIX.

Состав и функции ДЕМОС 32.2 перечислены в табл. 1.

ОС ДЕМОС состоит из ядра, работающего в привилегированном режиме, и программных компонентов окружения, обеспечивающих среду пользователя.

ОС ДЕМОС размещается на внешних запоминающих устройствах прямого доступа. Текущая версия системы в качестве таких устройств может использовать магнитные диски типа СМ 5408. Ядро системы является резидентным и считывается из файловой системы в процессе начальной загрузки ОС в память ЭВМ.

Пространство на дисках делится на следующие части: корневая файловая система; область подкачки; монтируемые файловые системы.

Корневая файловая система содержит основные программы, необходимые для нормального функ-

Таблица 1

Подсистема	Состав функций
Ядро ОС	Управление устройствами Управление процессами
Управление ОС	Создание и контроль файловой системы Сохранение и восстановление файловой системы Генерация, реконфигурация, проверка и сохранение ОС
Пользовательский интерфейс	Информационные команды Управление файлами Обеспечение многопользовательского режима
Системы программирования	Реализация языков программирования Реализация командных языков (интерпретаторов)
Разработка программ	Отладка, сборка и проверка программ Автоматизация и контроль разработки и сопровождения программ
Обработка текстов	Редактирование текстов Форматирование текстов
Межмашинное взаимодействие	Передача файлов Обмен почтой



ционирования ОС. Устройство, на котором расположена корневая файловая система, задается при настройке ОС и может быть изменено при ее вызове.

Область подкачки служит для поддержки виртуальной памяти машины. Расположение области подкачки указывается при настройке системы и также может быть изменено при вызове системы.

Монтируемые файловые системы — это диски (или части дисков), на которых находятся файловые системы ОС или пользователей. Монтируемые файловые системы подключаются после вызова ОС либо по команде оператора, либо автоматически.

**Поддержка технических средств.** ДЕМОС 32.2 предназначена для управления ВК СМ 1700 и обеспечивает работу со следующими устройствами: один процессор СМ 2700; четыре ОЗУ СМ 1700.3522 (1 Мбайт);

два накопителя на СМД (СМ 5408); накопитель на магнитных дисках типа «Винчестер» емкостью 150 Мбайт СМ 5504 (СМ 5514); консольное устройство загрузки СМ 5218; накопитель на магнитных лентах СМ 5309; консольное устройство печати с клавиатурой СМ 6380 (СМ 6333); терминал типа ВТА 2000-15М (СМ 7238.01); устройство печати СМ 6361 (СМ 6334).

## **Диалоговая единая мобильная операционная система для 32-разрядных малых ЭВМ СМ 1702 (ДЕМОС 1702)**

ДЕМОС для 32-разрядной малой ЭВМ СМ 1702 (ДЕМОС 1702) — это универсальная ОС, предназначенная для широкого применения в различных системах управления и обработки данных и обеспечивающая мультипрограммный, многопользовательский режим, а также имеющая развитые инструментальные средства для разработки программ.

ДЕМОС 1702 разрабатывается в рамках семейства ОС ДЕМОС для различных ЭВМ с единым интерфейсом пользователя, и программно-совместимых с ОС UNIX.

Функциональным назначением ДЕМОС 1702 является обеспечение пользователя развитыми средствами подготовки и отладки программ на языках высокого уровня, а также средствами управления программами на стадии их выполнения.

ДЕМОС 1702 обеспечивает выполнение следующих основных функций: управление виртуальной памятью объемом 4096 Мбайт; управление оперативной памятью от 1 Мбайт и более, динамически разделяемой между пользователями; работу до 40 пользователей в режимах разделения времени и пакетном одновременно; обслуживание иерархической файловой системы; приоритетное управление выполнением задач; обслуживание операций ввода-вывода; запуск задач по инициативе оператора или других задач; автоматический рестарт системы после сбоя; управление машинными часами и таймером; взаимодействие задач пользователя с помощью механизмов именованных конвейеров и разделяемых областей памяти; управление файлами с

последовательным и простым методами доступа; обмен файлами между ЭВМ через асинхронные линии связи; генерацию системы под конкретный состав технических средств; подключение разработанных пользователем драйверов дополнительных внешних устройств; управление терминалами через компилируемую БД характеристик терминалов.

ДЕМОС 1702 состоит из набора программных модулей, представляющих пользователю следующие возможности: управление ОС с помощью специального языка управления; редактирование и форматирование текстовых файлов; сервисные средства разработки прикладных программ; создание, отладку и выполнение программ на языках программирования высокого уровня, необходимых в инженерных, научных, экономических и других применениях: СИ, Фортран, Паскаль; обслуживание библиотек исполняющих систем языков программирования высокого уровня; управление вводом-выводом; унифицированный механизм обработки регулярных выражений; контролируемый доступ к системе; организацию и управление файлами на внешних запоминающих устройствах; генерацию системы под требуемый состав технических средств; организацию межмашинного взаимодействия; организацию работы в многомашинных комплексах.

Дополнительные программные компоненты, расширяющие функциональные возможности ДЕМОС 1702 и работающие под ее управлением, включают в себя: средства построения и взаимодействия с сетями ЭВМ; базовые графические средства; дополнительные системы программирования на языках высокого уровня (Модула-2, Пролог, Кобол, Бейсик и др.); интерфейсные средства взаимодействия с системой.

## **Диалоговая единая мобильная операционная система для ЭВМ СМ 1820 (ДЕМОС 1820)**

ОС ДЕМОС 1820 — многопользовательская, многозадачная система, создаваемая для микрокомпьютеров с мультимнодоступом, построенных на базе микропроцессоров типа Intel 80286 (СМ 1820, РС/АТ, PS/2 моделей 50, 60) и других аппаратно-совместимых ВК.

ДЕМОС 1820 имеет мобильность, гибкость и модульность операционной среды, а также свойства, которые делают этот продукт коммерчески ценным как для деловых применений, так и для разработки ПО.

ДЕМОС 1820 состоит из трех самостоятельных систем: система разделения времени (Timesharing System), включающая ядро ДЕМОС 1820 и основной набор утилит; система разработки ПО (Software Development System), включающая системы и языки программирования, редакторы связей, отладчики и другие утилиты, необходимые для разработки программ (в том числе кросс-средства разработки приложений для ОС МЛОС и MS DOS); система обработки текстов (Text Processing System), в которую входят текстовые форматоры, макроблоки, утилиты текстовой обработки



Timesharing System является основной и обязательной системой, управляющей работой остальных двух систем ДЕМОС 1820.

Эти две системы являются дополнительными независимыми друг от друга системами и включаются в состав ДЕМОС 1820 по желанию пользователя при настройке операционной среды.

ДЕМОС 1820 обладает совместимостью с ОС Xenix System V/286 и Microsoft Xenix V.3.0, и приложения, разработанные для этих двух систем, будут выполняться в среде ДЕМОС 1820 без каких-либо модификаций.

ДЕМОС 1820 расширена поддержкой разработки приложений для ОС МДОС и MS DOS: в систему ДЕМОС 1820 включены утилиты для передачи файлов между форматированными дисками ДЕМОС и МДОС: эти системы поддерживают одинаковые стандарты языков программирования C Compiler, Pascal, Fortran, Cobol, Macroassembler и Basic Interpreter; ДЕМОС и МДОС поддерживают иерархическое файловые системы, независимые от устройств ввода-вывода, сходные системные вызовы: ДЕМОС 1820 включает полную среду кросс-разработки программ на языке C для МДОС и MS DOS; объектные модули программ, полученные в системе ДЕМОС 1820, идентичны по формату аналогичным модулям МДОС и MS DOS; набор библиотек ДЕМОС 1820 снабжен стандартными подпрограммами ОС МДОС, обеспечивая пользователя возможностями отладки версий программ для МДОС в среде ДЕМОС 1820.

В состав ДЕМОС 1820 включены средства поддержки межмашинного обмена — пакет Micnet, который обеспечивает поддержку дистанционного выполнения команд между двумя или более системами ДЕМОС и распределенной электронной почтой посредством соединений RS-232.

Помимо стандартных командных интерпретаторов sh и csh, в ДЕМОС 1820 реализован визуальный командный процессор для непрофессионального пользователя, построенный на основе распадающегося меню и аналогичный используемому в системах Multiplan, MS-Word, MS-Project и MS-Chart.

Каждый пользователь может максимально адаптировать для себя визуальную рабочую среду, изменяя размеры и месторасположение окон и удаляя или добавляя команды меню.

Кроме этого ДЕМОС 1820 имеет следующие возможности: работу в режиме деления времени; командные интерактивные языки; иерархическую файловую систему; аппарат управления процессами; средства межпроцессного обмена (семафоры, разделяемая память); средства разработки программ и сложных программных систем; средства разработки и сопровождения документации; процедуры настройки на комплекс технических средств; независимый ввод-вывод; управление одновременным доступом и корректировкой файлов; поддержку целостности файлов и записей файлов; динамическую переконфигурацию системы; распределенную электронную почту; развитую систему утилит-подсказок; средства быстрого запуска системы.

В ДЕМОС 1820 включены многие утилиты, разработанные в университете Беркли (Калифорния): vi, csh, style, diction, curses, termcap, more, mkstr, strings, nstr.

ДЕМЕС 1820 полностью согласуется со спецификациями System V Interface Definition (SVID) с точки зрения базового интерфейса, основных утилит, расширений базового текстового процессора и ПО.

В состав ДЕМОС 1820 входят следующие языки и системы программирования: Microsoft FORTRAN Optimizing Compiler, версия 4.0; Microsoft C Compiler, версия 4.0; Microsoft Pascal Compiler, версия 3.3; Microsoft Basic Interpreter, версия 5.4; Microsoft COBOL Compiler, версия 2.1; Microsoft COBOL Tools, версия 1.0; Microsoft BASIC Compiler, версия 5.7; Microsoft Macro Assembler, версия 4.0.

Минимальный состав технических средств, необходимый для функционирования ДЕМОС 1820; модуль управления системой МУС; модуль центрального процессора МЦП; модуль оперативный запоминающий МОЗ (1 Мбайт); контроллер внешней памяти КВП; видеоконтроллер цветной универсальный ВКЦУ; накопитель на жестком магнитном диске; накопитель на гибком магнитном диске; монитор цветной графический; клавиатура алфавитно-цифровая; устройство печатающее алфавитно-цифровое.

## Диалоговая многотерминальная система деления времени для решения информационно-логических задач ДИАМС-3 версии 3 (ДИАМС 3.3)

### Назначение системы

Диалоговая многотерминальная система деления времени для решения информационно-логических задач ДИАМС 3 версии 3 (ДИАМС 3.3) представляет собой развитие системы ДИАМС 3. Эта система ориентирована на создание и ведение баз данных (БД) большого объема, решение информационно-логических задач на различных конфигурациях ЭВМ СМ-4, СМ 1420 и программно-совместимых с ними.

Прототипом системы является ОС DSM 11.V3. Возможности ДИАМС 3.3 значительно расширены по сравнению с DSM 11.V3, в частности добавлена возможность работы с отечественными внешними устройствами, возможность работы с русским алфавитом, расширен набор команд, функций и системных переменных входного языка, усовершенствована работа на языке программирования Ассемблер, а также исправлены ошибки оригинала.

ОС ДИАМС 3.3 предназначена для использования в сложных территориально распределенных автоматизированных информационных системах административного управления, больших информационно-справочных системах, где требуются оперативное хранение и обработка больших объемов данных при одновременной работе до 63 пользователей.



Носителем ЕС ДИАМС 3.3 является системный диск, который содержит загружаемый образ системы, программы и глобальные массивы системного каталога, типы используемых дисков — DK, DM, DB, DP и DR. Несколько физических дисков можно объединять в единую среду хранения ВД — набора томов. БД располагается в том наборе дисковых томов, где расположен его каталог. В процессе функционирования ДИАМС 3.3 возможно монтирование (размонтирование) набора томов. Имеется возможность работы с частью дискового пространства как с устройством последовательного доступа, а также возможность физического чтения (записи) на диск средствами языка программирования.

ДИАМС 3.3 предоставляет пользователям широкий круг возможностей: многопрограммный режим выполнения заданий; работу в режиме программиста и в прикладном режиме; создание и ведение на дисках баз данных; простое обращение к широкому набору внешних устройств, входящих в номенклатуру технических средств СМ ЭВМ; одновременный доступ к базе данных многих пользователей (до 63) с различных, в том числе удаленных, терминалов; авторизацию доступа, защиту программ и данных; взаимосвязь между заданиями пользователей; генерацию версий системы под конкретную конфигурацию технических средств и заданные функции; оперативную модификацию конфигурации системы; организацию распределенных баз данных на многомашинных комплексах; использование подсистемы спулинга как для буферизации вывода результатов, так и для их временного хранения; возможность оперативного изменения приоритетов исполняемых заданий; диагностический контроль ошибок при работе как системных программ, так и программ пользователей.

В табл. 2 приведено сравнение функциональных возможностей ОС ДИАМС 3.3, ДИАМС 3 и ДИАМС 2.

Основой языка программирования является язык MUMPS ANSI/MDC XII.1, 1984 г. Язык ДИАМС ориентирован на обработку текстовой информации и ведения базы данных иерархической древовидной структуры. В язык ДИАМС встроены средства работы с обобщенными деревьями (дерево с узлами, упорядоченными на каждом уровне индексации в пределах родителя), что позволяет строить не только иерархические базы данных, но и с незначительными затратами имитировать реляционные и сетевые структуры. При этом достигаются значительно более высокие эксплуатационные характеристики (надежность, время доступа, простота модификации), чем при использовании систем управления БД (СУБД) — «Сетор», «Квант» и т. д.

Входной язык ДИАМС, являясь языком высокого уровня интерпретирующего типа, позволяет пользователю непосредственно с терминала создавать, отлаживать и выполнять программы. В языке отсутствуют описатели типа данных. Тип данных определяется динамически.

Язык ДИАМС состоит из команд и функций. Система ДИАМС 3.3 в процессе функционирования создает ряд системных переменных, доступных пользователю через команды и функции языка, и позволяет командам и функциям языка использовать их в процессе программирования. Перечень системных переменных существенно расширен по сравнению с ДИАМС 3.

Функциональная возможность	ДИАМС 2	ДИАМС 3	ДИАМС 3.3
Стандарт языка	MUMPS ANSI, 1977 г.	MUMPS ANSI, 1984 г.	MUMPS, ANSI, 1984 г., расширен
Динамическое построение выбранного варианта при загрузке	Нет	Да Операционные модули берутся из ОЗУ	Да Операционные модули берутся с МД
Монтируемые наборы томов	»	До 4	До 8
Доступ к массивам других пользователей через передаточную таблицу кнопок	»	Да	Да
Резидентные в ОЗУ программы	»	»	»
Динамически загружаемые наборы ⊗* ZCALL — функций	»	Нет	»
Загружаемые драйверы	»	Да	»
Возможность разработки и подключения драйверов нестандартных устройств	»	»	»
Ввод (вывод) на перфоленду	Да	Нет	Да
Системный журнал	МЛ	МЛ, МД	МЛ, МД
Защита программ от просмотра и модификации	Нет	Нет	Да
Дисплейно-независимые коды в командах	»	»	»
Экранные редакторы	»	»	»
Внутрисистемная программа физического копирования томов	»	МЛ, МД	МЛ, МД
Разделяемый буфер VIEW	»	Да	Да
Построение гибридных программ (ДИАМС+АССЕМБЛЕР)	»	Нет	»
Построение ДИАМС-функций пользователя	»	»	»
Вызов ДИАМС-программ с параметрами	»	»	»
ДИАМС-отладчик	»	Да	»
Получение справочной информации при работе системы («?» в командном режиме)	»	Нет	»

Во входной язык системы ДИАМС 3.3 включен ряд возможностей, предложенных Международным

\* ⊗ данный знак соответствует





комитетом развития MUMPS: функция  $\otimes$  HIGH — обратный относительно  $\otimes$  ORDER переход по индексам; функция  $\otimes$  GET — присваивание значения переменной; вызов подпрограмм с передачей параметров; функции пользователя, написанные на входном языке (функции  $\otimes \otimes \dots$ ).

Помимо этого, входной язык расширен рядом допустимых с точки зрения стандарта Z-функций, команд и системных переменных, расширяющих функциональные возможности системы.

ОС ДИАМС 3.3 полностью резидентна в ОЗУ, за счет чего обеспечивается достаточно высокая реактивность системы.

## Применение

Целесообразность использования ОС ДИАМС 3.3 определяется ее функциональными характеристиками.

Система ДИАМС 3.3 поддерживает следующие внешние устройства: различные типы дисковых ВЗУ; магнитные ленты; гибкие диски; АЦПУ; различные типы терминалов; мультиплексоры; сетевой микропроцессор.

На этапе генерации пользователь определяет состав технических и программных средств системы.

ПО полностью резидентно в оперативной памяти и состоит из четырех функциональных элементов ОС: все четыре функциональных элемента постоянно взаимодействуют друг с другом. В рамках общего обзора системы каждый из них описан как самостоятельный элемент.

**Диспетчер разделения времени** представляет собой системный супервизор, который управляет функционированием системы ДИАМС 3.3 в режиме разделения времени. Каждому новому пользователю он выделяет следующий доступный раздел. Диспетчер разделения времени передает управление системой от одного пользователя к другому, оптимизируя использование технических ресурсов (центральный процессор, диски).

**Монитор ввода-вывода** управляет работой таких периферийных устройств, как магнитная лента, гибкие диски, печатающие устройства и терминалы. Он обращается к устройствам ввода-вывода и поддерживает их работу с помощью соответствующих механизмов прерывания. Монитор ввода-вывода управляет также терминальным вводом-выводом, обеспечивая связь с интерпретатором с помощью буферов. Следя за заполнением и освобождением этих буферов, монитор может совмещать вывод с обработкой программы, требующей ввода-вывода.

**Интерпретатор языка** обеспечивает исполнение операций по выполнению программ, написанных на языке ДИАМС системы ДИАМС 3.3, и по управлению ими. В частности, интерпретатор проверяет и анализирует каждый очередной оператор языка и выполняет определенную этим оператором операцию. Интерпретатор также считывает программы с диска и записывает их на диск.

Интерпретатор осуществляет проверку правильности программы при ее выполнении и выдает сообщения обо всех ошибках языка на терминал (при отсутствии программной обработки ошибок).

Интерпретатор дает возможность осуществлять отладку программ непосредственно за терминалом.

**Супервизор БД** управляет логическим и физическим размещением информационных дисковых структур, которые образуют базу данных в системе ДИАМС 3.3. Дисковые структуры, в соответствии с которыми организовано хранение информации в базе данных, называются блоками. В блоках хранятся записи данных, к которым вы обращаетесь символическим образом с помощью глобальных переменных. Индексированные глобальные переменные образуют массивы, записи данных о которых хранятся в блоках в соответствии с внутренней древовидной структурой каждого из таких массивов.

Супервизор БД использует только то пространство дисковой памяти, которое необходимо для хранения имеющих значения элементов глобального массива. Когда глобальный массив уже не требует того дискового пространства, которое было ему выделено или ему уже не нужна часть этого пространства, супервизор баз данных освобождает соответствующие блоки и передает их в системный пул доступной дисковой памяти.

К сервисным средствам системы относятся следующие системные возможности: ведение системного журнала; использование спулинга; использование устройства системы последовательного доступа (СПД); системные и библиотечные программы.

С другими ОС ДИАМС 3.3 имеет возможность обмена данными через магнитные ленты и гибкие диски.

ОС ДИАМС 3.3. позволяет легко обращаться к широкому набору внешних устройств. Можно выделить два типа устройств: реальные физические устройства (терминал, АЦПУ, магнитная лента) и виртуальные (СПД, взаимосвязь задач через ОЗУ, флаговые устройства). Однако за счет специальной системы номеров устройств и унификации всех процедур обмена для пользователя эти устройства ничем не отличаются. Номера устройств неизменны и не зависят от конфигурации технических средств. Если при генерации устройство не было включено в систему, его номер остается незанятым и все попытки обратиться к нему приведут к ошибке.

При выполнении операции ввода-вывода программисту не нужно знать специфические характеристики периферийных устройств, поскольку передача данных осуществляется с помощью строк, не превышающих 255 символов. Однако существуют определенные физические характеристики этих устройств, которые могут быть интересны программисту, например, перемотка магнитной ленты, ограничение поля вывода на терминал и т. д.

Числа считаются специальным случаем строки. С этим свойством языка связаны отсутствие операторов описания данных и автоматическое преобразование типов операндов, участвующих в выражении.

Строка — это любая последовательность символов длиной не более 255 символов. Число в ДИАМС 3.3 имеет знак и плавающую точку. Во всех арифметических операциях система ДИАМС 3.3 может выдавать результат с точностью до 31 знака после запятой. (Точность устанавливается при настройке системы).



## Диалоговая система разделения времени для ЭВМ класса «Электроника-85» (ДИАМС 85.1)

Разработка ОС разделения времени ДИАМС 85 версии 1 (ДИАМС 85.1) обусловлена расширением сферы применения ДИАМС, в настоящее время широко распространенных для ЭВМ с архитектурой «Общая шина» (ОШ), благодаря совместимости и преемственности прикладного ПО.

ОС ДИАМС 85.1 представляет собой развитие системы ДИАМС 3.3 для работы на ЭВМ типа «Электроника-85» и программно-совместимых с ней.

Прототипом ДИАМС 3.3, в свою очередь, является ОС DSM 11.V3. Возможности ДИАМС 3.3 (следовательно, и ДИАМС 85.1) расширены по сравнению с DSM 11.V3. В частности, добавлена возможность работы с русским алфавитом, расширен набор функций и системных переменных входного языка, усовершенствована работа на языке программирования Ассемблер.

Система ДИАМС 85.1 ориентирована на создание и ведение, решение информационно-логических задач, создание справочно-информационных систем, систем делового применения.

ОС ДИАМС 85.1 поставляется на ГМД. В процессе установки системы пользователю необходимо провести настройку ПО на требуемые функциональные возможности и перенести систему на диски винчестерского типа.

Носителем ОС ДИАМС 85.1 является системный диск винчестерского типа, который содержит загружаемый образ системы, программы и глобальные массивы системного каталога. ГМД используются для размещения БД. В процессе функ-

ционирования ДИАМС 85.1 возможно монтирование (размонтирование) набора томов. Имеется возможность работы с частью дискового пространства как с устройством последовательного доступа, а также возможность физического чтения (записи) на диск средствами языка программирования.

ДИАМС 85.1 обеспечивает: управление оперативной памятью до 2 Мбайт; создание и ведение иерархической древовидной структуры баз данных на двух типах дисков — гибких и «Винчестер»; обслуживание операций ввода-вывода на внешних устройствах; взаимодействие задач пользователя; язык программирования высокого уровня интерпретирующего типа (ДИАМС), ориентированного на обработку текстовых данных переменной длины; работу в диалоговом и программном режимах; генерацию системы под заданные функции; диагностический контроль ошибок при работе системных и пользовательских программ; организацию работы резидентных программ; монтирование наборов томов дисков; организацию межпроцессорной связи для многомашиных комплексов.

Надежность функционирования ДИАМС 85.1 обеспечивается: защитой от несанкционированного доступа; авторизацией доступа к системе; корректировкой ошибок на устройствах; устойчивостью к ошибкам оператора; регистрацией и анализом состояния системы при аварийных ситуациях; возможностью сохранения и восстановления информации на носителе данных.

ОС ДИАМС 85.1 полностью резидентна в ОЗУ, за счет чего обеспечивается высокая реактивность системы.

Программное средство ДИАМС 85.1 функционирует на ЭВМ «Электроника-85» с ОЗУ от 512 Кбайт до 2 Мбайт и со следующей минимальной конфигурацией технических средств. ВЗУ на магнитном диске типа «Винчестер» емкостью 5000 (или 10000) Кбайт; ВЗУ на ГМД емкостью 400 Кбайт; терминалом; устройство печати.

## ЯЗЫКИ И СИСТЕМЫ ПРОГРАММИРОВАНИЯ

### Языки программирования операционной системы МОС ВП

#### Назначение и область применения

Существенное возрастание возможностей вычислительной техники, рост ее производства и расширение сфер применения привели к увеличению ее номенклатуры и сделали необходимым как развитие традиционных языков программирования, так и создание новых. Наиболее представительной в этом отношении является многофункциональная операционная система с виртуальной организацией памяти (МОС ВП) для 32-разрядной ЭВМ СМ 1700, в составе которой по-прежнему наиболее используемыми программистами являются такие языки программирования, как Фортран и Паскаль.

Практически в каждую новую модель ЭВМ в качестве основных компонент системного ПО входят системы программирования для указанных языков.

Наряду с традиционными прикладными программами перед значительной группой разработчиков встают задачи расширения системных программных компонентов для 32-разрядной ЭВМ СМ 1700. В этом случае эффективным инструментальным средством является язык Блисс, на котором реализована значительная часть МОС ВП.

Повышение производительности ВК и увеличение объемов запоминающих устройств создали предпосылки для разработки вычислительных систем пятого поколения, к числу которых можно отнести систему логического программирования Пролог. Система логического программирования Пролог, входящая в состав МОС ВП, представляет собой средство для реализации экспертных систем, систем автоматизированного проектирования и моделирования, а также других элементов си-



стем, представляющих возможность решения проблем искусственного интеллекта.

К числу наиболее сложных задач, решаемых на ЭВМ, относятся задачи реального времени. Этот класс задач отличается как размерами разрабатываемых программ, так и количеством их внутренних связей. Необходимость коллективной работы над большими программными системами, сопряженная с рядом естественных трудностей, связанных с организацией взаимодействия исполнителей, управлением, разработкой, стыковкой отдельно разрабатываемых программ в единое целое вызвали потребность в интегрированных системах разработки программ. Так, язык программирования Модуль-2 представляет собой совокупность функциональных и языковых средств, позволяющих эффективно работать с программой практически на всех этапах ее жизненного цикла, и обеспечивает функционирование разработанного ПО в режиме реального времени.

## Система программирования ФОРТРАН

В состав системы программирования Фортран входят такие основные компоненты, как компилятор FORTRAN, библиотека исполнительных процедур, библиотека справочных файлов.

Фортран вместе со средствами МОС ВП обеспечивает реализацию функций: компиляцию программных компонент, написанных на языке программирования Фортран; компоновку программы из отдельно откомпилированных программных компонент; выполнение программы; анализ причин возникновения ошибок; получение справочной информации.

**Создание исходных файлов программ.** Первым этапом разработки программ на языке Фортран является формирование исходного файла программы. В целях выполнения этой операции может быть использована программа редактирования текстов (редактор) операционной системы МОС ВП.

Редактор представляет собой диалоговую программу общего назначения и предусматривает два режима редактирования: редактирование строк, при котором операции осуществляются на отдельных строках исходного текста программы, и редактирование символов, при котором операции выполняются как с отдельными символами, так и с последовательностью символов.

Редактор обеспечивает защиту текстов пользователя и при прекращении редактирования сервисную поддержку в режиме редактирования строк с исходным текстом программы путем ввода команды HELP, а в режиме редактирования символов — нажатием клавиши «HELP» на клавиатуре терминального устройства.

Для вызова редактора вводится команда EDIT/EDT. Для прекращения выполнения команд редактора и формирования выходного файла по содержанию главного буфера для текстов следует использовать команду EXIT. При необходимости прекращения выполнения редактора без формирования выходного файла используется команда QUIT.

**Компилирование программ.** Для компилирования исходных файлов программ в объектные модули используется команда FORTRAN.

Основные функции компилятора языка Фортран SM 1700: проверка исходных операторов на языке Фортран и выдача сообщений в случае возникновения каких-либо ошибок; генерирование команд на машинном языке по исходным операторам программы на языке Фортран; группирование этих команд в объектный модуль для компоновщика; выдача подробного листинга компиляции с указанием всех обнаруженных в процессе компиляции ошибок.

Команда FORTRAN обеспечивает инициирование компиляции исходной программы. Квалификаторы команды указывают на подлежащие выполнению компилирующей программой специальные операции или особые свойства входных или выходных файлов. На подлежащие компиляции исходные файлы указывает перечень спецификаций файлов.

Листинг компиляции, выдаваемый согласно команде FORTRAN с квалификатором/LIST, состоит из секций: исходного кода, содержащей исходную программу в том виде, в котором она записана во входном файле; машинного кода, являющейся символическим представлением генерируемого компилятором объектного кода; структур запоминающих устройств с перекрестными ссылками; компилирования, содержащей данные о машинных ресурсах, затраченных на выполнение компилирующей программы.

**Компоновка и выполнение программы на Фортране.** Построение загрузочного модуля программы осуществляется компоновщиком LINK. С его помощью все ранее откомпилированные программные компоненты объединяются в единое целое и настраиваются для непосредственного исполнения.

Команда LINK обеспечивает инициализацию процесса компоновки объектного файла. Квалификаторы команды и квалификаторы файла указывают на варианты выходных и входных файлов. Перечень спецификаций файлов указывает на подлежащие компоновке объектные файлы.

Компоновщик обеспечивает формирование выполняемого загрузочного модуля, который имеет расширение файла EXE.

Инициализацию выполнения загрузочного модуля обеспечивает команда RUN. Спецификация файла в команде указывает на подлежащий выполнению загрузочный модуль.

При выполнении главной программой оператора END выполнение загрузочного модуля прекращается. В МОС ВП прекращение выполнения загрузочного модуля обуславливает осуществление системой разнообразных операций возврата в исходное: закрытие объектных файлов, высвобождение ресурсов системы и другие.

**Отладка программ.** При каждой операции компилирования, компоновки и выполнения программы пользователю предоставляется возможность указывать квалификаторы команд, влияющих на порядок обработки ошибок следующим образом: при компиляции квалификатор /DEBUG в команде FORTRAN обеспечивает выдачу символической информации, используемой программой-отладчиком; при компоновке квалификатор /DEBUG в команде LINK обеспечивает возможность ввода символиче-



ской информации в отладчик; при выполнении программы квалификатор /DEBUG в команде RUN осуществляет прерывание программы и вызов отладчика.

Отладку необходимо осуществлять тогда, когда возникает одна из ситуаций: сообщение об ошибках компиляции; сообщение об ошибках выполнения программы; ошибочные выходные данные при выполнении программы указывают наличие логической ошибки.

Отладчик позволяет управлять выполнением программы пользователя таким образом, чтобы пользователь располагал возможностями контролировать значения конкретных переменных, изменять эти значения, проверять последовательность выполнения программы, обнаруживать и устранять возникающие ошибки.

Отладчик имеет ряд полезных свойств, в том числе: функционирует в диалоговом режиме; воспринимает переменные и типы данных на языке Фортран; воспринимает компоненты, написанные на других языках программирования; предоставляет возможность обратиться за получением справочной информации с помощью команды HELP.

**Входные и выходные данные.** Основной структурной единицей информации системы программирования Фортран является файл ОС МОС ВП. Каждый файл определяется двумя основными атрибутами: именем и расширением имени. Имя программной компоненты должно быть, как правило, уникальным. Расширение имени используется для предоставления различных форм информации.

В системе программирования Фортран используются наиболее распространенные стандартные расширения имен:

- FOR — исходная программа на языке Фортран (входные данные для компилятора);
- COM — файл командных процедур;
- DAT — входной или выходной файл данных;
- DIR — справочный файл;
- EXE — загрузочный модуль (тип выходного файла для компоновщика и входного файла для загрузки и исполнения);
- HLB — файл справочной библиотеки;
- HLP — входной текст для библиотеки справочных файлов;
- JOU — журнальный файл для программы редактирования EDT;
- LIS — листинг компиляции (тип входного файла для команд PRINT, TYPE);
- LOG — выходной файл для групповой обработки данных;
- MAP — файл распределения памяти программы (перечень типов выходных файлов компоновщика);
- OBJ — объектный файл (тип выходного файла для компилятора, тип входного файла для компоновщика);
- OLB — библиотека объектных модулей;
- TLB — библиотека текстов входных данных для компилятора;
- TMP — временный файл;
- TXT — входной файл для библиотеки текстов.

## Система программирования ПАСКАЛЬ

В состав системы программирования Паскаль входят следующие программные компоненты: компилятор PASCAL; библиотека справочных файлов; текстовая библиотека.

Эти компоненты совместно с универсальными средствами МОС ВП — компоновщиком, символическим отладчиком, исполнительной библиотекой — обеспечивают реализацию следующих основных функций: компиляцию программных компонент, написанных на языке Паскаль; компоновку выполняемого образа программы из отдельно откомпилированных программных компонентов; выполнение созданной программы; анализ причин возникновения ошибок в процессе выполнения программы; получение справочной информации по языку Паскаль.

**Создание и компиляция программ.** Для создания и выполнения программ на языке Паскаль необходимо вводить последовательно следующие команды МОС ВП:

```
EDIT/EDT спецификация файла <ВК>  
PASCAL СФ <ВК>  
LINK СФ <ВК>  
RUN СФ <ВК>
```

Команда включает информацию системе о дальнейшем ходе работы. Каждая команда требует указания спецификации файла (СФ), определяющей файл для обработки. Допускается использование квалификаторов для файла и команды, позволяющих изменить способ обработки файла. Командная строка завершается при нажатии клавиши <ВК> (RETURN).

Исходный текст программы можно сформировать на устройстве прямого доступа любым из редакторов МОС ВП, например EDIT. Команда EDIT позволяет создавать исходный файл, содержащий операторы исходных программ или модулей. Для идентификации исходных файлов ПАСКАЛЬ применяется тип файла PAS.

После создания исходных программ их необходимо скомпилировать. Команда PASCAL вызывает компилятор Паскаль, который проверяет наличие ошибок в исходной программе. Компилятор преобразует входной исходный файл в двоичную форму для дальнейшей интерпретации компоновщиком. Компилированный код называется объектным модулем и записывается в файл объектного модуля, имеющий по умолчанию тип OBJ.

Формат:  
PASCAL СФ [...]

Параметры СФ определяют один или более исходных файлов, подлежащих компиляции. Если не определен тип входного файла, то по умолчанию используется тип файла PAS.

При разделении спецификации файлов запятыми (,) каждый файл компилируется отдельно. При разделении спецификации файлов знаком (+) файлы объединяются и компилируются как единый входной файл. При этом генерируются единый объектный файл и единый файл листинга.

Для указания специальных действий, выполняемых при компиляции над перечисленными файлами, используются квалификаторы.



Квалификаторы компилятора позволяют: генерировать код контроля времени выполнения; создавать листинг перекрестных ссылок; генерировать данные для символического отладчика; завершать компиляцию после указанного числа ошибок; включать представление машинных кодов в файл листинга; генерировать наиболее эффективный код для оптимизации компилируемой программы; создавать объектный файл; распечатывать сообщения, указывающие на использование расширений языка Паскаль; распечатывать диагностику для ошибок предупреждающего уровня.

Некоторые возможности, предоставляемые квалификаторами, могут быть получены с помощью указания переключателей режимов компиляции в тексте исходной программы.

При наличии в компилируемой модели ошибок компилятор выдает на терминал сообщения об ошибках.

**Компоновка и выполнение программ на языке Паскаль.** Объектный модуль не является выполняемым. Обычно объектный модуль содержит ссылки на программы или подпрограммы, находящиеся в других объектных модулях. Все эти модули должны быть объединены до выполнения программы. Объединение производится компоновщиком, вызываемым командой LINK.

Компоновщик разрешает все ссылки в объектном модуле и устанавливает абсолютные адреса для размещения символов. Компоновщик создает выполняемый образ, т. е. файл типа EXE, содержащий программу в выполняемом формате.

Для изменения выхода компоновщика, а также для включения возможностей отладки или трассировки могут быть использованы квалификаторы. Если квалификатор относится к команде, его действие распространяется на все перечисленные файлы. Если он относится к файлу, его действие распространяется только на этот файл.

Квалификаторы команды LINK подразделяются на три группы: файла образа; карты распределения памяти; отладки.

После создания выполняемого образа можно вводить команду для выполнения программы.

При первом запуске программа может выполняться неправильно. Если это вызвано логической ошибкой, обнаружить ее можно с помощью файла листинга или символического отладчика для проверки выходных данных программы. После установления причины ошибки следует повторить последовательность команд EDIT, PASCAL, LINK и RUN соответственно для корректировки и проверки результатов.

**Отладка программ, написанных на языке Паскаль.** При возникновении ошибки компилятор и исполнительная система выдают сообщения об ошибках и предупреждения. Данная информация может использоваться для локализации ошибки в программе и ее последующего исправления.

Логические ошибки и ошибки программирования могут быть обнаружены пользователем самостоятельно с помощью отладчика, упоминавшегося при рассмотрении системы программирования Фортран. Выполняемые отладчиком функции относительно компонент, реализованных на языке Паскаль, аналогичны.

**Входные и выходные данные.** Как и в системе программирования Фортран, основной структурной единицей информации системы программирования

Паскаль является файл операционной системы. Аналогично каждый файл определяется двумя атрибутами — именем и расширением имени.

Наряду с перечисленными при рассмотрении языка Фортран в системе программирования Паскаль используется такое стандартное расширение имени, как PAS-текст программной компоненты на языке Паскаль.

## Система программирования Блисс

Языку Блисс, ориентированному на системное программирование, присущи следующие основные свойства: высокооптимизированный объектный код; простые и согласованные средства для операций над адресами; управляющие конструкции, обеспечивающие хорошо структурированный исходный код, делающий программы надежными, понятными и легко сопровождаемыми; эффективные средства для определения представления сконструированных пользователем структур данных и способа доступа к данным таких структур; возможность доступа к аппаратуре ЭВМ и ОС; достаточно высокий уровень средств для определения соглашений о сопряжениях, используемых при вызове процедур.

**Состав системы программирования Блисс.** Система программирования включает компилятор, набор библиотек и файлов сервисной поддержки. Традиционные компоненты систем программирования (редактор, компоновщик, отладчик) являются универсальными аналогично рассмотренным ранее языкам.

Программирование на Блисс реализуется с помощью компилятора BLISS 32.EXE. Исходный текст программы на языке Блисс является входом компилятора, который генерирует из него неотредактированный объектный файл. При этом компилятор может выполнять сложную масштабную оптимизацию объектного кода. Компилятор не назначает абсолютные адреса процедур и данных, а выражает их через адресные смещения относительно определенных базовых адресов.

Блисс имеет специальные библиотеки XPORT.REQ и XPORT.OLB, в первой из которых содержатся текстовые определения, а во второй — процедуры времени выполнения. Вторая библиотека использует файл XPORT TMSG.EXE для выдачи сообщений во время выполнения программ Блисс.

Файлы поддержки предназначены для облегчения программирования и для обучения. К ним относятся:

- MODULE.BLI — образец построения модуля, процедуры;
- EZIO32.B32 — модуль упрощенного ввода-вывода;
- EZIO.DOC — описание EZIO32.B32;
- TUTIO.R32 — макросы примитивов ввода-вывода на терминал;
- TUTIO.DOC — описание TUTIO.R32;
- CONDIT.R32 — макросы обработки событий.

Кроме того, интерактивный режим работы программиста с Блисс поддерживает справочная система HELP.



Обслуживающие программы Блосс — это набор специальных программ, поддерживающих программирование на Блосс. К ним относятся программы BCREF.EXE, обеспечивающие построение таблицы перекрестных ссылок, и XDUMP.EXE, дампирующая структуры данных. BCREF.EXE при работе использует файлы BCMF.EXE и GCMF.EXE, содержащие сообщения программы BCREF.EXE.

**Процесс разработки программ на языке Блосс.** Реализация логических структур проекта программы на языке Блосс возможна через организацию взаимодействия множества процедур, оперирующих структурами данных.

Каждая процедура соответствует четко определенной, относительно независимой функции или подфункции проекта. Одна из процедур является главной. Эта процедура вызывается для выполнения при запуске программы. Она управляет выполнением других процедур, вызывая их, а те, в свою очередь, тоже могут вызывать процедуры, и так далее пока не выполняется все запланированные действия по обработке данных.

После проектирования структуры программы и структуры данных программируются на языке Блосс.

Процедуры объединяются в модули для компиляции. Каждый модуль представляется в виде текстового файла, подготовленного посредством редактора. Такой файл называется исходным файлом.

Запрограммированные модули, приведенные в виде исходных файлов, обрабатываются компилятором. Результатом компиляции каждого модуля является объектный файл, который представляет собой последовательность машинных инструкций и данных, закодированных для работы компоновщика, и эквивалентен исходному модулю.

Процесс компоновки осуществляется, когда все модули программы скомпилированы. Компоновщик связывает различные объектные модули, ссылающиеся друг на друга, дополняет программу процедурами из общей библиотеки и превращает закодированные компилятором относительные адреса в действительные машинные адреса.

Результатом компоновки является загрузочный файл, содержащий выполняемый образ программы, который может выполняться при успешном завершении процедуры компоновки. Первые прогоны программы выполняют с помощью отладчика. Он позволяет проследить ход выполнения разрабатываемой программы, передачи управления, возникающие условия, изменения в данных, что значительно облегчает отладку программы. После отладки программа готова к эксплуатации.

**Входные и выходные данные.** Входные данные компилятора — это исходные файлы модулей Блосс, имеющие расширения B32, BLI, R32, REQ.

Выходные данные, полученные в результате компиляции, следующие:

- OBJ — объектные файлы;
- L32 — библиотечные файлы;
- LIS — файлы листинга;
- CRF — файлы перекрестных ссылок.

Файлы типа OBJ и L32 могут затем использоваться компоновщиком для получения выполняемых образцов программ Блосс.

## Система логического программирования ПРОЛОГ

Отличительными факторами СП ПРОЛОГ, послужившими причинами быстрого роста ее популярности в приложениях, связанных с обработкой символьной информации, являются: наличие мощных средств, заменяющих оператор присваивания в обычных языках; возможность возврата назад по дереву выбора решения, что позволяет процедуре автоматически генерировать последовательность альтернативных решений данной задачи; общность представления структур данных и утверждений языка в виде деревьев, что позволяет обрабатывать на языке Пролог сами утверждения программы; гибкость процедур языка Пролог, в которых можно не объявлять заранее, какие аргументы являются входными, а какие — выходными. При этом одни и те же аргументы при одном обращении являются входными, а при другом — выходными. Процедуры на языке Пролог могут также возвращать в качестве выходных значений неполные структуры данных, содержащие переменные, значения которых не были установлены. Эта неполнота может быть затем снята последующими процедурами: краткостью, выразительностью и относительной легкостью, с которыми пишутся программы на языке Пролог, хотя переход к использованию нетрадиционного описательного стиля программирования может представлять некоторые трудности.

**Состав и функции ПРОЛОГ.** В состав системы ПРОЛОГ входят интерпретатор в виде исполняемого образа PROLOG.EXE и ряд вспомогательных программ, написанных на самом языке Пролог и загружаемых перед началом эксплуатации системы в БД интерпретатора. После загрузки введенная из вспомогательных программ информация защищается от несанкционированного изменения.

Интерпретатор языка Пролог обеспечивает выполнение следующих функций: проведение инициализации БД; ввод и синтаксический контроль утверждений программы; занесение утверждений программы в БД; ввод и синтаксический контроль запросов к БД; согласование запросов с БД; вывод ответов на запросы; обработка вызовов встроенных предикатов; управление памятью; интерфейс с ОС.

Вспомогательные программы обеспечивают выполнение следующих функций: определение встроенных предикатов языка Пролог через внутренние параметры интерпретатора; определение дополнительных сервисных предикатов; определение процедур сортировки; определение процедур работы с множествами; защита встроенных предикатов Пролог от несанкционированного изменения.

**Описание задачи.** В большинстве языков программирования программа является описанием алгоритма. Операторы программы задают последовательность шагов, предписывающих, как идти к намеченной цели, но сами по себе они значения не имеют.

Такой стиль программирования является описательным, декларативным, и ему соответствует де-



кларативная семантика языка. Например, следующая программа:

БОЛЬШЕЕ -ИЗ (-X, -X, -X).  
БОЛЬШЕЕ -ИЗ (-X, -X, -Y):— -X>-Y.  
БОЛЬШЕЕ -ИЗ (-Y, -X, -Y):— -X<-Y.

представляет собой определение отношения между двумя числами с именем «БОЛЬШЕЕ -ИЗ». Первый аргумент — большее из двух чисел, описанных вторым и третьим аргументами отношения.

Первое правило безусловно, т. е. является фактом, и говорит, что если сравниваются два одинаковых значения, то само это значение является большим из двух.

Второе правило утверждает, что из двух значений -X и -Y большим является -X, если выполняется неравенство -X>-Y.

Третье правило противоположно второму.

Каждое из правил является истинным высказыванием об отношении «БОЛЬШЕЕ -ИЗ».

Для проведения вычислений в языке Пролог требуется задать вопрос, который будет согласовываться интерпретатором с утверждениями, находящимися в БД. Процесс вычислений в языке Пролог состоит в поиске таких значений переменных из запроса, с которым пользователь обращается к БД, при которых каждое высказывание вопроса является следствием из утверждений, находящихся в БД.

Осуществляется это посредством просмотра всех утверждений БД для каждого высказывания вопроса и сравнения этого высказывания с выводом (левой частью утверждения), содержащимся в утверждении (факте или правиле).

Если высказывание вопроса и вывод, содержащийся в утверждении, совпадают, то условия, заданные в утверждении-правиле, образуют новый вопрос, ответ на который дает решение для данного высказывания. Процесс согласования прекращается, когда в БД найдены утверждения, согласующие всю цепочку выводов от исходного высказывания вопроса, или когда в БД отсутствуют такие утверждения.

Такое сведение высказывания к новому высказыванию есть программирование, основанное на выборе правил по шаблону. Этот метод программирования все больше используется в работах по искусственному интеллекту, в особенности в экспертных системах.

**Входные и выходные данные.** Входные данные ПРОЛОГ — это совокупность утверждений БД (фактов и правил), описывающих объекты решаемой задачи и взаимосвязи между этими объектами, и вопросы, которые интерпретатор пытается согласовать с утверждениями БД.

Утверждения могут вводиться в БД как с терминала, так и из файлов. Вопросы обычно вводятся с терминала, но могут присутствовать и в самом файле. В этом случае они называются командами и выполняются без выдачи ответов.

После согласования вопроса с БД интерпретатор сообщает на терминал, как завершился процесс согласования: успешно или нет.

Если при вводе и обработке утверждений интерпретатор встретится с ошибкой, то на терминал будет выведено сообщение, информирующее пользователя.

## Система программирования

### МОДУЛА-2

Основные характеристики системы программирования МОДУЛА-2: поддержка концепции структурного программирования; надежность создаваемого ПО за счет наличия хорошо специфицированного, независимо от реализации интерфейса между отдельными программными компонентами (модулями); наличие средств для программирования задач реального времени; возможность создания мобильных легко модифицируемых программ.

**Состав и функции.** Система программирования МОДУЛА-2 обеспечивает реализацию следующих основных функций: компиляцию компонент, написанных на языке Модула-2; компоновку выполняемого образа из отдельно откомпилированных программных компонент; выполнение созданной программы.

В состав системы МОДУЛА-2 входят следующие функциональные компоненты: резидентный монитор MODULA; компилятор COMP; компоновщик LINK; генератор выполняемого образа EXE-GEN.

Резидентный монитор MODULA является ядром системы и осуществляет управление работой всех остальных ее компонентов.

Компилятор COMP обеспечивает компиляцию исходных текстов программ, написанных на языке Модула-2, и формирование файлов промежуточного кода для каждой, отдельно откомпилированной компоненты.

С помощью компоновщика LINK все ранее откомпилированные программные компоненты объединяются в единое целое и настраиваются для непосредственного исполнения. В результате работы компоновщика формируется файл загрузочных кодов программы. Выполнение этого файла осуществляется под управлением резидентного монитора MODULA. Следует подчеркнуть, что в рассматриваемой реализации системы МОДУЛА-2 компоновщик LINK в отличие от рассмотренных ранее языков не является универсальным. Это специальная индивидуальная компонента, входящая в систему МОДУЛА-2.

С помощью генератора выполняемого образа EXE-GEN файл загрузочных кодов программы может быть преобразован в файл выполняемого образа, выполнение которого может быть инициировано непосредственно из операционной системы.

**Программирование и выполнение.** Исходный текст программы на языке Модула-2 готовится с помощью редактора EDIT.

Компиляция программ, написанных на языке Модула-2, выполняется компилятором COMP, который обеспечивает: отдельную компиляцию программных компонент; полный контроль интерфейсов между связанными отдельно компилируемыми компонентами; выдачу подробного листинга компиляции с указанием всех обнаруженных в процессе компиляции ошибок; генерацию эффективного промежуточного кода программы, основанного на стековом распределении динамической памяти.



Построение файла загрузочного кода программы выполняется компоновщиком LINK. С его помощью все ранее откомпилированные программные компоненты объединяются в единое целое и настраиваются для непосредственного исполнения.

Выполнение файла загрузочного кода производится под управлением резидентного монитора MODULA, осуществляющего: загрузку файла загрузочного кода в оперативную память и запуск программы; обработку ошибочных ситуаций, возникающих при выполнении программы; формирование файла образа памяти программы при ее аварийном завершении для использования при последующем анализе причин возникновения ошибки; организацию связи программы пользователя с операционной системой; поддержку программ, написанных на языке Модула-2, в период их выполнения.

Генератор выполняемого образа EXEGEN позволяет получить файлы выполняемого образа программы, разработанной на языке Модула-2, для запуска ее непосредственно из ОС.

**Входные и выходные данные.** Основной структурной единицей информации системы МОДУЛА-2 является файл ОС. Каждый файл определяется двумя атрибутами — именем и расширением имени.

Файлы, содержащие информацию об одной программной компоненте, должны иметь одно и то же имя — имя программной компоненты. Расширение имени используется для идентификации различных форм представления информации.

В системе программирования МОДУЛА-2 используются следующие стандартные расширения имен:

MOD — текст программной компоненты на языке Модула-2;

DEF — текст компоненты определения на языке Модула-2;

LST — листинг компиляции;

LNK — файл промежуточного кода программной компоненты;

REF — файл ссылок программной компоненты;

SYM — файл символов компоненты определения;

LOD — файл загрузочного кода программы;

MAP — файл распределения памяти программы.

Информация, выдаваемая на различных стадиях разработки программы, может затем являться входной на последующих этапах работы.

## Применение языков программирования

Для функционирования систем программирования ФОРТРАН, ПАСКАЛЬ, БЛИСС, ПРОЛОГ, МОДУЛА-2 необходим следующий минимальный состав технических средств из номенклатуры ВК СМ 1700: процессор 32-разрядный; оперативное запоминающее устройство емкостью не менее 2 Мбайт; накопитель на сменных магнитных дисках типа СМ 5408 емкостью 14 Мбайт; видеотерминал типа ВТА 2000-15М.

Для получения твердой копии текстов программ и другой информации о работе систем программирования необходимо устройство печати растровое типа СМ 6334 или консольное устройство печати с клавиатурой типа СМ 6380.

Компоненты систем программирования ФОРТРАН, ПАСКАЛЬ, БЛИСС, ПРОЛОГ, МОДУЛА-2 поставляются на магнитной ленте с плотностью записи 1600 бит/мм и для установки на ВК требуют наличия устройства внешней памяти на магнитной ленте типа СМ 5309.

Операционной средой для систем программирования является МОС ВП СМ 1700.

## Система программирования МОДУЛА-2 — средство разработки встроенных систем

### Назначение и область применения

Система предназначена в качестве языка программирования для мини-ЭВМ, обладает важной концепцией «модуль» и систематическим, современным синтаксисом, простотой изучения, структурируемостью создаваемых программ, широким набором средств описания структур данных, строгостью использования типов. В дополнение к классическим свойствам система обеспечивает программирование задач реального времени, моделирование параллельных вычислений и обработку прерываний. Язык Модула-2 обладает возможностями решения задач системного программирования.

Язык является существенно машинно-независимым, за исключением ограничений, связанных с размером машинного слова. Все машинно-зависимые средства выделены в отдельные модули (ввод-вывод, управление файлами, памятью процессами).

Система представляет хорошие возможности для коллективного труда большой группы разработчиков, занимающихся созданием крупных программных систем, обеспечивая независимость от реализации интерфейсы между отдельно разрабатываемыми модулями. Это позволяет: применять язык как средство проектирования программного обеспечения; автоматизировать контроль правильности использования глобальных структур данных и процедур уже на стадии проектирования; легко модифицировать реализацию программы, оставаясь в рамках проекта.

Одной из компонент системы программирования МОДУЛА-2 является специальная библиотека, обеспечивающая программирование встроенных систем (СПВС), которая будет далее рассмотрена отдельно.

### Функции системы программирования МОДУЛА-2

В состав системы программирования МОДУЛА-2 входят следующие компоненты: резидентный монитор MODULA; компилятор COMP; компоновщик LINK; отладчик DEBUG.

Инструментальные средства системы программирования обеспечивают выполнение следующих



основных функций: компиляцию программных компонент, написанных на языке Модула-2 и генерацию файла промежуточного кода; построение файла загрузочного кода программы; выполнение программы с возможностью анализа причин возникновения ошибок исполнения.

Компиляция программных компонент, написанных на языке программирования Модула-2, выполняется с помощью компилятора СОМР, обеспечивающего: отдельную компиляцию программных компонент; полный контроль интерфейсов между связанными, отдельно компилируемыми компонентами; выдачу подробного листинга компиляции с указанием всех обнаруженных в процессе компиляции ошибок; генерацию эффективного промежуточного кода программы, основанного на стеквом распределении динамической памяти; возможность получения дополнительной информации о программной компоненте, используемой в дальнейшем для отладки программы в терминах исходного текста.

Построение файла загрузочного кода программы осуществляется компоновщиком LINK. С его помощью все ранее откомпилированные программные компоненты объединяются в единое целое и настраиваются для непосредственного исполнения.

Выполнение файла загрузочного кода осуществляется под управлением специального резидентного монитора MODULA, в функции которого входят: загрузка файла загрузочного кода в оперативную память и запуск программы; обработка ошибочных ситуаций, возникающих при выполнении программы; формирование файла образа памяти программы при ее аварийном завершении для использования при последующем анализе причин возникновения ошибки; организация связи программы пользователя с ОС; поддержка программ, написанных на языке Модула-2 в процессе их выполнения.

Анализ причин аварийного завершения программы может быть выполнен с помощью отладчика DEBUG, обеспечивающего: просмотр цепочки процедур, выполнявшихся в момент возникновения ошибки; анализ значений переменных в момент аварийного завершения; выдачу информации в терминах исходного текста программы.

Основной структурной единицей информации системы программирования МОДУЛА-2 является файл ОС.

Каждый файл определяется двумя основными атрибутами: именем и расширением имени. Файлы, содержащие информацию об одной программной компоненте должны, как правило, иметь одно и то же имя — имя программной компоненты. Расширение имени используется для идентификации различных форм представления информации.

В системе программирования МОДУЛА-2 используются следующие стандартные расширения имен:

MOD — текст программной компоненты на языке Модула-2;

DEF — текст компоненты определения на языке Модула-2;

LST — листинг компиляции;

LNK — файл промежуточного кода программной компоненты;

SYM — файл символов компоненты определения;

LOD — файл загрузочного кода программы;

MAP — файл распределения памяти программы.

Входной информацией для созданной системы является информация, отражающая состояние разрабатываемого ПО на различных этапах его создания. Кроме того, на вход подается последовательность команд, определяющая порядок выполняемых действий.

Выходом является созданное с помощью системы программирования МОДУЛА-2 ПО, а также дополнительная информация, обеспечивающая процесс разработки и сопровождения.

Информация, выдаваемая на ранних стадиях разработки программы, затем может являться входной на последующих этапах работы.

## Средства программирования встроенных систем

Язык Модула-2 является эффективным средством разработки ПО встроенных систем. Инструментальные средства разработки ПО встроенных вычислительных систем на базе 16-разрядных микропроцессоров с интерфейсом «Общая шина» представляют собой специальный набор (СПВС), функционирующий вместе с системой МОДУЛА-2.

СПВС состоит из трех основных компонент: исполняющей системы языка Модула-2; обслуживающих программ; библиотеки специальных модулей.

Библиотека специальных модулей представляет разработчику набор основных понятий и процедур, необходимых для разработки ПО встроенных систем. Все эти модули реализованы на языке Модула-2 и обеспечивают выполнение следующих основных функций: управление задачами; учет реального времени; организацию взаимодействия между задачами; обмен с видеотерминалом; управление объектом с помощью устройств дискретного и аналогового ввода-вывода; межмашинный обмен в соответствии с протоколом SECS-1.

Обслуживающий программы СПВС включают в себя: специальный компоновщик LINKS для программ, написанных на языке Модула-2; генератор LDA-файлов LDAGEN для программ на языке Модула-2; программу формирования характеристик внешних устройств EXTDEF.

Компоновщик LINKS осуществляет построение независимых программ на языке Модула-2, пригодных для размещения в постоянном запоминающем устройстве.

С помощью генератора LDAGEN программа на языке Модула-2 преобразуется в формат LDA, используемый абсолютным загрузчиком для переноса на встроенную ЭВМ.

Программа EXTDEF используется для формирования характеристик внешних устройств, используемых программой на языке Модула-2, и для преобразования информации в файл формата LDA.

Все обслуживающие программы написаны на языке Модула-2.

Исполняющая система языка Модула-2 обеспечивает поддержку функционирования ПО в период



выполнения на встроенной ЭВМ. Это единственная компонента, написанная на ассемблере.

Процесс разработки ПО встроенных систем с помощью СПВС осуществляется на инструментальной ЭВМ.

ПО встроенной системы, созданное с применением СПВС, состоит из трех частей: исполняющей системы языка Модула-2; собственно программы на языке Модула-2, осуществляющей процесс управления; информации для связи программ на языке Модула-2 с внешними устройствами.

Каждая из этих частей представляется отдельным файлом формата LDA для загрузки во встроенную ЭВМ.

Исполняющая система языка Модула-2 поставляется пользователю в готовом виде и не требует разработки.

Разработка программы для встроенной ЭВМ осуществляется на языке программирования Модула-2 с использованием библиотеки специальных модулей СПВС. Процесс разработки в целом аналогичен обычному процессу создания независимой программы на языке Модула-2 и включает компиляцию исходных текстов программы с помощью компилятора COMP с последующей обработкой компоновщиком LINK. При необходимости получения ПО, пригодного для размещения в постоянном запоминающем устройстве (ПЗУ), необходимо использовать специальный компоновщик LINKS.

Полученный в результате работы компоновщика файл загрузочных кодов преобразуется затем с помощью обслуживающей программы LDAGEN в файл формата LDA.

Разработанная с помощью СПВС программа для встроенной ЭВМ требует задания дополнительной информации о конкретных физических характеристиках внешних устройств. Для создания такой информации используется обслуживающая программа EXTDEF. Входными данными программы EXTDEF являются характеристики внешних устройств, задаваемые пользователем в диалоговом режиме. Выходом программы является файл

формата LDA, содержащий все эти характеристики в специальной форме представления.

## Применение

Система программирования Модула-2, включая и СПВС, функционирует в среде операционной системы ОС РВ 3.0 на вычислительном комплексе СМ 1420 или совместимых с ним.

Для нормального функционирования системы необходим следующий минимальный состав технических средств: 16-разрядный процессор ОЗУ объемом не менее 128 Кбайт; устройство внешней памяти на магнитных дисках типа СМ 5400 или других типов, поддерживаемых ОС РВ; видеотерминал типа ВТА 2000 или других типов, поддерживаемых ОС РВ.

Создаваемое с помощью СПВС ПО осуществляет поддержку следующего комплекса технических средств встроенной ЭВМ: процессор типа «Электроника-60»; ОЗУ или ПЗУ объемом до 56 Кбайт; видеотерминал типа «Электроника 015-ИЭ»; восемь устройств дискретного ввода типа 15-КВ-60/32-001; восемь устройств дискретного вывода типа 15-КВВ-60/30-001; четыре аналого-цифровых преобразователя (АЦП) типа 15-КА-60/8-010; восемь цифро-аналоговых преобразователей (ЦАП) типа 15-КА-60/4-009; блок управления БУ-3 ОРИОН-3 М3.858.117; шестнадцать устройств последовательного обмена (УПО) типа «Электроника МС 4601».

ПО системы МОДУЛА-2 и СПВС поставляется на магнитной ленте. Поэтому инструментальный комплекс на момент установки МОДУЛА-2 и СПВС должен быть оснащен устройством внешней памяти на магнитной ленте типа СМ 5300.

Разработанное на инструментальной ЭВМ ПО встроенной системы переносится на встроенную ЭВМ либо через перфолену, либо через канал межмашинной связи.

## ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА ПРОГРАММИРОВАНИЯ

### Инструментальная система программирования ТУРБО ПАСКАЛЬ (СП ТУРБО ПАСКАЛЬ)

#### Назначение и основные функции

Комплекс инструментальных средств ТУРБО ПАСКАЛЬ представляет собой аппарат для создания, редактирования, трансляции и отладки программ на языке Паскаль. Данный комплекс функ-

ционирует под управлением МДОС 1810 (МДОС 1820).

Инструментальная СП ТУРБО ПАСКАЛЬ предназначена для использования при разработке автоматизированных систем различного назначения, функционирующих на базе СМ 1810 и программно-совместимых с ней ЭВМ.

Инструментальная система программирования представляет собой комплекс интегрированных инструментальных средств, включающий в себя компилятор интегрированной среды и автономный компилятор, использующий командную строку вызова.

Система обеспечивает выполнение следующих основных функций: ввод и редактирование тексто-



вых файлов, в том числе исходных текстов программ на языке Паскаль; компиляцию программ, написанных на языке Паскаль (с записью результата компиляции как на внешние носители, так и в оперативную память с последующим выполнением); выполнение скомпилированных программ с контролем ошибок времени выполнения.

СП ТУРБО ПАСКАЛЬ позволяет разделить программу на модули, каждый из которых занимает до 64 Кбайт памяти, обеспечивая таким образом выполнение программ, занимающих более 64 Кбайт. Благодаря модульности построения программ обеспечивается высокая скорость компиляции до 27 000 строк/мин.

СП ТУРБО ПАСКАЛЬ содержит мощную библиотеку стандартных модулей, обеспечивающую использование полных возможностей дисплея и клавиатуры, включая управление режимом экрана, цветом, звуком, окнами; широкие графические возможности; библиотеку исполняющей системы, включающую различные операции обработки файлов, динамическое распределение памяти и т. д.

СП ТУРБО ПАСКАЛЬ содержит многофункциональный встроенный экраный редактор, обеспечивающий редактирование, компиляцию, обнаружение и исправление ошибок в интегрированной среде.

Кроме того, система программирования содержит встроенное математическое обеспечение для арифметического сопроцессора.

## Состав системы

В состав системы программирования входят следующие компоненты: компилятор интегрированной среды; компилятор, использующий командную строку; библиотека стандартных модулей; сервисная программа настройки интегрированной среды; сервисная программа, создающая карту распределения памяти для использования символического отладчика; сервисная программа, использующаяся для управления библиотеками стандартных модулей; другие сервисные программы.

**Использование интегрированной среды.** Интегрированная среда предоставляет пользователю возможность интерактивной работы с системой программирования. Пользователь может создать и отредактировать текст программы, скомпилировать и выполнить ее, не выходя из интегрированной среды.

Управление работой СП ТУРБО ПАСКАЛЬ осуществляется путем выбора из представленных на экране вариантов меню и выполнения их. Выбор и выполнение соответствующего режима можно осуществлять несколькими способами.

СП ТУРБО ПАСКАЛЬ предоставляет пользователю многоуровневое меню для некоторых режимов. Возврат из меню нижних уровней в меню более высокого уровня происходит при нажатии клавиши <E.SIC>.

СП ТУРБО ПАСКАЛЬ предоставляет справочную информацию о своих возможностях, отображая ее в любой момент времени в окне подсказки по нажатию клавиши <F1> (окно HELP). Экран HELP позволяет получить многоуровневую справочную информацию посредством выбора од-

ного из ключевых слов, содержащихся в каждом информационном сообщении.

При запуске СП ТУРБО ПАСКАЛЬ на терминале отображается основной экран интегрированной среды и информация о версии системы (нажатием клавиши <Alt—F10> в любой момент времени можно получить эту информацию).

Основной экран интегрированной среды состоит из четырех частей: основное меню; окно редактирования; окно вывода; строка подсказки о допустимых ключах.

В основном меню интегрированной среды возможен выбор из следующих режимов.

Режим FILE предоставляет различные варианты загрузки существующих файлов, создания новых и сохранения файлов. Указанный рабочий файл автоматически загружается в редактор. После завершения работы с файлом пользователь может сохранить его в любом каталоге и с любым именем. Кроме того, в этом режиме пользователь может перейти в любой каталог, высветить на экране имена файлов, находящихся в этом каталоге, временно выйти из интегрированной среды в окружение МДОС 1810.

В режиме EDIT пользователь может с помощью многофункционального встроенного экранного редактора создавать или модифицировать рабочий текстовый файл. Находясь в режиме редактирования, можно перейти в основное меню нажатием клавиши <F10>. Исходный текст при этом остается отображенным на экране, и к нему можно вернуться, нажав <ESC>.

В режиме RVN вызывается компилятор, если редактируемый файл был изменен после последней компиляции. Затем программа выполняется. После завершения программы или при ошибках времени выполнения интерактивная среда переключается на режим редактирования.

Команды режима COMPILE используются для: компиляции программ (COMPILE); построения программ (BUILD); построения программ с проверкой версий (MAKE); установки назначения объектного файла на диск или в память (DESTINATION); поиска ошибки времени выполнения (FIND ERROR); установки основного рабочего файла (PRIMARY); получения информации о текущем исходном файле (GET INFO).

Режим OPTIONS основного меню позволяет изменить установки, определяющие варианты функционирования интегрированной среды.

Команды данного режима включают:  
COMPILED;  
ENVIRONMENT;  
DIRECTORIES;  
PARAMETERS;  
LOAD OPTIONS;  
SAVE OPTIONS.

С помощью команды COMPILED пользователь может указывать различные директивы компилятора, включая проверку диапазона, стека, ошибок ввода-вывода и т. д. Те же директивы могут указываться непосредственно в программе пользователя.

Команда ENVIRONMENT указывает СП ТУРБО ПАСКАЛЬ местонахождение файлов, необходимых для компиляции, компоновки. Кроме того, с ее помощью можно изменять настройку интегрированной среды.



Команда ENVIRONMENT предлагает следующие дополнительные возможности: резервирование исходного файла; автоматическое сокращение исходного файла после редактирования; автоматическое сохранение конфигурации; сохранение содержимого экрана; установка размера табуляции; увеличение размера окон; изменение размера экрана.

Команда DIRECTORIES позволяет помещать СП ТУРБО ПАСКАЛЬ и его рабочие файлы в любой каталог. Эта команда предлагает дополнительное меню для размещений самой системы программирования, файлов EXE, включаемых по директиве INCLUDE файлов, модулей, объективных файлов и PICK-файлов.

Команда PARAMETERS позволяет указать параметры или аргументы командной строки выполняемых программ, как если бы они были набраны в командной строке ОС МДОС 1810.

Команды SAVE OPTIONS и LOAD OPTIONS позволяют сохранять все установки СП ТУРБО ПАСКАЛЬ в файле конфигурации и загружать его.

Автономный компилятор командной строки используется СП ТУРБО ПАСКАЛЬ для работы вне интегрированной среды. Эта версия компилятора идентична версии компилятора интегрированной среды.

При вызове компилятора указывается имя исходного файла с текстом программы и директивы компилятора. Программа компилируется, компонуется с необходимыми модулями и записывается в файл с расширением EXE. Компилятор содержит такой же компоновщик, как и компилятор интегрированной среды.

Директивы автономного компилятора также аналогичны директивам компилятора интегрированной среды, но в отличие от него, устанавливаются в командной строке вызова. Компилятор позволяет указывать список директив, разделенных запятыми. Кроме того, в текст программы можно включать директивы условной компиляции, позволяющие создавать различный код из одного и того же текста.

Действиями компилятора командной строки можно управлять с помощью директив. Директива компилятора представляет собой комментарий со специальным синтаксисом.

Директивы компиляции можно помещать в исходную программу в <⊗ директива>, в командную строку автономного компилятора, используя формат /⊗ <директива>, в файле конфигурации автономного компилятора или с помощью режима OPTIONS/COMPILER в интегрированной среде.

Существует три типа директив: директивы-флаги — эти директивы включают или выключают определенные действия компилятора путем указания после имени директивы знака + или —; директивы с параметром — эти директивы указывают параметры, влияющие на компиляцию, например, имена файлов или размер памяти; директивы условной компиляции — эти директивы используются для управления условной компиляцией частей текста исходной программы, основанной на использовании условных переменных, определенных пользователем.

Во всех директивах, кроме директив-флагов, между именем директив и параметрами должен быть хотя бы один пробел.

Директивы-флаги делятся на локальные и глобальные. Глобальные директивы влияют на всю компиляцию, локальные — только на ту часть программы, которая располагается от этой директивы до следующего появления этой же директивы.

Глобальные директивы помещаются перед разделом описаний программы или модуля. Локальные директивы могут располагаться в любом месте программы или модуля.

В один комментарий можно группировать несколько директив-флагов, разделив их запятыми. В этом случае пробелы между директивами можно не ставить.

Могут использоваться следующие директивы-флаги:

локальная директива В. Эта директива выполняет переключение между двумя различными моделями генерации кода для логических операторов AND и OR. В состоянии В+ компилятор генерирует код для вычисления всего логического выражения. Это означает, что будет вычисляться каждый операнд логического выражения, построенного с помощью операций AND и OR, даже если результат всего выражения уже известен. В состоянии В — компилятор генерирует код для короткого вычисления логического выражения, что означает прекращение вычислений, как только результат всего выражения становится очевидным;

локальная директива D. Эта директива разрешает или запрещает генерацию информации для отладки. Информация содержит для каждой процедуры таблицу, которая отображает адреса объектного кода в соответствующие номера строк исходного файла. При появлении ошибки времени выполнения в программе или модуле СП ТУРБО ПАСКАЛЬ использует эту информацию для указания местоположения оператора исходного текста, который вызвал ошибку;

локальная директива F. С помощью этой директивы выбирается модель вызова для компилируемых процедур и функций. Процедуры и функции, скомпилированные с F+, всегда используют модель вызова FAR. При использовании F — компилятор автоматически выбирает нужную модель: FAR если процедура или функция объявляется в разделе интерфейса модуля или NEAR — в противном случае. Директива F не оказывает никакого влияния на вложенную процедуру. Вложенная процедура всегда имеет модель вызова NEAR;

локальная директива I. С помощью этой директивы разрешается или запрещается автоматическая генерация кода, который проверяет результат вызова процедуры ввода-вывода. Если указана I+ и процедура ввода-вывода возвращает ненулевой код ошибки, программа завершается, выдавая сообщение об ошибке времени выполнения. Если I —, пользователь сам проверяет ошибки ввода-вывода, используя функцию IORESULT;

глобальная директива L. Эта директива разрешает или запрещает буферизацию в памяти при компоновке программ на диск;

глобальная директива N. С помощью этой директивы выполняется переключение между двумя различными моделями генерации кода чисел с плавающей точкой, поддерживаемых СП ТУРБО ПАСКАЛЬ. В состоянии N — генерируется код для выполнения всех вычислений типа REAL путем вызова подпрограмм библиотеки исполняющей систе-



мы. В состоянии N+ для выполнения всех вычислений типа REAL генерируется код, использующий арифметический сопроцессор;

локальная директива R. Эта директива разрешает или запрещает генерацию кода проверки диапазона. В состоянии R+ все массивы и выражения с индексами проверяются на указанные границы, а все присвоения для скалярных и перечисляемых переменных проверяются на пределы диапазона;

локальная директива S. Эта директива разрешает или запрещает генерацию кода проверки переполнения стека;

глобальная директива T. Эта директива разрешает или запрещает генерацию файла с расширением. TRM при компиляции программы на диск. Файл будет содержать карту памяти при компоновке программы;

локальная директива V. Эта директива управляет проверкой типа параметров — переменных при вызове процедуры, если параметры строкового типа.

Могут использоваться следующие директивы с параметрами:

локальная директива I. Эта директива используется для указания включения файла в компиляцию. При наличии этой директивы текст файла включается в исходный текст после использования директивы.

СП ТУРБО ПАСКАЛЬ. позволяет открывать одновременно пять входных файлов. Это означает, что включаемые файлы могут быть вложенными;

локальная директива L. Эта директива указывает компилятору имя объектного файла для компоновки с компилируемой программой или модулем;

глобальная директива M. С помощью этой директивы указываются параметры распределения памяти программы;

локальная директива U. Эта директива указывает имя исходного файла и имя объектного модуля в тех случаях, когда они различны.

Директивы условной компиляции, используя определенные пользователем переменные, позволяют создавать различный код из одного и того же исходного текста.

Используются две конструкции для условной компиляции:

конструкция IFxxx...ENDIF вызывает компиляцию исходного текста, расположенного между IF и ENENDIF только в том случае, если условие, указанное в IF, будет истинным; конструкция IFxxx...ELSE...ENDIF вызывает компиляцию исходного текста между IF и ELSE или между ELSE и ENENDIF, исходя из истинности условия, указанного в IF.

Конструкции условной компиляции могут иметь до 16 уровней вложенности.

Условная компиляция основана на использовании условных переменных, которые могут быть установлены в состоянии TRUE и FALSE с помощью директив DEFINE и UNDEF. Можно также использовать ключ / D в автономном компиляторе или выбор в меню интегрированной среды. Условные переменные следуют тем же правилам, что и идентификаторы в языке Паскаль.

Библиотека стандартных модулей. СП ТУРБО ПАСКАЛЬ поддерживает модульное программиро-

вание и предоставляет средства, позволяющие разбивать большие системы на программные модули и управлять библиотеками модулей. Кроме того, СП ТУРБО ПАСКАЛЬ содержит уже готовый набор стандартных модулей.

Библиотека TURBO.TPL содержит следующие модули: SYSTEM, PRINTER, DOS, CRT, GRAPH.

Модули SYSTEM, PRINTER и DOS предоставляют возможности для работы с файлами, обработки прерываний, системных вызовов, обработки даты и времени.

Модуль CRT предоставляет возможности терминального ввода-вывода текста, управления режимом терминала, разделением экрана на текстовые окна и независимой работы с ними.

Модуль GRAPH необходим, если пользователь работает с графической информацией. С помощью процедур, включенных в модуль, пользователь может определить графические окна, определить систему координат, рисовать точки, линии, окружности, выводить текст с использованием различных шрифтов, работать с битовыми образами.

Программа настройки интегрированной среды. Используя программу настройки интегрированной среды, пользователь может изменить различные первоначальные установки СП ТУРБО ПАСКАЛЬ, такие, как размер экрана, команды редактирования, цвета меню, каталоги по умолчанию.

Программа настройки предоставляет пользователю меню выбора, с помощью которого легко сделать необходимые установки.

Меню позволяет следующее: установить имена каталогов, используемых для включаемых по директиве INCLUDE файлов, модулей, файлов конфигурации, справочных файлов. PTCK — файлов и скомпонованных программ; определить клавиши для команд редактора; установить умолчания для встроенного экранного редактора; установить режим дисплея; изменить цвета, используемые при работе интегрированной среды; изменить размеры окон; установить умолчания для исходного файла.

После выполнения программы настройки создается файл конфигурации интегрированной среды, который используется при работе СП ТУРБО ПАСКАЛЬ.

## Применение

Для функционирования инструментальной СП ТУРБО ПАСКАЛЬ необходима следующая минимальная конфигурация технических средств: процессор CM 1810.2204; оперативная память не менее 256 Кбайт; устройство внешней памяти на гибких дисках типа CM 5639; видеотерминал типа CM 7209; устройство печати типа CM 6329.01.

Объем дополнительной оперативной памяти и внешней памяти на магнитных дисках значительно повышают производительность системы, определяются условиями конкретного применения комплекса инструментальных средств и зависят от объема разрабатываемого ПО.



# Инструментальные средства автоматизации разработки программного обеспечения для СМ 1810 (ИНТЕРПРО 1810)

## Назначение и применение

Комплекс инструментальных средств ИНТЕРПРО 1810 представляет аппарат для контроля за процессом разработки проектов ПО ведения библиотек проектов, корректировки проектов, изготовление программ и программной документации.

Этот аппарат функционально идентичен средствам разработки ПО «Program management tools» фирмы INTEL.

ИНТЕРПРО 1810 предназначен для использования при разработке автоматизированных систем различного назначения, функционирующих на базе СМ 1810 и программно совместимых с ней ЭВМ.

Для функционирования ИНТЕРПРО 1810 на ЭВМ СМ 1810 необходима следующая минимальная конфигурация технических средств: модуль ЦП СМ 1810.2204; модуль системного контроля СМ 1810.2005; видеотерминал типа СМ 7209; устройство печати типа СМ 6329; устройство внешней памяти на ГМД типа СМ 5640; устройство внешней памяти на жестких магнитных дисках типа СМ 5415.

ИНТЕРПРО 1810 реализован на базе универсального программного интерфейса, что обеспечивает функционирование ИНТЕРПРО 1810 в любой операционной среде СМ 1810, где реализована его поддержка (БОС 1810, ДОС 1810).

## Состав и основные функции

Инструментальные средства ИНТЕРПРО 1810 реализованы в виде отдельных компонентов, выполняющих следующие функции: управления библиотекой проекта; генерации командных файлов; макрообработки текста; ввода и редактирования текста; подготовки документации; отладки программ.

В состав ИНТЕРПРО 1810 входят следующие компоненты: программа управления библиотекой проекта SVCS; программа генерации командных файлов MAKE; текстовый макропроцессор MPL; экранный редактор текстов EXTED; пакет программ подготовки документации; символьный отладчик программ PSCDPE.

Средства управления библиотекой проекта — программа SVCS предназначена для автоматизации процесса разработки ПО. Она позволяет: создавать библиотеки проектов разрабатываемого ПО, управлять доступом пользователей к модулям библиотеки проекта; автоматически запоминать информацию об истории изменений каждого программного модуля, включая информацию о том, кто, когда и по какой причине модифицировал модуль; распечатывать историю изменения модуля; группировать различные версии модулей в варианты для построения различных версий ПО и хра-

нить их в одной библиотеке проекта; хранить в библиотеке проекта дополнительную информацию, используемую при разработке ПО (командные файлы для построения системы, спецификации модулей, описание интерфейса и т. д.).

SVCS создает и обслуживает библиотеку проекта (БП), состоящую из так называемых элементов, над которыми могут выполняться различные операции. Элемент подразделен на классы и может содержать до четырех таких подэлементов. Механизм образования вариантов позволяет создавать в БП различные версии элементов. Вариант представляет собой некоторую совокупность (версию) таких элементов, определенных в БП. Единица информации, которая может быть считана или записана в БП, идентифицируется тремя составляющими: именем элемента, классом элемента и именем варианта.

Разработка проекта предполагает манипуляцию различными объектами разрабатываемого ПО. Объекты могут включать в себя исходные файлы и относящиеся к ним объектные файлы, а также скомпонованные программы и их оверлеи. В пределах БП такие объекты рассматриваются как элементы библиотеки. Элемент содержит всю информацию, относящуюся к определенному объекту ПО. Большинство этих элементов относится к программным модулям и содержит исходный текст модуля, соответствующий объектный код, историю изменений модуля и некоторую информацию о структуре проекта, относящуюся к данному модулю.

Каждый элемент БП может содержать информацию четырех классов. Для программного модуля это: исходный текст модуля; объектный код модуля; история изменений (для исходного модуля и относящейся к нему информации); структура (описывающая модуль и его интерфейс).

Средства управления БП определяют некоторые характеристики каждого из этих классов: исходный класс (SO) — этот класс содержит исходный модуль; объектный класс (OJ) — этот класс содержит объектный модуль, генерируемый транслятором из исходного модуля. Загрузочные модули, построенные из объектных модулей компоновщиком, тоже относятся к этому классу, но соответствующие им элементы не будут содержать исходного класса; класс истории изменений (HI) — этот класс содержит информацию о том, кто, когда и зачем внес изменение в исходный модуль. Каждое изменение в исходном модуле автоматически регистрируется; дополнительный класс (CP) — этот класс может использоваться программистом произвольно. Его можно использовать для облегчения документирования системы. Элемент этого класса может быть описанием модуля и его интерфейса, описанием процедуры генерации, командным файлом построения и т. д.

Механизм образования вариантов позволяет программисту создавать различные версии элементов, используемых для построения целевой системы или входящих в нее подсистем. SVCS упрощает процесс разработки различных версий (реализаций) целевой системы, рассматривая совокупность таких элементов, как варианты. Каждый вариант определяется именем и содержащимися в нем элементами. Это позволяет в одной библиотеке проекта накапливать все элементы для различ-



ных вариантов. Один файл истории изменений обслуживает все варианты. Этот метод позволяет программисту в любое время отличить новый вариант от старого.

Имена, присваиваемые вариантам, не имеют predetermined значения, кроме варианта умолчания. Текущая рабочая версия идентифицируется именем WORK. Именованные варианты могут отыскиваться, модифицироваться и возвращаться в БП так же, как и текущая рабочая версия.

Создание вариантов — это не просто дублирование элементов. Если система состоит, например, из 50 модулей и новый вариант создан из варианта WORK с четырьмя модифицированными модулями, то оставшиеся 46 модулей будут относиться к обоим вариантам, при этом ненужного дублирования не происходит.

Поскольку библиотека проекта строится на существующей системе файлов, она разбивается на множество файлов (в зависимости от количества элементов и вариантов). На один элемент БП может приходиться до трех файлов, плюс один файл на вариант.

Библиотеки больших проектов будут содержать соответственно и большее количество файлов, что увеличивает время доступа к элементам БП. По этой причине большой проект следует разбивать на несколько библиотек по логическим составляющим (ядро, система ввода-вывода, надстройка и т. д.). Физических ограничений для количества элементов и вариантов в библиотеке проекта не существует. Однако оптимальным является количество элементов и их вариантов от 50 до 100.

Рекомендуется, чтобы БП занимала отдельный поддиректорий и произвольные файлы не смешивались с файлами БП.

Программа SVCS позволяет выполнять следующие основные функции относительно библиотеки проекта:

ADMIN, используемую для создания библиотеки проекта, включения и удаления элементов и их вариантов;

GET, используемую для чтения информации из библиотеки проекта для последующей модификации. Эта функция используется для информирования SVCS о классе элемента, варианте элемента и пользователе этой информации;

PUT, используемую для записи модифицированной информации в библиотеку проекта. Если PUT относится к исходному классу элемента, автоматически запоминается информация об истории изменений этого элемента, при этом можно ввести текст, идентифицирующий данное изменение;

RETURN, разрешающую доступ к информации, которая была считана функцией GET для модификации, но по какой-то причине не была модифицирована или записана функцией PUT в библиотеку проекта.

Все эти функции выполняются одноименными командами SVCS.

Команды SVCS могут использоваться с дополнительными ключами, позволяющими идентифицировать пользователя информации, устанавливать доступ пользователей по умолчанию к специфицированным вариантам БП, вносить информацию об истории изменений модулей, создавать новые элементы и варианты в БП и т. д.

Средства генерации командных файлов — программа MAKE предназначена для генерации

командных файлов, которые можно использовать для построения необходимой версии разрабатываемого ПО.

Программа MAKE выполняет следующие основные функции: разбор специфицированного пользователем входного файла MAKE, описывающего взаимную связь между модулями, входящими в систему; проверку версий файлов, специфицированных во входном файле MAKE; генерацию командного файла, который содержит только те командные строки, которые необходимы для генерации новой версии ПО.

Программа MAKE начинает обработку входного файла с проверки характеристик модификаций и взаимных связей выбранных вами файлов.

На основе данных зависимости и информации о модификации, хранящейся в библиотеке проекта, программа MAKE автоматически идентифицирует строки задач, выполнение которых необходимо для генерации текущей или требуемой версии программы или программного модуля. Так, если пользователь точно определил, что данный файл используется в пяти из двадцати модулей, участвующих в построении всей программы, то модификации названного файла вызовут впоследствии перекомпиляцию этих пяти модулей. Этот процесс позволяет: избежать перекомпиляции модулей ПО, которые не модифицировались после последней компиляции.

Для подготовки входного файла MAKE используется специальный командный язык, который позволяет: описывать узлы подчиненности, связывающие целевой файл (или список целевых файлов) с файлом (или списком файлов), от которых он зависит; описывать строки задач (вызовы соответствующих программ), которые будут выведены в генерируемый командный файл в случае истинности условия, заданного в узле подчиненности; определять макропеременные, заменяемые строкой заданных знаков, когда такие макропеременные встречаются в командной строке MAKE или строке задачи; использовать команды итерации для генерации набора команд MAKE; использовать команды определения доступа к файлам, хранящимся в БП, которые позволяют извлекать из БП информацию о их модификации, когда имена этих файлов встречаются в списке целевых или подчиненных файлов; использовать специальные макроконструкции для эффективной записи команд MAKE.

Текстовый макропроцессор MPL предназначен для автоматизации процесса разработки проблемно ориентированного ПО. Он позволяет эффективно работать с текстовой информацией, вводить ее в удобной для программиста форме, осуществляя ее преобразование и анализ.

Макропроцессор MPL предоставляет программисту следующие основные возможности: использование встроенных макрофункций; создание новых макрофункций, необходимых пользователю.

Текстовый макропроцессор MPL, обрабатывая исходный текст, реагирует только на вызовы макрофункций и управляет процессом замены вызова макрофункции на текстовое значение, возвращаемое этой макрофункцией. Признаком вызова макрофункции является появление в тексте определенного символа (метасимвола). Для создания пользовательского макро также необходим макровывод. Создает макро встроенная макрофункция



**DEFINE.** Встроенные макрофункции — это предопределенная часть макроязыка, поэтому они могут быть вызваны без предварительного определения.

Макропроцессор **MPL** имеет широкий набор встроенных макрофункций, к числу которых относятся: **SET** — назначает значение числового выражения макроидентификатору и запоминает его в таблице символов макро; **EVAL** — принимает выражение как аргумент и возвращает шестнадцатеричное значение выражения; набор макрофункций, возвращающих логическое значение операции сравнения двух аргументов — строк текста; **LEN** — принимает символьную строку как аргумент и возвращает ее длину в шестнадцатеричном виде; **SUBSTR** — возвращает подстроку ее текстового аргумента; **IF**, **REPEAT**, **WHILE**, **EXIT** — функции управления макрообработкой текста в смысле, аналогичном смыслу операторов, с такими же названиями в процедурных языках; **DEFINE** — функция определения пользовательского макро.

**DEFINE** является наиболее важной встроенной функцией **MPL**, определяющей имя макро, шаблон вызова и тело макро. Шаблон, состоящий из параметров и разделителей, описывает, как должны распознаваться аргументы в макровывозе.

В шаблоне могут использоваться три типа разделителей. Литеральные и идентификаторные разделители появляются непосредственно в шаблоне, а подразумеваемые пробельные разделители — там, где за шаблоном не следует явный разделитель. Литеральные разделители являются самыми обычными, чаще всего они включают в себя запятые, круглые скобки и другие знаки пунктуации. Разделители-идентификаторы — это разделители, которые выглядят и распознаются как идентификаторы. Наличие подразумеваемого пробельного разделителя означает, что предшествующий аргумент завершается первым логическим пробелом.

За распознаванием макроимени (которое уникально идентифицирует макрофункцию) следует согласование текста вызова с шаблоном вызова. Строки в тексте вызова, соответствующие параметрам в шаблоне вызова, становятся значениями этих параметров.

Параметрам, появляющимся в теле макроопределения, предшествует метасимвол. Тело расширяется после вызова, вызов функции заменяет параметры на соответствующие аргументы.

Различные типы разделителей удобны в использовании и способствуют читабельности и гибкости макровывозов и макроопределений.

Редактор текстов **EXTED** является интерактивным экранным редактором текстов с выбором из меню режимов и команд редактирования.

К основным достоинствам редактора **EXTED** относятся: возможность непосредственного редактирования текста, отображаемого на экране дисплея (редактирование в «образе» экрана); выбор из меню режимов и команд редактирования;

(подсказка-меню включает список команд, имеющих в распоряжении оператора. Для выбора необходимой команды достаточно ввести ее первый символ); работа с виртуальным терминалом. Для настройки на терминал, отличный от виртуального, используется файл, загруженный при запуске редактора; возможность работы с шестнадцатеричными кодами символов; возможность задания счетчика повторений, обеспечивающего выполнение

команд редактирования необходимое количество раз; одновременное редактирование двух файлов с возможностью копирования фрагментов текста из одного файла в другой; возможность редактирования в двухоконном режиме. Можно, например, просмотреть ошибки в файле листинга в одном окне и исправить исходный файл в другом; возможность определения и использования макросов — именованной последовательности команд редактора. Вместо ввода ряда команд можно вызвать ранее определенный макрос; наличие встроенного калькулятора; широкие возможности по изменению стандартных режимов выполнения отдельных операций редактором.

Редактор **EXTED** имеет развитый набор команд редактирования, среди которых можно выделить следующие основные группы команд:

команды перемещения курсора, которые позволяют с помощью клавиш управления курсором переместить его в любую точку рабочей зоны экрана, отобразить на экране следующую (предыдущую) страницу текста;

команды удаления / восстановления текста, которые позволяют удалить символ, на который установлен курсор, или слова от него, строку или часть строки справа или слева от курсора, а также восстановить текст, удаленный последней из команд;

команды ввода и замены текста, которые позволяют ввести новый текст или выполнить замену, при которой каждый введенный символ последовательно заменяет соответствующий символ строки исходного текста;

команды поиска и замены последовательности символов, осуществляющие поиск заданной последовательности символов в тексте файла в прямом или обратном направлении от места положения курсора и, при необходимости, замену ее новой последовательностью символов;

команды работы с обменным буфером, позволяющие выделить участок текста в редактируемом файле и скопировать его в обменный буфер с удалением или без удаления из редактируемого файла. Текст из обменного буфера можно скопировать в любое место редактируемого файла либо во внешний файл;

команды макро, позволяющие создать макрос, удалить макрос, сохранить макрос в макрофайле, выдать список имен макросов и т. д.;

команды изменения стандартных режимов работы редактора, позволяющие переопределять значения полей табуляции, вводить заданное количество пробелов вместо знаков табуляции, выводить на экран дисплея символы, шестнадцатеричная величина которых превышает 7FH, просмотр строк, длина которых превышает 80 символов и т. д.

Пакет программ подготовки документации предназначен для автоматизации процесса создания, коррекции и выдачи текстовых документов. Подготовка и коррекция текстовых документов выполняется редактором текстов, хранение текстовых документов осуществляется на магнитных дисках.

Основной компонентом пакета является программа **DOC** — форматор документов, выполняющая следующие функции: форматирование документа; сбор содержания и предметного указателя документа; вывод откорректированного документа на печать (диск).



При большом объеме документа выгодно готовить его по частям, выбирая в качестве отдельно формируемой части документа его раздел или группу разделов. В пакете имеются средства, специально предназначенные для выпуска документа отдельными частями независимо друг от друга. Эти средства позволяют выполнять: формирование отдельных частей документа (программа DOCFOR); сбор содержания документа (программа DOCCNT); сбор предметного указателя документа (программа DOCIND); вывод сформатированных частей документов (программа DOCPRT).

Средства управления выводом документов позволяют осуществить следующие дополнительные варианты вывода: вывод заданного диапазона страниц документа; вывод только четных страниц документа; вывод только нечетных страниц документа.

Средства управления формированием документа включают набор директив — командных строк для форматора документов, определяющих структурные единицы документа, правила оформления фрагментов текста и т. д.

При подготовке документов могут быть использованы следующие основные группы директив форматора:

директивы установки режимов и границ форматирования, позволяющие управлять режимами обработки фрагментов текста, переопределять правые и левые границы, а следовательно, и длину выходных строк структурных единиц документа;

директивы формирования текста и оформления листа — основные директивы форматора, позволяющие: форматировать фрагменты текста с красной строки относительно установленной страницы форматирования; центрировать строки относительно границ; оформлять нумеруемые и ненумеруемые элементы перечисления; копировать фрагменты текста без обработки (литеральный режим); выполнять условный и безусловный переход на новый лист, изменять интервал печати текста и т. д.;

директивы описания структуры документа, облегчающие оформление заголовков, их автоматическую нумерацию и включение в содержание документа;

служебные директивы, позволяющие переключаться на обработку вспомогательных файлов (например, предварительно подготовленных рисунков, таблиц и т. д.).

Для выделения участков текста каким-либо способом (яркой печатью, подчеркиванием, особым шрифтом и т. д.) могут быть использованы управляющие термы. Текст, расположенный между двумя такими термами, будет выделен при печати соответствующим способом. Для каждого печатающего устройства способы выделения текста могут различаться. Конкретные способы выделения на устройстве определяются по настроечному файлу, в котором для каждого терма задается соответствующий параметр со значением управляющего кода, реализующего заданный способ выделения текста на данном устройстве.

Символический отладчик программ PSCOPE предназначен для автоматизации процесса разра-

ботки ПО. PSCOPE представляет собой интерактивный символический отладчик для объектных модулей, полученных компилятором PASCAL-86 (версия 1.0 и выше), компилятором PL/M-86 (версия 2.0 и выше) и компилятором FORTRAN-86 (версия 1.1 и выше). Отладчик позволяет проверять и модифицировать выполнение программ для обнаружения логических ошибок.

Программа PSCOPE реализует следующие основные функции: два способа пошагового выполнения программы; возможность символической проверки и модификации всех типов данных PASCAL-86 и PL/M-86. Виртуальная таблица символов хранит информацию о символах и номерах строк; возможность отслеживания за выполнением отлаживаемой программы, используя точки прерывания и точки трассировки; использования условных команд и управляющих блоков PSCOPE для построения блоков команд отладчика; определение различных типов объектов отладки, включая LITERALLY (форма макрокоманды) и переменных отладки, и манипулирование ими. PSCOPE позволяет создавать и редактировать процедуры отладки и символы отладки; добавление и удаление кода на уровне инструкций без изменения исходного текста программы. Заплатки используются только в том сеансе отладки, в котором они генерируются; редактирование команд, процедур отладки и заплаток с помощью встроенного экранно-ориентированного редактора PSCOPE.

Кроме того, вспомогательные команды PSCOPE: предоставляют вспомогательную информацию (команда HELP); позволяют сохранить часть или весь отладочный сеанс в регистрационный файл; позволяют выполнить командный файл.

Как PSCOPE, так и программа, подлежащая отладке, размещается в оперативной памяти емкостью до одного мегабайта.

Отладчик содержит встроенный редактор для создания и модификации объектов отладки. Он позволяет создавать длинные командные строки и редактировать последнюю командную строку.

Встроенный редактор экранно-ориентированный является усеченным вариантом редактора EXTED.

Редактор использует информацию в файле PSCOPE.CRT (аналогичном настроечному файлу редактора EXTED) для определения последовательностей управления курсором, если они не такие же, как у виртуального терминала. Кроме того, файл PSCOPE.CRT позволяет перепрограммировать специальные клавиши, такие, как стрелки направлений, ESC и HOME для передачи этих функций другим клавишам.

PSCOPE позволяет определять процедуры отладки для расширения языка команд отладчика и средств разработки отлаживаемой программы. Процедуры отладки являются одной из наиболее мощных возможностей.

Процедуры отладки (тип PROC) могут использоваться для автоматизации процесса тестирования ПО, установки точек прерывания на основе значений данных или булевских условий. Процедуры отладки позволяют в отличие от LITERALLY использовать параметры.



# СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ (СУБД)

## Базовая система для создания и ведения в реальном масштабе времени локальных баз данных реляционного типа на СМ ЭВМ (СУБД БАРС)

СУБД БАРС предназначена для поддержания реляционной модели данных в автоматизированных системах управления различного назначения, построенных на основе технических средств СМ ЭВМ. Система обеспечивает оперативный сбор и интеграцию информации в базах данных с возможностью доступа к ней с помощью интерактивного языка в многотерминальном режиме, а также из оперативных программ в реальном времени.

Функционирование системы БАРС поддерживается в сети мониторами РАФОС и РАФОС-II, за исключением РМ — монитора РАФОС-II. Для работы пользователей с нескольких терминалов необходим ТS — монитор РАФОС-II. Для доступа к данным из программ пользователя, составленных на ФОРТРАНЕ СМ и ПАСКАЛЕ, необходим ХМ — монитор РАФОС-II. Система БАРС не вносит ограничений на режимы функционирования РАФОС и РАФОС-II.

БАРС является многобазовой системой, т. е. в ней параллельно и независимо может осуществляться взаимодействие пользователей с несколькими локальными базами данных (ЛБД). В соответствии с реляционной моделью данные внутри ЛБД логически представлены в виде двумерных таблиц-отношений. Такое представление просто, естественно и особенно удобно для пользователя — непрограммиста, обеспечивает высокую степень независимости программ от данных.

Данные в отношениях физически хранятся покортежно (построчно), в один кортеж могут входить данные разных типов. Возможны целые, вещественные и символьные типы данных. Предусмотрена работа и со строками символов различной длины.

Доступ к данным в БАРС осуществляется в диалоговом режиме с помощью специального процедурного языка манипулирования данными ЯНОТ, с помощью специальных команд БАРС, а также из программ пользователя, выполненных на Макроассемблере, Фортране и Паскале.

Логическая организация данных. Каждая ЛБД, управляемая БАРС, содержит отношения двух типов: отношения общего пользования и отношения пользователя. Отношения первого типа создаются администратором ЛБД и доступ к ним разрешен всем пользователям БД. Отношения второго типа доступны только тем пользователям ЛБД, для которых они созданы.

Метаданные ЛБД сведены в каталоги, в целях унификации доступа также представлены в виде отношений.

Доступ к ЛБД осуществляется через системную БД SYSBAS, находящуюся в ведении администратора БАРС. Ядром SYSBAS служит каталог DIRECTBASE, содержащий информацию о всех ЛБД. Кроме того, администратор БАРС имеет возможность создавать и вести в SYSBAS информационные отношения, содержащие информацию, например, статистическую, о функционировании системы в целом. Логическая структура информации БАРС представлена на рис. 1.

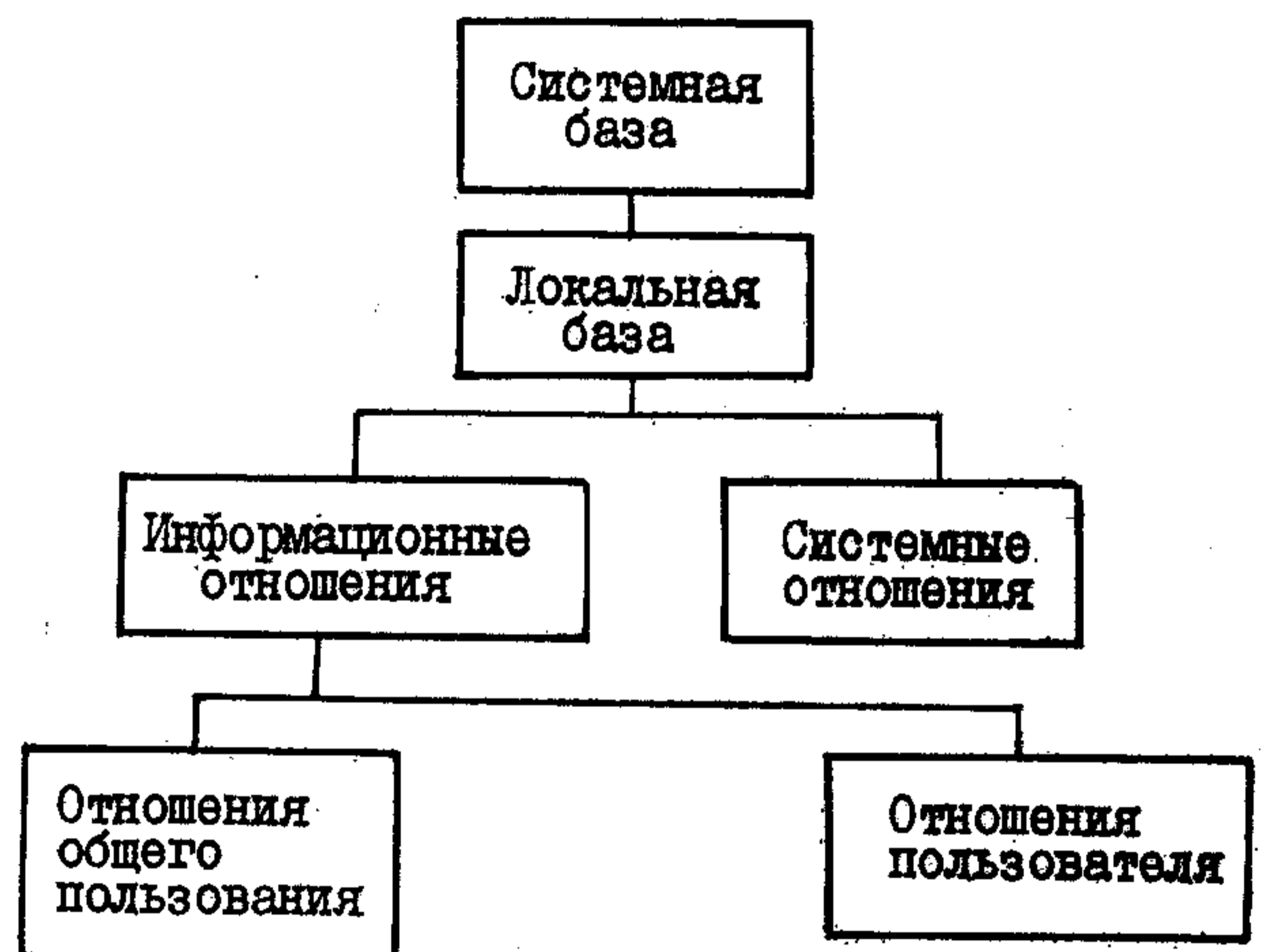


Рис. 1. Логическая структура информации БАРС

Доступ к отношениям ЛБД осуществляется через системные отношения (каталоги) RELATION, ATTRIBUTE, USER, INTEGRITY.

Каталог RELATION содержит параметры всех отношений ЛБД, в том числе и системных. Каждому отношению ЛБД соответствует один кортеж RELATION, последний атрибут которого позволяет отличать системные отношения от информационных.

Сведения о всех атрибутах всех отношений ЛБД сведены в системное отношение ATTRIBUTE. Каждому атрибуту ЛБД соответствует один кортеж этого отношения, содержащий информацию о принадлежности атрибута к отношению, местонахождении в кортеже этого отношения, типе данных атрибута и потере атрибута в ключе (для отношений с ключевыми способами хранения данных).

Пользователи ЛБД регистрируются в каталоговом отношении USER, кортежи которого содержат имя и пароль пользователя.

Ограничение целостности, наложенные на отношения ЛБД, хранятся в кортежах каталогового отношения INTEGRITY, имеющего два атрибута — номер отношения и предложение ограничения целостности, сформулированные на языке ЯНОТ.

Организация хранения данных. Выбор физической организации данных в БАРС основан на удовлетворении следующих требований, предъявляемых к СУБД: гибкость, заключающаяся в отсутствии необходимости реорганизации памяти в процессе добавления, модификации и исключения данных; эффективность доступа, предполагающая



возможность совмещения последовательного просмотра с прямым доступом по ключу с минимальными затратами; минимизация времени поиска, достигается за счет минимизации числа обращений к внешней памяти; эффективность использования памяти, предполагающая оптимальное соотношение между объемом выделенной памяти и объемом фактически хранимых данных.

Информационные компоненты БАРС включают СБД SYSBAS и ЛБД (см. рис. 1).

С точки зрения РАФОС (РАФОС-II) каждая из информационных компонент БАРС представляет собой отдельный файл либо совокупность таких файлов на разных дисках, представляющих том ЛБД. ЛБД может иметь до семи томов.

Элементы данных в БАРС могут быть следующих типов: единичный символ в коде КОИ-7, занимающий 1 байт памяти; целое число (2 байт); вещественное число (4 байт); строка из N символов КОИ-7 (N байт).

Элементы данных размещаются в составе кортежей отношения.

Память для размещения отношений БАРС имеет страничную организацию. Данные в отношениях размещаются покортежно, кортеж всегда размещается в пределах одной страницы. Размер страницы кратен размеру физического блока (512 байт), зависит от размеров кортежей отношения и задается при создании схемы отношений. Максимальный возможный размер страницы — четыре физических блока.

В БАРС реализованы три структуры хранения данных, в дальнейшем называемые схемами сохранения данных: неключевая схема сохранения данных — «куча»; ключевые схемы сохранения данных — «контрольная» и «обобщенный справочник».

Под отношения с ключевой схемой сохранения, помимо основных страниц, отводятся страницы переполнения, куда направляются кортежи, для которых нет места в основных страницах.

При неключевой схеме сохранения данных «куча» кортежи помещаются в отношение независимо от конкретных значений атрибутов последовательно в порядке поступления (новые кортежи, добавляемые в отношение, просто присоединяются к существующим). «Куча» требует минимальных затрат памяти и времени при записи кортежей в отношение и является рациональной в тех случаях, когда необходимо полностью просканировать отношение.

Неключевую схему сохранения рекомендуется использовать для отношений малого объекта либо для сбора данных.

В «контрольной» схеме сохранения данных адреса хранения кортежей в пределах отношения представляются в виде значений некоторой функции, у которой аргументом являются значения ключевых атрибутов кортежей. Эта схема обеспечивает самый быстрый доступ к отдельным участкам отношения, если корректно и полно заданы значения набора ключевых атрибутов отношения. По мере заполнения страниц поступающими кортежами может возникнуть необходимость в использовании области страниц переполнения.

«Контрольная» схема сохранения данных предоставляет максимум выгод для ситуаций, при которых доступ к данным обуславливается уникальным ключевым значением.

Использование «контрольной» схемы сохранения обеспечивает доступ к отдельному кортежу за минимальное количество обращений к внешней памяти. Основным недостатком «контрольной» схемы сохранения является несоответствие (в общем случае) получаемой последовательности адресов в памяти лексикографической упорядоченности ключей.

В документацию системы включены указания пользователю по программированию и включению в ПО «контрольных» функций пользователя.

Схема сохранения по методу «обобщенного справочника» является одной из индексно-последовательных схем, в которой значения ключевых атрибутов (в отличие от «контрольной» схемы) определяют не точный адрес местонахождения кортежа, а группу страниц отношения, которую уже необходимо просканировать. Ключевых атрибутов может быть несколько (до пяти), значения каждого из них (в заданной последовательности) все более точно определяют интервал страниц, в который должен попасть (или из которого должен выбираться) данный кортеж.

В случае отсутствия места в интервале кортеж направляется в область страниц переполнения.

Определение интервала страниц отношения осуществляется с помощью многоуровневого справочника, представляющего собой на каждом уровне таблицу соответствия диапазонов значений ключевого атрибута диапазонам страниц отношения. Каждому ключевому атрибуту соответствует свой уровень, определяемый местом атрибута в составном ключе. Справочник формируется при создании схемы отношения в ЛБД.

В случае рациональной организации «обобщенного справочника» доступ к данным через него практически не уступает по производительности прямому доступу.

**Манипулирование данными.** Средством формулирования запросов пользователя в БАРС служит язык манипулирования данными ЯНОТ. ЯНОТ является непроцедурным, реляционно полным языком реляционного исчисления с переменными — кортежами.

Язык ЯНОТ системы БАРС включает те же возможности, что и язык QUEL системы INGRES, но наряду с английской имеет и русскую нотацию.

Запросы в БАРС могут подаваться по одному или блоками (с терминала подаются только отдельные запросы). Друг от друга запросы в блоке отделяются специальными разделителями.

Запросы в ЯНОТ могут быть следующих типов: запросы на манипулирование данными; запросы на упорядочение данных; запросы на наложение ограничений целостности.

В общем случае запрос включает в себя блок описаний, блок инструкций и блок условий.

В блоке описаний описываются кортежные переменные, одна или несколько. В описании кортежной переменной указывается ее имя и имя отношения, кортежи которого составляют область определения переменной. В одном запросе могут участвовать до пяти переменных по одному или различным отношениям.

В блоке инструкций указывается один из операторов манипулирования данными, а также может присутствовать оператор упорядочения корте-



жей отношения по убыванию или возрастанию значений заданного атрибута.

Блок условий запроса предназначен для описания критериев выборки, которым должны соответствовать данные целевого списка операторов блока инструкций. Условия обычно накладываются на значения атрибутов с помощью знаков сравнения. Отсутствие условия (блока условий) означает, что условия выполнены всегда.

Таким образом, операторы языка ЯНОТ можно разделить на следующие группы по типам запросов: оператор определения переменной RANGE (русская нотация ПЕРЕМ); операторы манипулирования данными RETR (НАЙДИ), APP (ДОБАВЬ), REPL (ЗАМЕНИ), DEL (УДАЛИ); операторы упорядочения кортежей в отношениях UP (СОРТВ) и DOWN (СОРТУ); оператор наложения ограничений целостности INT (ЦЕЛ); оператор условия WHERE (ЕСЛИ); агрегатные операторы языка COUNT (КОЛ), MAX (МАКС), MIN (МИН), SUM (СУММА), AVG (СРЕДН).

Оператор RANGE открывает блок описания переменных.

Для операторов манипулирования данными указывается имя целевого отношения и целевой список атрибутов.

Оператор RETR является оператором выборки. Цель запроса с таким оператором — создание нового отношения с именем целевого отношения. В целевом списке оператора указываются имена атрибутов создаваемого отношения и способы образования кортежей нового отношения из кортежей исходных отношений. Если в качестве имени нового отношения указаны TT: или LP:, то результат выводится на терминал или устройство печати пользователя соответственно.

Цель запроса с оператором APP — добавление новых кортежей к уже существующему отношению. Не указанные в целевом списке атрибуты кортежа заполняются значениями неопределенности.

Оператор REPL служит для коррекции значений атрибутов в кортежах существующего отношения. В целевом списке оператора указывается способ получения новых значений атрибутов тех кортежей отношения, которые удовлетворяют блоку условий запроса.

Оператор DEL позволяет исключить из указанного отношения те кортежи, на которых блок условий принимает значение ИСТИНА. В этом операторе отсутствует целевой список.

Для упорядочения кортежей отношения по значениям заданного атрибута в ЯНОТ используются операторы UP — упорядочение по возрастанию и DOWN — упорядочение по убыванию.

В запросах на упорядочение отсутствует блок условий. При выполнении операции упорядочения из отношений удаляются дубликаты кортежей.

На корректировку значений атрибутов отношения могут быть наложены ограничения целостности с помощью оператора INT. Ограничения описываются одним из знаков сравнения  $>$ ,  $<$ ,  $=$ ,  $>=$ ,  $<=$ .

Оператор WHERE открывает блок условий запроса.

К совокупности значений атрибута отношения могут быть применены агрегатные операторы ЯНОТ. Это операторы COUNT — подсчет числа значений атрибута в отношении, MAX — выбор

максимального значения, MIN — выбор минимального значения, SUM — вычисление арифметической суммы значений и AVG — подсчет среднего арифметического значения атрибута.

**Средства создания и ведения баз данных.** СУБД БАРС предоставляет администратору и пользователю БД широкий набор сервисных средств для ее ведения. Реализация конкретной функции сервиса системы осуществляется с помощью команды терминального монитора БАРС. Команда задается в виде двухсимвольного идентификатора.

Набор команд БАРС позволяет инициировать следующие операции: создание схем ЛБД и отношений; удаление ЛБД и отношений из ЛБД; восстановление удаленных отношений; переименование отношений; визуализация каталоговой информации; вывод содержимого отношений на терминал и устройство печати; «уплотнение» информации в ЛБД и отношениях; копирование отношений ЛБД в файлы на внешних носителях и обратно; модификация схем сохранения отношений; ведение санкционирования доступа к ЛБД; наложение и снятие ограничений целостности; загрузка отношений ЛБД заранее подготовленной информацией из файла на внешнем носителе; формирование данных для программ печати отчетов; выполнение запроса в интерактивном режиме; инициация обмена БАРС с программой пользователя; распечатка справки о составе команд терминального монитора; завершение работы БАРС (выход).

Возможность выполнения конкретной команды пользователя зависит от его права доступа к необходимой информации ЛБД.

Параметры выбранной команды уточняются в ходе диалога.

СУБД БАРС предоставляет возможность доступа к ЛБД из программ пользователя, составленных на языках Макроассемблер, Фортран СМ, Паскаль.

Для обеспечения доступа в программном режиме в состав системы БАРС включены предтрансляторы для каждого языка программирования.

Текст программы на исходном языке со встроенными запросами к ЛБД на языке ЯНОТ предварительно обрабатывается соответствующим исходному языку предтранслятором.

Результатом работы предтранслятора является программа на исходном языке, содержащая макрокоманды обмена данными с программами БАРС.

После компиляции, компоновки и загрузки программы пользователя (средствами соответствующей системы программирования РАФОС (РАФОС-II) вызывается БАРС и с помощью команды терминального монитора инициируется обмен данными с программами. После чего БАРС переходит в режим удовлетворения запросов из задач пользователя.

**Программное обеспечение базовой системы** для создания и ведения в реальном масштабе времени ЛБД реляционного типа включает в себя следующие компоненты: программа генерации однотерминальной версии системы; программа генерации многотерминальной версии системы; программа создания и ведения БД; предтрансляторы с исходных языков программирования; генератор системной базы; генератор программ печати отчетов.

Программы генерации системы предназначены для создания версий программы ведения БД, ори-



ентированных на конкретное применение.

Генератор системной базы необходим для создания системной базы, используемой при работе программы создания и ведения БД.

Предтрансляторы осуществляют обработку исходных текстов программ пользователя на Макроассемблере, Фортране СМ и Паскале с внедренными в исходный текст запросами манипулирования данными на языке ЯНОТ.

Генератор программ печати отчетов предназначен для автоматизированного получения выходных форм. Он представляет собой предпроцессор транслятора системы программирования Паскаль. На основании текста на входном языке генерируется Паскаль — программа печати необходимого отчета. Входной язык генератора позволяет описать формат отчета и осуществить привязку информации из базы данных к отчету. Основные возможности, предоставляемые входным языком, следующие: задание формата листа выходного документа; описание заголовков листов и граф таблиц; нумерация листов; оформление заглавных и итоговых строк.

Пользователю предоставляется возможность разграфки отчета. Пографное описание отчета позволяет легко модернизировать выходной документ: удалить графу, изменить порядок граф, расширить или сузить графы. Для печатаемых данных предусмотрена возможность их выравнивания по правому краю графы.

Сгенерированные программы печати отчетов являются системно независимыми к БАРС и работают с входной информацией, скопированной из ЛБД в файл РАФОС (РАФОС-II) с помощью сервисных средств БАРС. Этот файл может быть подготовлен пользователем и другими средствами.

## Применение

Для функционирования БАРС необходима следующая минимальная конфигурация технических средств: процессор СМ 2103 или СМ 2104; оперативная память объемом 28 Кслов; видеотерминал типа СМ 7205; устройство внешней памяти на магнитных дисках типа СМ 5400.

Объем дополнительной оперативной памяти, внешней памяти на магнитных дисках и количество дополнительных терминалов (до 7) определяется условиями конкретного применения системы и зависит от количества и объемов ЛБД и количества одновременно работающих пользователей.

СУБД БАРС может функционировать также на всех комплексах микро- и мини-ЭВМ, программно-совместимых с СМ-3, 4 («Электроника-60», «МЕРА-60», «Электроника-100/25»), достаточных для функционирования РАФОС и(или) РАФОС-II и имеющих устройство внешней памяти на магнитных дисках.

Возможности системы БАРС позволяют эффективно ее использовать в автоматизированных системах управления технологическими процессами, научным экспериментом, а также в различных информационно-поисковых системах и в системах деловых применений.

## ИНТЕРЕАЛ — мобильная СУБД с интерфейсом реляционного типа

СУБД с интерфейсом реляционного типа (ИНТЕРЕАЛ) является мобильной СУБД и предназначена для поддержания реляционной модели данных АСУ различного назначения. В настоящее время разработана и эксплуатируется пользователями СУБД ИНТЕРЕАЛ для работы на ЭВМ типа СМ-4, СМ 1420, «Электроника 100/25» (в ОС РВ, RSX), СМ 1810 (БОС 1810), а также разрабатывается для ЭВМ СМ 1425 и СМ 1820. Система ИНТЕРЕАЛ является оригинальной отечественной СУБД, отвечающей современным требованиям СУБД реляционного типа согласно концепции СМ ЭВМ четвертого поколения.

ИНТЕРЕАЛ является многопользовательской системой, в ней параллельно и независимо может осуществляться взаимодействие нескольких пользователей с БД. В соответствии с реляционной моделью данные внутри БД логически представлены в виде двумерных таблиц-отношений (до 32768). Такое представление просто, естественно и особенно удобно для пользователя-непрограммиста, обеспечивает высокую степень независимости программ от данных.

Система обеспечивает ведение БД на устройствах с прямым доступом, поддерживаемых ОС, и не накладывает никаких ограничений на ее функционирование.

Данные в отношении физически хранятся покортежно (построчно). В каждом отношении может быть до 65535 кортежей (строк). В один кортеж могут входить данные разных типов (символы, целые и вещественные числа, а также строки символов различной длины). Каждое отношение (таблица) может содержать до 255 атрибутов (столбцов).

В ИНТЕРЕАЛ реализован аппарат создания и ведения индексных списков на указанные (объявленные) ключевые атрибуты.

С помощью непроцедурного языка манипулирования данными ИНТТЕРМ, а также других языковых и программных средств ИНТЕРЕАЛ пользователь может: создавать отношения; уничтожать отношения; помещать и удалять данные из отношений; производить операции поиска и выборки данных по различным критериям; осуществлять теоретико-множественные операции над отношениями одинаковой структуры (объединение, пересечение и т. п.); выгружать отношения в файлы операционной системы; загружать отношения заранее подготовленной информацией; уплотнять отношения путем перезаписи информации на освободившиеся в процессе ведения базы данных места.

Помимо упомянутого выше непроцедурного языка манипулирования данными ИНТТЕРМ в ИНТЕРЕАЛ реализованы еще три языковых средства: табличный знак манипулирования данными ИНТТАБ, позволяющий осуществлять операции поиска, добавления, замены и удаления данных при работе с одним отношением в экранном режиме; алгоритмический язык прикладного программиста ИНТКОМ, предоставляющий средства описания процедур интерактивного общения и обработки информации. ИНТКОМ обеспечивает ИН-



ТЕРЕАЛ широкий круг применения в непромышленной сфере — системах деловых применений, информационно-поисковых системах и т. п.; алгоритмический язык генерации отчетов ИНТРПТ, средства которого позволяют воспользоваться данными базы для подготовки отчетных форм любой структуры.

В системе ИНТЕРЕАЛ реализованы унифицированные средства доступа к данным с терминалов и из программ пользователей. Основой этих средств является CALL-интерфейс ИНТЕРЕАЛ, которым можно воспользоваться в программах на любых языках, реализованных в операционных системах и имеющих CALL-вызов со стандартной передачей параметров.

Функционирование ИНТЕРЕАЛ в высокореактивных операционных средах ОС РВ, RSX, БОС 1810 позволяет использовать ИНТЕРЕАЛ для оперативного сбора и интеграции информации в автоматизированных системах.

Для функционирования ИНТЕРЕАЛ на ЭВМ СМ 1810 необходима следующая минимальная конфигурация технических средств: процессор СМ 1810.2204; сопроцессор арифметики с плавающей точкой; оперативная память объемом не менее 512 Кслов; видеотерминал типа СМ 7209; устройство внешней памяти на магнитных дисках типа СМ 5639; устройство внешней памяти на магнитных дисках типа СМ 5514; устройство печати типа СМ 6329.01.

Для функционирования ИНТЕРЕАЛ на ЭВМ СМ 1420 необходима следующая минимальная конфигурация технических средств: процессор СМ 2402; оперативная память объемом не менее 128 Кслов; видеотерминал типа СМ 7202; устройство внешней памяти на магнитных дисках типа СМ 5410; устройство печати типа СМ 6305.01.

Объем дополнительной оперативной памяти, внешней памяти на устройствах с прямым доступом и количество дополнительных терминалов значительно повышают производительность вычислительного комплекса, определяются условиями конкретного применения и зависят от объема БД и числа одновременно работающих пользователей.

Система ИНТЕРЕАЛ оперирует с информацией, собранной в таблицы (отношения), что наиболее понятно и близко любому из пользователей.

Доступ к информации в БД осуществляется через системные каталоги, выполненные в виде отношений (системные отношения). Распоряжаются этими каталогами только программы системы ИНТЕРЕАЛ. Кроме того, он создает рабочие файлы для временного хранения промежуточной информации при обработке многопеременных запросов на манипулирование данными.

Основные функции ИНТЕРЕАЛ обеспечивают описание данных, обработку запросов на манипулирование данными и поддержку исполняющих систем языка генерации отчетов и языка прикладного программиста. К этим функциям относятся: создание схемы БД; создание схемы отношения; выборка, обновление, включение и удаление данных; исполнение программ на языке генерации отчетов; исполнение программ на языке ИНТКОМ.

Функции ИНТЕРЕАЛ реализуют отдельные программы системы. Схема БД создается пользователем в диалоговом режиме. Пользователь, создавший эту схему, становится администратором этой БД.

Манипулирование данными может осуществляться как в интерактивном режиме через посредство исполняющих систем языков ИНТТЕРМ и ИНТТАБ, так и из программ пользователя на языках Ассемблер, Фортран, Паскаль и т. п.

Исполнение программ на языках ИНТРПТ и ИНТКОМ производится в два прохода: трансляция в промежуточный код и интерпретация этого кода. Оба языка имеют средства подачи запросов к БД на манипулирование данными.

Наличие языка прикладного программиста позволяет существенно облегчить процесс создания прикладных систем пользователя, а возможность с его помощью конструировать и выполнять разнообразные меню позволяет воспользоваться достоинствами других языковых средств системы.

Программы, входящие в ИНТЕРЕАЛ, не только обеспечивают описание данных и манипулирование ими, но и имеют сервисные средства, которые расширяют эксплуатационные возможности системы, в их числе: загрузку (выгрузку) отношений; загрузку (выгрузку) описаний отношений и атрибутов; модификацию организации данных в отношении; сжатие данных в отношении; вывод содержимого отношения на устройство печати или видеотерминал; выдачу разнообразной справочной информации.

Для автоматизированного получения выходных форм в составе ПО ИНТЕРЕАЛ реализован генератор отчетов (транслятор+интерпретатор с языка ИНТРПТ).

Подсистема справок позволяет получить информацию о текущем состоянии БД и ее компонентах, о параметрах запуска ядра системы и очередей, которые могут быть перенастроены (в уже сгенерированной системе) в соответствии с требованиями конкретного приложения пользователя.

Средства администратора системы делают доступным процесс трассировки и управления состоянием активных каналов и очередей системы.

Основные структурные компоненты ИНТЕРЕАЛ и их взаимосвязь: приведены в структурной схеме ИНТЕРЕАЛ (рис. 2).

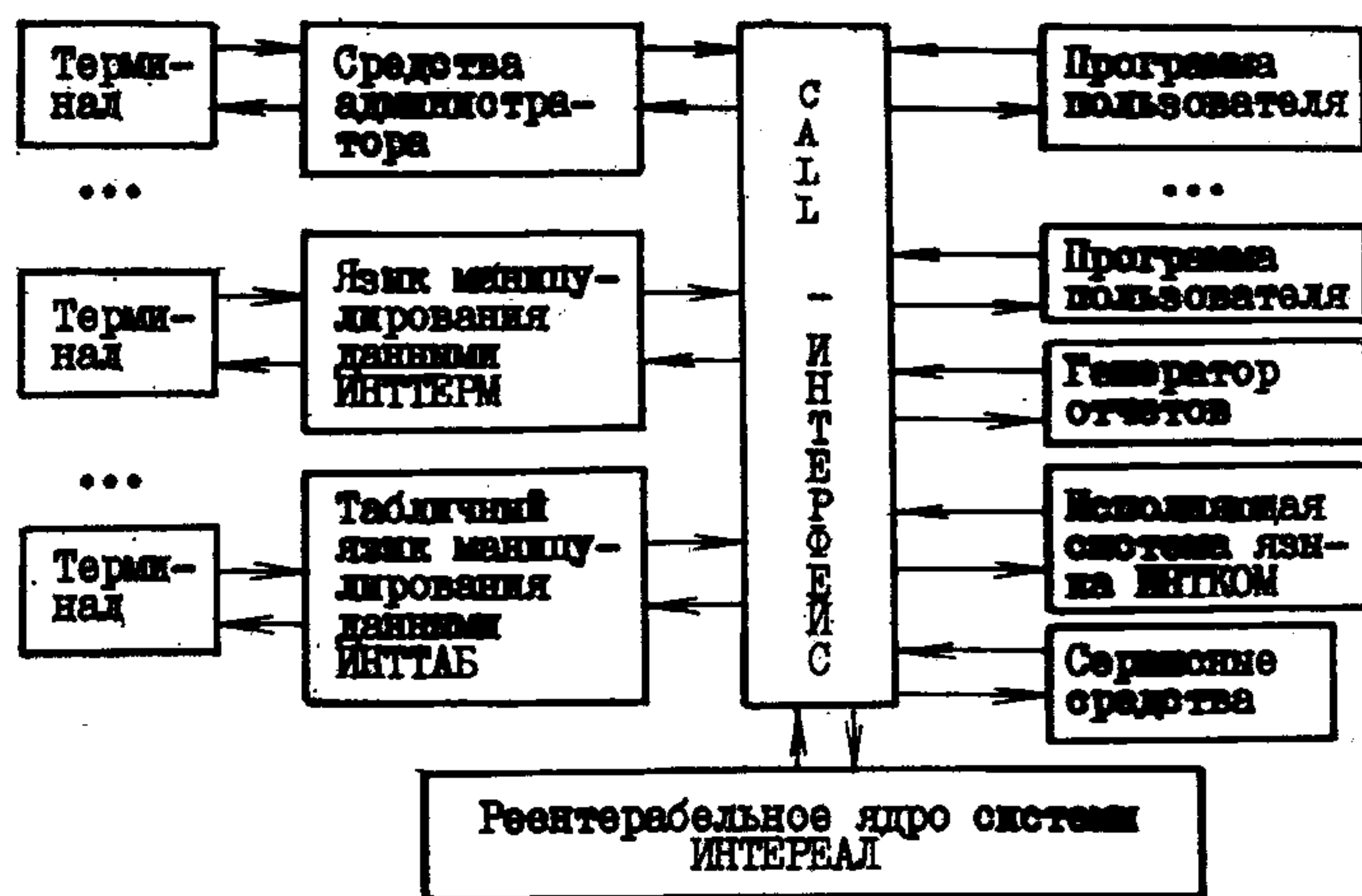


Рис. 2. Структурная схема ИНТЕРЕАЛ

Из схемы видно, что ИНТЕРЕАЛ — это комплекс задач, работающих под управлением ОС.

Обращения к данным базы от программ пользователя и компонент ИНТЕРЕАЛ идут через ядро



системы. Ядро ИНТЕРЕАЛ — это реентерабельная программа, осуществляющая диспетчеризацию и обработку запросов. Общение с ядром производится только через CALL-интерфейс ИНТЕРЕАЛ.

При получении запроса ядро производит анализ ресурсов системы. Если состояние ресурсов позволяет принять этот запрос на выполнение, то в зависимости от типа поступившего запроса его обработка организуется следующим образом: если принят запрос на манипулирование данными, то вызывается соответствующий модуль ядра; если принят запрос на манипулирование данными, то вызываются на выполнение средства трансляции, осуществляющие синтаксический и семантический анализы запроса. Далее при удачной трансляции вызывается на выполнение модуль удовлетворения запроса.

При этом каждый запрос выполняется по определенному каналу. Обработка запроса не является непрерывным процессом. Единица непрерывного выполнения называется квантом обработки запроса. В промежутке между квантами средства диспетчеризации ресурсов имеют возможность перехватить инициативу, проанализировать ситуацию и перераспределить ресурсы системы между каналами.

Развитие системы ИНТЕРЕАЛ предусматривает последовательное создание новых версий системы для перспективных моделей мини- и микроЭВМ с параллельным переносом разработанных новых функций и сервисного окружения дополнительно к уже созданным версиям ИНТЕРЕАЛ.

Намечено проведение работ по реализации языковых средств для создания систем деловых применений, средств обработки крупноформатных электронных таблиц и средств сетевой поддержки, а также по расширению библиотеки методов доступа за счет реализации новых эффективных методов доступа, созданию более совершенной HELP-системы и реализации подсистемы восстановления БД с помощью прямого доступа к отношениям.

## Система управления реляционными базами данных для СМ 1810 (СМ 1820) (ДАТАРЕАЛ)

### Назначение

СУБД ДАТАРЕАЛ предназначена для организации, управления и использования деловой, научной и текстовой информации при разработке автоматизированных информационных систем различного назначения, функционирующих на базе технических средств ЭВМ СМ 1810 (СМ 1820) в среде малой дисковой операционной системы МДОС 1810 (МДОС 1820).

При достаточной степени мощности, гибкости и простоты использования ДАТАРЕАЛ позволяет создавать и поддерживать системы управления данными широкой ориентации и различных типов.

Степень освоения СУБД соответствует уровню сложности поставленной задачи — простые задачи решаются с использованием несложных средств СУБД, а более сложные требуют изучения и использования дополнительных возможностей программной системы. Для пользователей логическая структура организации БД представляется в виде двумерных динамических таблиц, создание и корректировка которых осуществляется путем выбора соответствующей позиции из выводимого на экран терминала меню. Такое представление и взаимодействие просто, естественно, несложно для обучения и особенно удобно для пользователей — непрограммистов, обеспечивает высокую степень независимости программ от данных. Пользователь имеет возможность менять структуру данных уже после того, как она создана. Система ДАТАРЕАЛ позволяет сравнительно просто добавлять или удалять отдельные типы данных, а также объединять, разделять или изменять целые файлы.

При использовании ДАТАРЕАЛ имеется возможность разрабатывать ход выполнения приложений в мельчайших деталях, создавать свои меню, задавать формы расположения информации на экране дисплея и распечатываемых листингах отчетов. Такая возможность особенно существенна, если приложение предназначено для использования его другими пользователями. С помощью СУБД ДАТАРЕАЛ можно создавать систему диалогов, которая дает пользователю возможность ориентироваться в самой сложной структуре данных.

Концепция мультифайловой обработки открывает пользователю системы ДАТАРЕАЛ возможности организации одновременной обработки файлов и формирования запросов сложной структуры.

Система ДАТАРЕАЛ обеспечивает ведение БД на ГМД и дисках типа «Винчестер».

Особенно привлекательной характеристикой ДАТАРЕАЛ для решения задач обработки данных и экономических применений является возможность создавать обобщающие отчеты по файлам БД и выдавать частичные суммы по ходу отчета.

В ДАТАРЕАЛ применяются два способа упорядочивания файлов — сортировка и индексирование. Сортировка в ДАТАРЕАЛ выполняется не в исходном файле. При сортировке файла сначала записи упорядочиваются в соответствии со значениями какого-либо конкретного поля, а затем копируются в этой новой последовательности. Для того чтобы защитить исходный файл, упорядоченные записи копируются в новый файл: это предохраняет исходный файл от повреждений, которые могут возникнуть, например, при выключении питания во время процесса сортировки.

В СУБД ДАТАРЕАЛ применяется процессор командных файлов, который выполняет командные файлы, объединяющие сотни операторов командного языка. Для более быстрого построения прикладных программ можно воспользоваться многоуровневой технологией разработки командных и процедурных файлов, позволяющей проектировать и разрабатывать приложения на основе принципов модульного программирования.

Богатые средства форматирования ввода и вывода предоставляют удобные способы ввода, вывода, накопления и обработки данных с использованием нескольких режимов отображения информации на экране дисплея.



Основными функциями ДАТАРЕАЛ являются: создание законченных БД; ввод, добавление, удаление, редактирование и вывод на экран дисплея или печатающее устройство информации из БД; составление и печать отчетов по одной или нескольким БД; описание данных на специальном языке описания данных (ЯОД), обеспечивающем создание, добавление, удаление и модификацию БД без потерь данных; создание, редактирование и запуск командных файлов; обработка данных одновременно из нескольких файлов (до пяти) с использованием языка манипулирования данными (ЯМД), включающего операторы поиска, обновления и реорганизации данных из базы данных; организация данных в дисковой памяти; сортировка записей в файлах.

Программные средства ДАТАРЕАЛ обеспечивают реализацию следующих сервисных средств: меню-ориентированные программные средства для реализации прикладных приложений с возможностью автоматической генерации меню; реализация визуальной концепции при проектировании БД, при которой пользователь описывает и создает структуры и поля на экране; при этом содержимое экрана оперативно редактируется, просматривается, перемещается влево или вправо, сдвигается построчно вниз или вверх; осуществление доступа к командам операционной системы МДОС; создание командных файлов; обеспечение пошаговой помощи при вводе команд ДАТАРЕАЛ неподготовленным пользователем с использованием подсказок или заданием синтаксических шаблонов; сжатие данных в БД; связывание нескольких существующих таблиц в одну новую; управление устройствами; одновременный ввод данных в несколько таблиц; быстрая сортировка с использованием дополнительного пространства оперативной памяти; многоуровневая организация командных и процедурных файлов; подсчет промежуточных и полных сумм, а также минимума, максимума, среднего и других количественных характеристик в БД.

## Логическая структура

Каждая БД, создаваемая системой ДАТАРЕАЛ на диске, состоит из трех файлов, каждый из которых имеет идентификационный номер — 1, 2 или 3, а также расширение DRF. Например, БД с именем МУБАЗА состоит из следующих файлов: МУБАЗА1.DRF, МУБАЗА2.DRF, МУБАЗА3.DRF. Первый файл содержит определение структуры базы и информацию о расположении данных на внешнем носителе. Второй файл содержит собственно данные БД. Третий файл состоит из указателей для ключевых столбцов. Все три вышеупомянутых файла должны одновременно присутствовать на диске и никогда не должны разделяться. Ни один из файлов не является текстовым и не может быть модифицирован непосредственно каким-либо редактором текста по причине возможного повреждения данных.

БД состоит из таблиц отношений, число которых в одной базе не более 80. В каждой таблице может быть до 800 столбцов. Структура таблиц описывается пользователем и полностью зависит от его собственных требований к БД.

Каждая таблица БД состоит из одной или более строк данных (записей), которые, в свою очередь, делятся на один или более элементов данных (элементов столбцов). Таблица может вмещать такое количество записей, которое допускает внешнее запоминающее устройство, используемое для запоминания информации (гибкий диск или диск типа «Винчестер»). Отдельная запись (строка) в таблице может содержать до 4096 байт данных.

Каждая таблица в БД содержит от 1 до 800 столбцов. Столбец определяется собственным именем и типом элемента данных, который содержится в данном столбце. Допустимые типы данных для столбцов приведены в табл. 3.

Таблица 3

Тип данных	Описание
DATE	Буквенно-цифровая последовательность размером от 6 до 30 символов, определяющая дату. Представлена в формате: месяц, день, год. Устанавливается по команде SET DATE. Минимальное количество символов, определяющее месяц, день и год, может быть указано в формате MMDDYY
TIME	Буквенно-цифровая последовательность размером от 6 до 20 символов, определяющая время. Представлена в формате: часы, минуты, секунды. Устанавливается командой SET TIME. Минимальное количество символов, используемое для вывода времени, определяется в формате HHMMSS
CURRENCY	Тип данных, определяемый для работы со значениями, представленными в виде денежных единиц. Представляется цифровой последовательностью в диапазоне —99,999,999,999,999...+99,999,999,999,999,999, разделенной знаками «.» и «.» и устанавливаемой командой системы SET CURRENCY
REAL	Действительное число, содержащееся в диапазоне $-9E \pm 37 \dots +9E \pm 37$ . Действительные числа, имеющие до шести значащих цифр, изображаются как десятичные константы (989.989). Десятичные числа с более чем шестью значащими цифрами изображаются в формате с указанием порядка (987654321E-02)
DOUBLE	Действительные числа с двойной точностью в диапазоне $-10E \pm 307 \dots +10E + 307$ с пятнадцатью значащими цифрами
INTEGER	Целые числа в диапазоне $-999,999,999 \dots +999,999,999$
TEXT	Буквенно-цифровые данные. Длина по умолчанию составляет 8 символов. Максимальная длина текстового столбца — 1500 символов
NOTE	Текстовый элемент данных длиной до 4092 байт. Для практического использования столбец вмещает приблизительно 4050 байт

## Функции подсистем ДАТАРЕАЛ

В процессе работы с системой можно использовать следующие режимы и программные модули: командный режим; режим получения справок (HELP); режим подсказок (PROMPT); режим определения БД (DEFINE); режим загрузки данных (LOAD); режимы записи и выполнения файлов типа EXEC (RECORD, PLAYBACK); модуль определения баз данных (DEFINITION EXPRESS);



модуль создания приложений (APPLICATION EXPRESS); модуль работы с формами (FORMS EXPRESS); модуль генерации отчетов (REPORTS EXPRESS); модуль редактирования (DREDIT); модуль преобразования форматов файлов (FILEGATEWAY); модуль преобразования файлов в двоичный вид (CODELOCK).

Модули DEFINITION EXPRESS, FORMS EXPRESS и APPLICATION EXPRESS являются меню-управляемыми операционными модулями. Модули RECORD и PLAYBACK не имеют определенного отображаемого на экране формата. Модуль DREDIT является редактором произвольных текстов и исходных текстов приложений.

В командном режиме разрешается выполнять любую команду системы ДАТАРЕАЛ, пока высвечивается приглашение R>.

В режиме получения справок (HELP) пользователь обеспечивается оперативной информацией о назначении и использовании отдельных модулей и команд, а также сведениями о синтаксисе команд системы и о необходимой последовательности действий.

Неподготовленный пользователь получит необходимые сведения о составлении и выполнении стандартных команд системы, используя режим подсказки (PROMPT). Пользователь может создать свой собственный файл подсказок и использовать его во время работы в командном режиме.

В режиме DEFINE пользователем определяется структура БД, ее атрибуты, таблицы, пароли и правила. Этот режим является альтернативным определению БД с применением модуля DEFINITION EXPRESS, позволяющего в режиме меню и по подсказкам определять структуру БД неподготовленному пользователю.

В режиме LOAD пользователем вводятся собственно данные в определенную таблицу БД. Этот режим является альтернативным загрузке данных в БД с использованием модуля FORMS EXPRESS, позволяющего вводить информацию одновременно в пять таблиц.

В режиме RECORD пользователь может сохранить набранную с клавиатуры информацию в специальном файле ОС типа EXEC, который в дальнейшем может быть считан в режиме PLAYBACK.

С помощью программного редактора DREDIT пользователь имеет возможность создать исходный текст собственной программы и сохранить его в независимом от файлов БД файле, поддерживаемом ОС. С помощью DREDIT можно вводить или редактировать тексты размером до 800 строк по 80 символов в каждой строке.

Модуль FORMS EXPRESS используется для создания экрана ввода данных (входной формы) размером от одной до пяти таблиц; модуль REPORTS EXPRESS — для формирования выходных форматов отчетов или сводок по данным, хранящимся в БД; модуль APPLICATION EXPRESS — для создания неподготовленным пользователем полной системы приложений.

APPLICATION EXPRESS осуществляет процесс построения приложения — через меню или систему подсказок.

Модуль FILEGATEWAY является средством преобразования форматов файлов, отличных от

формата системы ДАТАРЕАЛ. Используя этот модуль в ДАТАРЕАЛ, можно получать и обрабатывать данные из файлов, созданных системами DBASE II и III, LOTUS 1-2-3, SYMPHONY и PFS:FILE, а также из DIF-файлов системы VISICALC, SYLK-файлов системы MULTIPLAN и файлов в формате ASCII из других систем. С помощью средств модуля FILEGATEWAY пользователь имеет возможность передать данные из файла БД, поддерживаемого ДАТАРЕАЛ, в файл, который может быть обработан системами LOTUS 1-2-3, SYMPHONY, VISICALC, MULTIPLAN или любой другой программой, работающей с файлами в формате DIF, SYLK или ASCII.

Модуль CODELOCK используется для преобразования файлов приложений в двоичный формат.

При этом возможно сокращение времени выполнения для некоторых командных файлов, поскольку модуль CODELOCK частично интерпретирует отдельные командные строки ДАТАРЕАЛ в процессе их преобразования. Кроме того, применяется как средство защиты файлов приложений, поскольку файлы в двоичном формате не могут быть изменены редактором текста.

Программные средства ДАТАРЕАЛ позволяют использовать информационные файлы программных систем LOTUS 1-2-3, SYMPHONY, DBASE II/III и PFS:FILE без дополнительных преобразований, а также позволяют осуществлять обмен информацией с широко используемыми системами обработки данных VISICALC, MULTIPLAN и обмен информационными файлами в формате ASCII, DIF, SYLK от других вычислительных систем при наличии поддержки вышеуказанного обмена операционной средой соответствующих вычислительных комплексов.

Система ДАТАРЕАЛ ориентирована на реализацию основных концепций, установленных в широко известной системе R:BASE SYSTEM V, и совместима с ней на программном и информационном уровне.

## Применение

Программные средства ДАТАРЕАЛ функционируют при следующей минимальной конфигурации технических средств СМ 1810: процессор СМ 1810.13; оперативная память не менее 512 Кбайт; видеоконтроллер типа СМ 1810.7005; модуль индикации М32Ц1; устройство печати типа СМ 6329.01; устройство внешней памяти на гибких дисках типа СМ 5639; устройство внешней памяти на магнитных дисках типа «Винчестер» СМ 5514.

Программные средства ДАТАРЕАЛ ориентированы на возможность эксплуатации под управлением операционных систем МДОС 1810 и MS DOS и моделей ЭВМ, программно-совместимых с СМ 1810.13 и IBM PC XT/AT. При наличии необходимых терминальных драйверов (ALFA, BETA), система ДАТАРЕАЛ может использоваться на ПЭВМ типа ЕС-1841 или «Искра-1030».



# Интегрированные средства обработки данных для СМ 1810 (СМ 1820) (ИНТЕРСОД)

## Назначение

Система ИНТЕРСОД — относится к классу интегрированных программных средств, предназначенных для автоматизации информационной технологии, на уровне рабочего места администратора, плановика, экономиста-аналитика, статистика, финансового работника и т. д.

ИНТЕРСОД представляет собой пакет программ, обеспечивающих следующие возможности на комплексах СМ 1810: вычисление числовых величин и анализ результатов с использованием технологии моделирования на основе крупноформатных электронных таблиц (КЭТ); электронное написание и редактирование текстовых документов (текстовой информации) с помощью текстового процессора; иллюстрирование и анализ изменений числовых величин с помощью средств построения графиков по числовым данным (средства деловой графики); структурирование и управление данными на основе табличной модели данных с помощью средств управления БД; организация и управление последовательностью обработки данных на основе широкого набора сервисных средств, включая многооконный интерфейс ввода-вывода, режим оперативной электронной помощи, режим обработки файлов. Использование возможностей ИНТЕРСОД осуществляется в интегрированной операционной среде. Интегрированность программных компонент системы является основной характеристикой ИНТЕРСОД. Это означает, что пользователю представляется возможность работать в специфической среде любой компоненты (например, в СУБД) и при необходимости переходить в среду другой компоненты (например, в среду обработки таблиц) достаточно просто и единообразно.

Очень близким аналогом системы ИНТЕРСОД является система SYMPHONY, имеющая кроме того, программную компоненту, обслуживающую связь (обмен данными) с другими компьютерами.

## Общие сведения о пакете

Реализация возможностей обработки данных, заложенных в ИНТЕРСОД, осуществляется путем генерации рабочих сред в соответствии с типом выполняемых работ, с помощью сервисных (служебных) средств, позволяющих настраивать пакет применительно к конкретным условиям его эксплуатации.

ИНТЕРСОД позволяет генерировать четыре рабочие среды: крупноформатных электронных таблиц; текстового процессора; управления БД; деловой графики.

Каждая из рабочих сред характеризуется своим набором операторов, доступных пользователю только в этой среде. При необходимости выполнить

оператор, относящийся к другой среде, пользователь должен предварительно изменить тип рабочей среды.

Ввод-вывод ИНТЕРСОД осуществляется на основе многооконного интерфейса и дает возможность пользователю просто и эффективно организовать обработку данных в соответствии со своими информационными потребностями.

Пользовательский интерфейс пакета реализован по двум типам: меню-стиль-выбор пользователем следующего действия из изображенного на экране меню; заполнение «электронных бланков» — ввод и изменение данных производится пользователем в поля предварительно форматированного и изображенного на экране терминала формуляра.

Для расширения возможностей обработки данных в пакете имеется набор встроенных функций различных типов. Набор достаточно разнообразен, значительно повышает сервис прикладного программирования.

Одним из важных преимуществ пакета является наличие электронной документации, включающей обучающую программу и средства консультативной контекстно-зависимой помощи, т. е. средства быстрого (нажатием клавиши) получения поясняющей информации по применяемым в текущий момент средствам системы.

Обучающая программа предоставляет возможность неподготовленному пользователю изучить и опробовать различные средства пакета без постоянной помощи.

Средства помощи включают взаимосвязанный набор экранов помощи и программу управления вызовом этих экранов с пользовательским интерфейсом по типу меню-стиль.

Наличие электронной документации, несмотря на большое разнообразие средств пакета, значительно упрощает освоение пакета и делает его «дружелюбным» по отношению к пользователю.

К основным характеристикам пакета следует отнести средства конфигурирования, которые используются при настройке пакета на конфигурацию вычислительной системы, а также пользовательский алфавит и изображение даты.

Ни одна, даже очень тщательно спроектированная система, не может реализовать все многообразие ситуаций реальной действительности, поэтому в пакете реализована возможность приостановить его работу и вызвать на выполнение прикладную программу, используя меню прикладных программ; кроме того, возможен переход в режим ОС, в среде которой функционирует пакет. После выполнения необходимых пользователю программ можно вновь вернуться в режим работы пакета.

Многооконный интерфейс ввода-вывода организован следующим образом. Окно в ИНТЕРСОД — это прямоугольная область на экране дисплея, которая отображает вводимые или выводимые пользователем данные независимо от их типа, а также выполняемой им работы. Все данные размещаются во время сеанса в оперативной памяти, организованной в форме рабочей КЭТ, в дальнейшем называемой просто рабочей таблицей. Таким образом, в каждый данный момент времени окно отображает некоторую прямоугольную область рабочей таблицы.

Необходимо отметить, что одна и та же часть рабочей таблицы может быть отражена через раз-



ные окна и одновременно одному окну могут быть назначены разные части рабочей таблицы.

Средства организации интерфейса ввода-вывода позволяют создать необходимое для пользователя количество окон. При этом каждое окно идентифицируется именем. Пользователь может работать одновременно только с одним окном, которое называется текущим. Смена текущего окна или доступ пользователя к данным в рабочей таблице через другое окно осуществляется по имени окна. Имена окон организованы в список по типу стека. Пользователь имеет возможность управлять размером окна и его положением на экране. Окна могут перекрывать друг друга, а также временно окно может быть скрыто. Скрытое окно не видно на экране, но его параметры известны системе, и имя его присутствует в стеке окон. Самой важной характеристикой окна является его тип. Указанием типа окна пользователь определяет (генерирует) необходимую ему рабочую среду. В соответствии с количеством рабочих сред в ИНТЕРСОД имеются четыре типа окна: окно типа SHEET позволяет определить среду КЭТ; окно типа DOC — текстового процессора; окно типа FORM — управления базами данных; окно типа GRAPH — деловой графики.

Организация пользовательского интерфейса на основе меню и заполнения электронных бланков значительно снижает квалификационные требования к пользователям и, несмотря на широкие возможности, заложенные в пакете, а также большое количество операторов, делает пакет ИНТЕРСОД удобным в эксплуатации и легко усвояемым.

ИНТЕРСОД предоставляет пользователю некоторый фиксированный набор команд, каждая из которых реализует отдельную функцию системы. Команда снабжена именем. Выполнение команды инициализируется выбором ее имени из меню на экране. Выбор осуществляется указанием имени с помощью курсора (прямоугольная область экрана) или набором на клавиатуре первой буквы имени. Все множество команд и соответственно множество их имен организовано в иерархическую структуру.

На самом верхнем уровне команды подразделяются на команды рабочих сред и служебные команды (сервисные). Каждой рабочей среде соответствует свой набор команд, общее описание которых приводится ниже. Служебные команды являются общими для всех рабочих сред и дают возможность пользователю управлять многооконным интерфейсом, печатать необходимые данные на различных устройствах, осуществлять обмен данными с дисковыми носителями (имеется набор средств обработки файлов), устанавливать параметры пакета, управлять выполнением дополнительных прикладных программ, конфигурировать ИНТЕРСОД.

В ИНТЕРСОД принята английская нотация имен команд.

Среда обработки КЭТ включает в себя следующие компоненты: табличный процессор, который обрабатывает таблицы, выполняет расчеты и интерпретирует инструкции входного языка; входной язык табличного процессора, который состоит из команд манипулирования данными, инструкций описания типов и форматов данных, синтаксис задания различных выражений, формул, операций и

т. д.; фактически это язык программирования для электронных таблиц; собственно КЭТ, описание которых дается на входном языке табличного процессора.

КЭТ в ИНТЕРСОД — это модель данных, аналогичная широко известным иерархическим, сетевым и реляционным моделям данных, используемая для организации хранения и обработки данных.

КЭТ можно условно представить как ОЗУ с прямым доступом или как обычный тетрадный лист в клетку, в ячейках или клетках которого помещается необходимая пользователю информация. Элементами КЭТ также являются ячейки (клетки), в которые пользователь записывает информацию или алгоритмы по ее обработке. Все ячейки организуются в виде таблицы, и каждой из них присваивается адрес, состоящий из обозначения столбца и номера строки.

Максимальные размеры таблицы в ИНТЕРСОД — 256 столбцов и 8192 строки. Столбцы обозначаются буквами латинского алфавита от А до Z (столбцы от 1 до 26), далее от AA до AZ (столбцы от 27 до 52), и т. д. до IU. Строки нумеруются числами от 1 до 8192.

При работе с моделью данных КЭТ пользователь, указывая адрес ячейки, производит различные манипуляции с данными, например, он может записать в ячейку число, текст или формулу, включающую процедуры обработки данных в других ячейках. Адрес ячейки состоит из обозначения столбца и номера строки, на пересечении которых стоит интересующая пользователя ячейка, например, A1867. В каждую ячейку пользователь может записать от 1 до 240 символов.

Во время работы ИНТЕРСОД хранит данные в оперативной памяти. КЭТ, с которой работает в данный момент пользователь, называется рабочей таблицей. Для работы в среде КЭТ необходимо, используя комплекс служебных команд, создать одно или несколько окон типа SHEET.

Для указания данных, с которыми должны быть проведены какие-либо действия, используется либо указатель ячейки (светлый прямоугольник на экране), либо ее адрес. В окне SHEET имеется ряд возможностей быстрого перемещения указателей ячейки в требуемую клетку таблицы.

В окне SHEET имеются следующие возможности размещения данных и манипулирования ими: ввод данных. В клетки таблицы могут быть введены: численные значения, алфавитно-цифровой набор символов, формула, имеющая в качестве аргументов содержимое клеток таблицы;

стирание введенных данных. Стирание может быть сделано как в отдельных клетках таблицы, так и в целых колонках (рядах) или в блоках из клеток таблицы (диапазонах);

удаление колонок (рядов) данных. На место удаленных колонок (рядов) ставятся смежные с ними колонки (ряды) данных;

перемещение данных. Данные могут быть перенесены из одной клетки в другую, из одной колонки (ряда) в другую колонку (ряд), из одного блока клеток (диапазона) таблицы в другой блок;

включение (вставка) колонок (рядов) данных. Ранее введенные колонки (ряды) данных «раздвигаются»;



копирование данных. Уже имеющиеся данные в клетке, колонке, ряде, блоке клеток копируются в другую часть таблицы. При копировании блоков клеток с данными можно расположить колонки данных как ряды, а ряды данных — как колонки (транспонирование);

изменение, задание формата выводимых данных. Система ИНТЕРСОД располагает многими форматами и стилями воспроизведения чисел, дат, времени суток. Пользователь может задавать требуемый ему формат в указанных им колонках, рядах, блоках данных. Ширина колонок и расположение данных в колонке (по левому краю, в середине, по правому краю) могут быть также заданы пользователем.

ИНТЕРСОД располагает широким набором средств, позволяющим устанавливать отношения между данными, расположенными в клетках таблицы. Такими средствами являются формулы-инструкции для ИНТЕРСОД по выполнению определенных действий над данными: арифметических, логических, строковой арифметики (выявления количественных характеристик введенного текста и формирование нового текста по заданной инструкции). Формулы вводятся в те же клетки таблицы, где должны находиться и результаты вычислений по этим формулам. При изменении аргументов формул, т. е. данных из клеток таблицы, новый результат каждой формулы появляется в «ее» клетке.

Простые арифметические и логические операции, используемые в формулах: возведение в степень; унарный (одиночный) плюс; унарный минус; умножение; деление; сложение; вычитание; меньше; больше; меньше или равно; больше или равно; равно; не равно; логическое НЕТ; логическое включающее ИЛИ; логическое И; конкатенация.

Кроме выполнения указанных стандартных операций, которые организует сам пользователь, используя соответствующие обозначения в формулах, ИНТЕРСОД может выполнять в КЭТ также вычисления специального вида с помощью встроенных инструкций, называемых *a*-функциями. В КЭТ имеется следующий набор *a*-функций: математические функции; функции для операций с датами; функции статистики БД; логические функции; финансовые функции; функции строковых операций; статистические функции.

В практических применениях часто требуется оценить, к каким результатам могут привести изменения каких-либо отдельных параметров по принципу «что произойдет, если ...». Такой анализ при больших размерах таблицы произвести непросто, так как анализируемые параметры могут оказаться в удаленных друг от друга клетках. Кроме того, изменение исходного параметра поведет к непредусмотренному изменению зависящих от него. ИНТЕРСОД в окне SHEET предоставляет пользователю средство «что, если», позволяющее построить таблицу анализируемых параметров и на ней наглядно «проигрывать» различные варианты изменяющихся параметров из различных клеток рабочей таблицы, в специально отведенном для этого участке таблицы.

Сервисная среда ИНТЕРСОД предоставляет пользователю средства вывода на печать данных, введенных в таблицы. При этом оказываются весьма полезными следующие свойства печатного форматирования, специфичные для окна SHEET:

Вывод на печать широких отчетов. В некоторых приложениях может потребоваться вывести на печать таблицу, имеющую ширину, превышающую максимальную длину строки печатающего устройства. Для печати таких таблиц пользователь должен задать требуемую ему ширину полосы, не превышающую максимальную длину строки, и ИНТЕРСОД отпечатает таблицу полосами.

Печатные метки. Такое средство ИНТЕРСОД позволяет выводить на печать заголовки колонок либо строк на каждой странице, что делает текст более удобным для использования.

Печать листинга с формулами таблицы. В ряде случаев необходимо знать не только результат вычислений, но и формулу, по которой были сделаны эти вычисления. ИНТЕРСОД предоставляет такие средства.

Фундаментальная характеристика КЭТ в системе ИНТЕРСОД — перевычисление формул, т. е. автоматическое вычисление новых результатов в ячейках с формулами каждый раз, когда меняются аргументы формул, т. е. данные, вводимые в какие-либо клетки таблицы. КЭТ располагает также средствами ручного управления перевычислением.

**Среда обработки текстовой информации (СОТИ)** в ИНТЕРСОД содержит: текстовый процессор, который выполняет операции по вводу и форматированию текстовых документов; входной язык текстового процессора, включающий команды и инструкции; собственно текстовый документ, формируемый и описываемый при помощи языка текстового процессора.

СОТИ и ИНТЕРСОД используется при написании писем, деловых бумаг, статей и т. д.

При вводе текста осуществляется автоматическое форматирование документа в соответствии с первоначальными установками. Возможно также его редактирование и изменение. Важным свойством СОТИ является возможность видеть текст на экране так, как он будет печататься на устройстве печати. Это избавляет пользователя от лишних затрат на печать документа при редактировании.

При работе СОТИ вводимый документ записывается в текущую рабочую таблицу. Ввод информации осуществляется через текущее окно типа DOS в левый крайний столбец диапазона, являющегося областью действия окна. Например, если область действия текущего окна A11...C125, то весь текстовый документ запишется в диапазон A11...A125.

Используя возможности системы, можно создавать на экране несколько окон и одновременно просматривать в них различные участки текста.

Средства автоматического форматирования документа позволяют осуществить: установку табуляции; выравнивание по левой границе; установку интервала распечатывания строк; установку левой и правой границ; отображение ВК как спецсимвола; отображение пробелов как интервалов.

Перечисленные установки системы действуют по умолчанию. Они могут быть изменены введением форматной строки или изменением установочной таблицы.

Средства редактирования позволяют быстро находить любое место в документе перемещением курсора на несколько символов, слов, предложений, абзацев, страниц экрана. Можно переместиться к указанному символу или строке. Средствами



системы можно быстро заменить один символ, слово или фразу во всем тексте. При редактировании можно выявить блок текста и выполнять с ним операции по удалению, перемещению и размножению. Таким образом, можно изменять структуру документа.

**Среда деловой графики (СДГ) ИНТЕРСОД** включает в себя: процессор деловой графики, который обрабатывает конструкции входного языка и строит изображения графиков; входной язык процессора, который включает инструкции, команды, синтаксис и тому подобные элементы, необходимые для получения деловых графиков; собственно деловые графики, описание которых дается на входном языке процессора деловой графики.

СДГ ИНТЕРСОД предназначена для представления числовых данных в форме различных графических изображений (графиков) на устройствах графического вывода информации (графический видеотерминал, устройство печати, графопостроитель).

Построение графиков производится на основе наборов числовых данных, обрабатываемых в рабочих таблицах ИНТЕРСОД. Графики быстро просматриваются на экране терминала и позволяют наглядней проанализировать соотношение и тенденцию изменений числовых переменных, чем при обычном просмотре рядов и колонок чисел в рабочих таблицах. Распечатка графиков производится вспомогательной программой. Print-Graph и требует затрат определенного времени. Большой выбор возможностей данной программы позволяет получать качественные табуляграммы графиков, которые используются при изготовлении самой различной документации, делая ее наглядней и доступней для усвоения. Программа позволяет получать табуляграммы созданных графиков при отсутствии графического видеотерминала, когда нет возможности увидеть график на экране.

СДГ ИНТЕРСОД обеспечивает поддержку цветных устройств графического вывода. При этом можно получать до семи различных цветов.

СДГ ИНТЕРСОД позволяет создать (определить) различные формы графиков по числовым данным, введенным в рабочую таблицу. При наличии графического терминала (черно-белого или цветного) можно просмотреть рисунок на экране, изменить информацию внутри графика и внешний вид графика. При этом нет необходимости заново набирать данные или создавать график, достаточно только сообщить ИНТЕРСОД, какой набор данных включить в график, либо какой вид рисунка графика необходим. ИНТЕРСОД позволяет моментально увидеть на экране рисунок нового графика.

На одном графике можно изобразить до шести наборов данных, составленных из чисел и формул, значения которых будут представлены в графической форме. Можно корректировать такие наборы данных и немедленно просматривать график. Новые значения данных будут автоматически отображены на графике.

СДГ ИНТЕРСОД поддерживает следующие типы диаграмм и графиков: круговая и разомкнутая круговая диаграмма (циклограмма) представляет собой круг, разделенный на части, которые показывают соотношения между различными величинами. Этот вид графических изображений может

быть особенно полезен для сравнения отдельных частей целого. СДГ ИНТЕРСОД может заштриховывать по-разному каждый сектор или отделять какие-либо секторы круговой диаграммы (разомкнутая круговая диаграмма). В отличие от других графиков круговая диаграмма может быть построена только по одному набору значений;

прямоугольная диаграмма (гистограмма) представляет собой одну или несколько групп прямоугольников, высота которых показывает соотношения различных данных. Прямоугольные диаграммы могут быть полезными для сравнения небольшого количества величин через определенные интервалы времени, а также и для других целей. Такие величины можно легко проанализировать путем наглядного сравнения высот соответствующих прямоугольников;

составная прямоугольная диаграмма (гистограмма) подобна обычным прямоугольным диаграммам и отличается только тем, что прямоугольники в ней располагаются друг над другом, а не рядом, как в обычных прямоугольных диаграммах. Это позволяет сравнивать не только величины отдельных прямоугольников, но и суммарные размеры групп прямоугольников. В этом случае нет необходимости иметь отдельный прямоугольник для представления общего количества величин;

верхне-нижне-закрыто-открытая схема (интервальная диаграмма) представляет особый тип прямоугольной диаграммы, где группа прямоугольников заменяется вертикальной линией с делениями (шкалой), соответствующими высотам прямоугольников. Это позволяет, например, выявлять тенденции и проводить другие анализы изменения величин через определенные периоды времени. Кроме графического представления верхних, нижних, закрытых и открытых значений можно строить графики только верхних значений, верхне-нижних значений или верхне-нижне-закрытых значений;

линейный график представляет собой набор точек данных, построенный в системе координат по горизонтальной оси или оси X. Место расположения точки данных на графике соответствует числовому значению элемента данных, которое отображается графически. Каждому набору данных соответствует определенный графический символ (например, треугольник, квадратик и т. д.), который изображается на графике в местах расположения точек данных. Стандартная процедура СДГ ИНТЕРСОД рисует графический символ на каждой точке данных, затем соединяет точки линией. Дополнительно каждая точка данных может быть отмечена на графике соответствующей меткой, называемой меткой данных. Линейные графики наглядно показывают, как числовые значения изменяются через определенные интервалы времени;

координатный график (график в осях XY) представляет собой график в системе координат. Ось X в этом графике больше не используется только для меток (меток оси X или меток диапазона X). Теперь это числовая шкала, точно так же, как и ось Y. Один или несколько наборов точек данных (значения по оси Y) строятся по другому набору точек данных (значения по оси X). Графики в осях XY представляют пары величин, а не последовательности отдельных величин, как это делалось в линейных графиках.



СДГ ИНТЕРСОД, используя возможности работы с окнами, может быть реализована в следующих режимах: совмещенный режим, позволяющий организовать работу в нескольких окнах разного типа одновременно; последовательный режим, в котором просматривается только текущее окно типа GRAPH с графиком. Если текущим является любое другое окно, то виден только контур окна GRAPH, а не сам график; режим двух экранов (двойной режим), предполагает наличие двух видеотерминалов. В этом случае видеотерминал используется только для просмотра графиков.

В разделенном режиме ИНТЕРСОД предоставляет пользователю следующие возможности: строить несколько графиков одновременно; воспроизводить различные графики различными способами. Например, можно вывести на экран несколько маленьких графиков для предварительного просмотра, затем увеличить любой из них до размеров всего экрана для более тщательного изучения; вывести графики и в то же время числовые данные, по которым эти графики построены. Это увеличивает возможности в оценке ситуации, возникающей при работе в среде КЭТ; изменять входные данные и заново воспроизводить график, который позволит наглядно проанализировать влияние внесенных изменений; строить график в зависимости от анализа условия «What-if» (что, если) посредством вычисления, точно так же, как при работе с КЭТ; использовать одно окно типа GRAPH для просмотра множества графиков один за другим.

**Примечание.** СДГ ИНТЕРСОД позволяет проводить всю работу по созданию графиков в окне типа SHEET, используя полностью мощные средства КЭТ. В этом случае определение и временный просмотр графика выполняют графическими командами из меню SHEET.

## Применение

Объектами обработки ИНТЕРСОД являются данные о любой предметной области и связи между ними, которые могут быть представлены в алфавитно-цифровой или цифровой форме. При обработке этих данных средствами ИНТЕРСОД наиболее приемлемыми моделями данных являются модели экономических процессов, имеющих целью анализ, прогнозирование и планирование. Другой важной моделью, эффективно реализуемой в ИНТЕРСОД, является экспресс-моделирование при выполнении проектных и научно-исследовательских работ.

Для упрощения построения моделей обработки данных в ИНТЕРСОД имеется большой набор встроенных в систему функций.

В комплект поставки ИНТЕРСОД входит программа установки, которая позволяет настроить пакет на имеющийся в распоряжении вычислительный комплекс (ВК). Для работы ИНТЕРСОД необходимо использовать ВК типа СМ 1810, имеющие в своем составе как минимум двусторонний накопитель на ГМД. Следует отметить, что конфигурация используемого ВК влияет на возможности, предоставляемые пользователям пакетом во время работы.

Для более полного использования возможно-

стей ИНТЕРСОД рекомендуется применение типового ВК СМ 1810.13.

Интегрированное средство обработки данных ИНТЕРСОД должно эксплуатироваться под управлением МДОС 1810. Оперативная память, используемая при работе с ИНТЕРСОД, должна быть не менее 512 Кбайт. Следует отметить, что от объема оперативной памяти, предоставленной ИНТЕРСОД, зависит размер КЭТ, доступной пользователю.

Пользователи, применяющие пакет ИНТЕРСОД, должны уметь работать с МДОС 1810 на уровне минимально подготовленного оператора, т. е. они должны уметь: включить ВК; загрузить МДОС 1810; выполнять такие команды ОС, как DIR, COPY, DELETE и т. п.

Пакет ИНТЕРСОД включает в себя следующие независимые программные комплексы: программа «InterSOD» реализует все четыре среды обработки данных, а также среду служебных команд; программа «Install» реализует настройку ИНТЕРСОД на вычислительный комплекс пользователя; программа «Tutorial» реализует обучающую программу ИНТЕРСОД; программа «Trans» реализует реформатирование файлов данных, подготовленных средствами ИНТЕРСОД, в форматы других систем и обратно; программа «Print Graph» обеспечивает печать деловых графиков, подготовленных в среде деловой графики.

Комплект поставки ИНТЕРСОД как программного изделия включает в себя следующие составные части: тексты программ на ГМД ( $\varnothing$  133 мм) в количестве шести штук; эксплуатационная документация в шести книгах на 600 листах; шаблон клавиатуры; упаковки изделий, в том числе текстов программ, эксплуатационной документации и всего изделия.

## Система управления базами данных реляционного типа, функционирующая под управлением дисковых операционных систем АСПО СМ ЭВМ (БАРД ДОС АСПО)

Система предназначена для интеграции информации в базах данных с доступом к ним в диалоговом режиме и из программ пользователя в вычислительных системах на базе ЭВМ архитектурной линии СМ-1/СМ-2 (СМ-2М, СМ 1634, СМ 1210).

Функционирование БАРД поддерживается дисковыми операционными системами (ДОС) АСПО СМ ЭВМ.

СУБД БАРД является многобазовой системой. В ней параллельно и независимо может осуществляться взаимодействие пользователей с несколькими БД. Доступ к ЛБД производится через системную базу, в которой содержится вся необходимая информация о всех имеющихся в системе



ЛБД. Количество ЛБД и размеры памяти под каждую из них задаются при логическом проектировании в зависимости от потребностей пользователя.

Внутри ЛБД данные группируются в двумерные таблицы (отношения). В каждой таблице данные хранятся построчно (покортежно). Строка может содержать данные различных типов, но данные одного столбца (атрибута) должны быть одного типа. Возможны целые, вещественные и символьные типы данных. Символьный тип представляет собой строку символов произвольной (до 256 символов), но фиксированной для данного столбца длины.

**Логическая организация данных.** Реляционная модель данных обеспечивает простую и удобную интерпретацию данных пользователем и в то же время возлагает на СУБД всю полноту функций по организации хранения и выбору путей доступа к данным. Это обуславливает довольно сложную логическую структуру информации системы БАРД.

Доступ к ЛБД системы БАРД осуществляется через системную базу «#систбаз». Логическим ядром последней является каталог «Базы», содержащий информацию о всех ЛБД системы. С другой стороны «#систбаз» можно рассматривать и как локальную базу. Ее пользователи имеют возможность создавать и вести информационные отношения.

Каждая ЛБД (в том числе и «#систбаз») содержит каталоги, с целью унификации доступа выполненные в виде отношений (системные отношения) и данные пользователей, собранные в информационные отношения. Доступ ко всем отношениям ЛБД осуществляется через стандартный набор каталогов: «Таблицы», «Столбцы» и «Доступ». Эти отношения имеют стандартную структуру и обязательно присутствуют в каждой ЛБД.

Информации о каждом (в том числе и системном) отношении БД соответствует один кортеж каталога «Таблицы». Здесь часть информации статична и помещается при создании отношения. Это номер и имя отношения, идентификатор владельца отношения, количество атрибутов в отношении, длина кортежа отношения в байтах, размер памяти, отведенной под отношение, а также информация о способе физического хранения данных отношения. В каталог заносится также признак, позволяющий отличать системные отношения от информационных. Динамические параметры отношений содержат информацию о длине отношений (количество кортежей), текущих адресах свободной памяти для размещения вновь поступающих кортежей, количество удаленных кортежей.

Параметры всех атрибутов всех отношений ЛБД сведены в системный каталог «Столбцы». Каждому атрибуту соответствует кортеж отношения, содержащий информацию о потере отношения ЛБД, к которому принадлежит атрибут, его местонахождение в кортеже этого отношения (сдвиг в байтах от начала кортежа и номер атрибута в отношении), типе данных атрибута, номере атрибута в ключе (для отношений с ключевыми способами организации хранения данных) и для символьного атрибута его нотация (русская, латинская, смешанная).

Пользователи ЛБД регистрируются в каталоге «Доступ», каждый кортеж которого содержит имя

и пароль пользователя допущенного для работы с этой ЛБД.

**Организация хранения данных.** Информационными компонентами БАРД являются БД. С точки зрения ДОО АСПО каждая ЛБД представляет собой отдельный непрерывный файл на магнитном диске, либо совокупность таких файлов на разных дисках (до 7) с именем, совпадающим с именем ЛБД. Все файлы имеют структуру файлов с записями постоянной длины в 512 байт (2 сектора). Такая запись называется блоком или страницей.

Каждый файл является томом БД. Размер тома задается в блоках и ограничен только объемом памяти магнитного диска, на котором располагается данный том.

ЛБД делится на область отношений и свободное пространство. Область отношений расширяется по мере создания новых отношений, уменьшая свободное пространство. Том ЛБД имеет страничную организацию. Отношение занимает целое число страниц тома и может переходить из тома в том.

Доступ к кортежам внутри отношений зависит от способа организации и хранения данных.

Способы размещения кортежей внутри отношения в БАРД называют схемами сохранения данных.

СУБД БАРД поддерживает три наиболее распространенных схемы сохранения: неключевая структура («Куча»); структура с прямым доступом по ключу («прямая» или «ХЕШ»); структура с индексно-последовательным доступом по ключу («индексно-последовательная» или «ИСАМ»).

При неключевой структуре кортежи помещаются в отношение независимо от конкретных значений атрибутов последовательно в порядке поступления. «Куча» требует минимальных затрат памяти и времени при записи кортежей в отношение. Она является рациональной в тех случаях, когда требуется полное (почти полное) сканирование отношения.

По сравнению с «Кучей» ключевые структуры обеспечивают наиболее быстрый доступ к данным, если корректно и полно заданы значения ключевых атрибутов.

Если для отношения можно указать ключевые атрибуты, значения которых однозначно определяют кортежи отношения, то пользователь БАРД может использовать ключевую схему сохранения «ХЕШ». При использовании этой схемы адреса хранения кортежей в пределах отношения определяются значениями некоторой ХЕШ-функции, для которой аргументом являются значения ключевых атрибутов кортежей. Эта схема обеспечивает быстрый доступ к отдельным участкам отношения. В составе ПО БАРД имеется одна программа вычисления значения ХЕШ-функции по методу деления. В документацию системы включены указания пользователю по программированию и включению в ПО ХЕШ-функций пользователя.

Индексно-последовательный метод доступа реализован в БАРД с помощью так называемого «обобщенного справочника». При такой организации значения ключевых атрибутов (в отличие от схемы «ХЕШ» определяют не точный адрес местонахождения кортежа, а группу страниц отношения, которую уже необходимо сканировать. «Обобщенный справочник» представляет собой таблицу соответствия диапазонов значений ключевого атрибута диапазонам страниц отношения.



Для обработки переполнений в ключевых схемах сохранения используются страницы переполнения.

**Манипулирование данными.** Используемый в БАРД декларативный язык Шаблон близок к известным языкам QVEL (INGRES) и ЯНОТ (СУБД БАРС). В отличие от них Шаблон является позиционным языком. За счет этого он практически не содержит ключевых слов, разделителей, ограничений на символы имен отношений и атрибутов, а также более нагляден и прост в изучении и использовании.

Шаблон является непроецедурным, реляционно полным языком реляционного исчисления с переменными — кортежами.

Шаблон ориентирован на неквалифицированного пользователя. При формировании запроса пользователь заполняет специальную таблицу, высвечиваемую на экране терминала. Ввод текста осуществляется синтаксическим кадровым редактором БАРД, который не позволяет оператору ввести неверный текст, своевременно сообщает о допущенной ошибке и снова высвечивает уже поправленный запрос.

Форма запроса на языке Шаблон.

Выбери	Выдай	Печатай	Передай	Добавь	Замени	Удали	Сортверх	Сортниз
			ЗН АК				Константа	

Первая строка представляет собой горизонтальное меню операторов языка.

Следующие три строки экрана занимает «шапка» запроса, в которую заносятся имена отношений, откуда (куда) будут выбираться (заноситься) данные. В первом поле указывается имя отношения — цели. В следующих трех полях могут быть указаны до трех отношений с исходными данными для реализации запроса. Эти же имена должны быть продублированы в полях после «знака».

Ниже «шапки» располагаются атрибутивные строки, которые состоят из девяти полей. В первом поле указываются имена атрибутов целевого отношения. В полях со второго по четвертое — имена атрибутов, соответствующих отношениям, заданным в «шапке», которые участвуют в левой части условий запроса. Пятое поле предназначено для указания знака в условиях запроса (>, <, =, >=, <=, ≠). В следующих трех полях задаются имена атрибутов-источников (для выбора данных в целевой атрибут) или имена атрибутов соответствующих отношений, которые участвуют в правой части условий запроса. В последнем поле (поле-константа) может быть задана константа любого типа, разрешенного в БАРД. Эта константа может быть источником для целевого атрибута или участвовать в правой части условия запроса.

Две последние строки экрана — строки редактора. Предпоследняя строка служит для выдачи сообщений синтаксического редактора и пользова-

телем не заполняется. В последней строке задаются команды редактора.

Запрос, набранный на экране, может быть принят системой БАРД непосредственно к исполнению или (после трансляции) может быть помещен в файл ДОС АСПО на магнитном диске. В последнем случае этот запрос может быть инициирован программой пользователя.

Любое имя, набранное в запросе, может быть помечено с помощью специального признака как программная переменная. Запрос, в котором есть программные переменные, может быть инициирован только с помощью программы пользователя. Перед выполнением запроса с помощью этой программы передаются фактические имена отношений, атрибутов или конкретные значения констант, на которые будут заменены указанные программные переменные.

Операторы «Выбери», «Выдай», «Печатай», «Передай» реализуют операцию выборки и выдачу результатов соответственно в базу данных, на терминал, на внешний носитель, в программу пользователя. Операция обновления реализуется с помощью оператора «Замени», включения — «Добавь» и удаление оператором «Удали». Все перечисленные

операции реализуются с учетом условий, сформированных в запросе.

Шаблон подобно языку QVEL не предусматривает автоматического удаления дубликатных кортежей при вычислении отношения. Однако в языке предусмотрены операторы сортировки «Сортверх» и «Сортниз» для устранения дубликатов в то время, когда отношение будет сортироваться по указанным атрибутам. Кроме того, эти операторы могут быть использованы для физического упорядочения отношения БД по указанным в порядке старшинства атрибутам.

В языке Шаблон реализованы следующие агрегатные функции: МИНИМ (наименьшее значение), МАКСИМ (наибольшее значение), СРЕДН (среднее значение), СУММА (сумма), ЧИСЛЗН (число значений). Все перечисленные функции могут присутствовать в запросе в качестве источника значений целевого атрибута или в правой части условий запроса.

С помощью команд редактора языка производится оперативное управление процессом формирования запроса на экране терминала. Командой редактора можно инициировать получение справочной информации по заполнению экрана.

Для более эффективной работы пользователя при формировании запроса на экране терминала предусмотрено программное управление курсором в поле экрана. Управление перемещением курсора



производится с помощью специальных клавиш терминала.

**Средства создания и ведения баз данных.** СУБД БАРД предоставляет администратору и пользователю базы данных большой набор сервисных средств для ее ведения. Реализация конкретной функции сервиса системы осуществляется с помощью команды терминального монитора БАРД. Набор команд системы предлагается в виде двухуровневого меню. Параметры выбранной команды уточняются в ходе диалога.

Набор команд БАРД позволяет инициировать следующие операции: логическое проектирование ЛБД; создание схем отношений (как в диалоговом режиме, так и из файла с заранее подготовленной информацией); создание файла запросов; санкционирование доступа к данным базы новых пользователей; удаление ЛБД из системной БД, удаление пользователей и отношений из ЛБД; переход от ключевых схем сохранения отношения к неключевой, а также от неключевой схемы к схеме «ХЕШ»; изменение имен отношений и паролей пользователей; корректировка ранее сформированных в файле запросов; восстановление ЛБД в системной БД и восстановление ранее удаленных отношений; «уплотнение» информации в ЛБД и отношениях; визуализация каталоговой информации; копирование отношений ЛБД в файлы на внешних носителях и обратно; формирование файлов для программ печати отчетов; загрузка отношений ЛБД заранее подготовленной информацией из файла на внешнем носителе; вывод содержимого отношений ЛБД или содержимого файла запросов на внешнее устройство; выполнение запроса набранного на экране или сформированного ранее в файле; инициация обмена БАРД с программами пользователя; выход из системы.

Возможность выполнения конкретной команды пользователя зависит от его права доступа к необходимой информации ЛБД.

СУБД БАРД предоставляет возможность доступа к ЛБД из программ пользователя, составленных на мнемокоде, Фортране и Паскале. Для этого программисту предоставляются средства обмена с БАРД, которые собраны в библиотеке модулей обмена. Эти средства позволяют использовать программу пользователя и БАРД в качестве сопрограмм. Обмен информацией осуществляется через псевдоустройство межзадачного обмена информацией (МЗОИ).

БАРД и задачи пользователя могут работать в разных разделах оперативной памяти машины, а могут быть скомпонованы в один загрузочный модуль и работать в одном разделе, если позволит размер оперативной памяти раздела.

Программа пользователя может работать с несколькими БАРД, число которых неограничено. В свою очередь, БАРД может работать с несколькими программами пользователя. Обмен с каждой из программ производится по своему каналу МЗОИ. Одновременно могут быть задействованы до 10 каналов. Для каждого канала определяется приоритет (от 1 до 10, разным каналам может быть указан один приоритет). При обращении к БАРД нескольких задач обмен будет производиться с задачей, имеющей меньшее значение приоритета канала МЗОИ.

Библиотека модулей обмена представляет собой набор процедур, реализующих конкретные функции передачи и приема данных.

Перечень модулей и порядок их использования определяются заданным в программе пользователя режимом работы БАРД.

Реализация запросов из программ пользователя осуществляется с помощью файла запросов, заранее подготовленного сервисными средствами БАРД. В одном файле может находиться до десяти запросов. Задав режим исполнения запроса, с помощью модулей обмена в БАРД передаются имя файла с запросом, номер запроса и файлы и, если есть в этом запросе программные переменные, определяются фактические имена отношений, атрибутов и значения констант. После этого, если необходимо, организуется прием результатов выполнения запроса. Данные принимаются покортежно с помощью соответствующего модуля обмена.

Пользователю предоставляется также возможность чтения и записи кортежей в отношения ЛБД, минуя запросную систему БАРД. Это существенно ускоряет процесс обмена. Чтение кортежей отношений может производиться как последовательно, так и по указанному номеру кортежа. Номера кортежей могут следовать в произвольном порядке.

Из программы пользователя можно инициировать некоторые сервисные средства СУБД БАРД: генерация схем отношений, используя файл с заранее подготовленной информацией; удаление отношений; «сжатие» отношений; копирование отношения в файл для программ печати отчетов.

При возникновении фатальных ошибок, не позволяющих продолжить дальнейшую работу, БАРД выдает соответствующее диагностическое сообщение на экран терминала и код ошибки передает в задачу пользователя. Из задачи пользователя может быть отключена выдача сообщений на экран терминала.

С точки зрения ОС ПО СУБД БАРД включает задачу создания и ведения ЛБД, генератор системной БД, задачу обеспечения коллективного доступа к данным, задачу получения справочной информации об очереди к данным и генератор программ печати отчетов (ГППО).

Все задачи, составляющие ПО БАРД, являются диск-резидентными. Задача создания и ведения БД сегментирована.

Каждый пользователь работает со своей «копией» системы создания и ведения БД. При этом возможна одновременная работа нескольких пользователей с данными одной или разных ЛБД, которая может повлечь нарушение непротиворечивости данных. Для избежания возможных коллизий при работе служит задача обеспечения коллективного доступа.

В режиме коллективного доступа с одной или несколькими ЛБД могут работать до десяти пользователей одновременно.

Режим коллективного доступа задается при вызове задачи создания и ведения ЛБД и запускается задача обеспечения коллективного доступа. Обмен между этими задачами производится по двум каналам МЗОИ. И при вызове задачи создания и ведения ЛБД каждому пользователю отводится элемент очереди к данным. В дальнейшем каждая операция пользователя над данными санк-



ционируется только в том случае, когда с этими данными не выполняется операция типа «запись». В противном случае производится временная задержка выполнения задания и пользователю сообщается о необходимости ожидания освобождения данных.

С помощью задачи получения справочной информации об очереди к данным можно индентифицировать текущее состояние очереди. Эта же задача необходима для принудительного освобождения элементов очереди к данным, оставшимся занятыми после аварийного завершения выполнения задачи создания и ведения ЛБД.

ГППО предназначен для автоматизированного построения программ печати выходных форм, необходимых пользователю БАРД.

На основании текста, составленного пользователем на входном языке ГППО, генерируется ПАСКАЛЬ-программа, пригодная для последующей компиляции и компоновки средствами системы программирования ПАСКАЛЬ ДОС АСПО СМ ЭВМ и осуществляющая печать отчета.

Программа ГППО обеспечивает следующие возможности: генерацию на основании текста на входном языке ГППО ПАСКАЛЬ-программы, осуществляющей печать соответствующего отчета; выдачу протокола генерации в форме листинга; выдачу сообщений об ошибках в исходном тексте.

Сгенерированные средствами ГППО программы печати отчетов являются системно независимыми по отношению к БАРД и работают с входной информацией, скопированной из БД в файл ДОС АСПО, с помощью сервисных средств БАРД. Этот файл может быть подготовлен пользователем и другим способом.

## Применение

Система БАРД предназначена для поддержания реляционной модели данных в автоматизиро-

ванных системах различного назначения, функционирующих в реальном масштабе времени и построенных на базе технических средств СМ ЭВМ архитектурной линии СМ-1/2.

Функционирование системы БАРД поддерживается многозадачной ДОС АСПО СМ ЭВМ. Система БАРД не накладывает ограничений на режимы функционирования ДОС АСПО. ОС, поддерживающая работу БАРД в режиме коллективного доступа и доступа к данным из задач пользователя, должна включать группу инициативных устройств МЗОИ. Два канала МЗОИ для обеспечения коллективного доступа и по одному каналу для каждой из задач пользователя работающих одновременно с БАРД. Для выбора команд из меню системы и управления курсором при работе редактора системы используются специальные (технологические) клавиши терминала.

Минимальный набор технических средств для функционирования БАРД включает: процессор архитектурной линии СМ-1/СМ-2; оперативную память объемом 32 Кслов; устройство внешней памяти на магнитных дисках типа А322-3; устройство печати типа А521-5 или А522-7; видеотерминал типа ДМ 2000 или ВТА 2000; устройство времени (таймер).

Программная система выполнена на языке программирования ПАСКАЛЬ с незначительными вставками на мнемокоде.

Оперативная память, занимаемая задачей создания и ведения ЛБД, не превышает 20 Кслов.

Возможности СУБД БАРД позволяют эффективно ее использовать в АСУ технологическими процессами (где не требуется жесткое ограничение на время реакции), научным экспериментом, а также в непромышленной сфере — в различных информационно-поисковых системах и в системах деловых применений.



# СОДЕРЖАНИЕ

	Стр.
Введение	1
Операционные системы	2
Диалоговая единая мобильная операционная система для 32-разрядных малых ЭВМ СМ 1700 (ДЕМОС 32.2)	—
Диалоговая единая мобильная операционная система для 32-разрядных малых ЭВМ СМ 1702 (ДЕМОС 1702)	4
Диалоговая единая мобильная операционная система для ЭВМ СМ 1820 (ДЕМОС 1820)	—
Диалоговая многотерминальная система разделения времени для решения информационно-логических задач ДИАМС-3 версии 3 (ДИАМС 3.3)	5
Диалоговая система разделения времени для ЭВМ класса «Электроника-85» (ДИАМС 85.1)	8
Языки и системы программирования	—
Языки программирования операционной системы МОС ВП	—
Система программирования МОДУЛА-2 — средство разработки встроенных систем	14
Инструментальные средства программирования	16
Инструментальная система программирования ТУРБО, ПАСКАЛЬ (СП ТУРБО, ПАСКАЛЬ)	—
Инструментальные средства автоматизации разработки программного обеспечения для СМ 1810 (ИНТЕРПРО 1810)	20
Системы управления базами данных (СУБД)	24
Базовая система для создания и ведения в реальном масштабе времени локальных баз данных реляционного типа на СМ ЭВМ (СУБД БАРС)	—
ИНТЕРЕАЛ — мобильная СУБД с интерфейсом реляционного типа	27
Система управления реляционными базами данных для СМ 1810 (СМ 1820) (ДАТАРЕАЛ)	29
Интегрированные средства обработки данных для СМ 1810 (СМ 1820) (ИНТЕРСОД)	32
Система управления базами данных реляционного типа, функционирующая под управлением дисковых операционных систем АСПО СМ ЭВМ (БАРД ДОС АСПО)	36

Редактор Т. И. Петрова

Техн. редактор О. А. Овечкина

Корректоры: М. И. Батищева, В. А. Агеева

---

Сдано в набор 19.06.89. Подп. в печать 05.10.89. Формат 60×90<sup>1</sup>/<sub>8</sub>. Бумага типогр. № 2. Гарнитура литературная. Печать высокая. Усл. печ. л. 5,0. Уч.-изд. л. 6,51. Тираж 10000. Заказ № 1636. Изд. № ГСП-159. Цена 1 р.

---

Всесоюзный научно-исследовательский институт информации  
и экономики (ИНФОРМПРИБОР)  
125877, ГСП, Москва, А-252, Чапаевский пер., 14

---

Типография ВНИИТЭМР, г. Щербинка



