

**БИБЛИОТЕЧКА
ПРОГРАММИСТА**



В. Ю. ДЬЯКОНОВ
И. А. КАЛИНЧЕВ
В. А. КИТОВ

**Программное
обеспечение
систем
телеобработки
данных**



БИБЛИОТЕЧКА
ПРОГРАММИСТА

В.Ю.ДЬЯКОНОВ.
И.А.КАЛИНЧЕВ
В.А.КИТОВ

ПРОГРАММНОЕ
ОБЕСПЕЧЕНИЕ
СИСТЕМ
ТЕЛЕОБРАБОТКИ
ДААННЫХ



МОСКВА «НАУКА»
ГЛАВНАЯ РЕДАКЦИЯ
ФИЗИКО-МАТЕМАТИЧЕСКОЙ ЛИТЕРАТУРЫ 1592

ББК 22.18
Д93
УДК 519.68

Серия «Библиотечка программиста»
основана в 1968 году

Дьяконов В. Ю., Калинин И. А., Китов В. А. Программное обеспечение систем телеобработки данных.— М.: Наука. Гл. ред. физ.-мат. лит., 1992.— 160 с. — (Библиотечка программиста).— ISBN 5-02-014443-6.

Описываются возможности аппаратуры передачи данных. Приводятся примеры эксплуатируемых мониторов телеобработки данных. Особое внимание уделяется разработанной авторами мультитерминальной системе распределенной обработки данных ОБЬ, а также функциональным системам СОИР, КОНТРОЛЬ, ФОРМАТОР, ТЕРМЕС, ПДТ, КДОМ, эксплуатируемых на многих ВЦ.

Для научных работников и программистов.

Ил. 5. Библиогр. 9 назв.

Д $\frac{1402060000}{053(02)-92}$ 118-92

©«НАУКА». Физматлит, 1992

ISBN 5-02-014443-6

ОГЛАВЛЕНИЕ

Введение	4
Глава 1.История развития средств телеобработки данных	8
Глава 2.Обеспечение ввода-вывода в ОС ЕС	14
2.1. Основные понятия ввода-вывода в ОС ЕС.....	14
2.2. Обеспечение ввода-вывода для диалоговых устройств	16
2.3. Базисный телекоммуникационный метод доступа.....	17
2.4. Общий телекоммуникационный метод доступа	18
Глава 3.Система телеуправления данными КАМА	20
3.1. Основные возможности системы КАМА	20
3.2. Взаимодействие компонентов системы КАМА.....	20
3.3. Прикладное программирование в системе КАМА.....	24
3.4. Работа оператора терминала в системе КАМА.....	29
Глава 4.Система коллективного доступа PRIMUS	32
Глава 5.Возможности системы ОБЬ.....	37
Глава 6.Принципы построения программного обеспечения системы ОБЬ	41
Глава 7.Технология работы пользователей за терминалами системы ОБЬ	56
Глава 8.Прикладное программирование в системе ОБЬ	65
Глава 9.Интерактивное взаимодействие с СУБД.....	78
9.1. СУБД ADABAS и диалоговая система NATURAL.....	78
9.2. Реализация специальных функций NATURAL.....	80
Глава 10.Инструментальные диалоговые программные средства.....	83
10.1. Редактирование текстов и отладка программ	83
10.2. Программа диалоговой проверки и корректировки данных, хранящихся на магнитных лентах	88
10.3. Программа проверки и корректировки информации, хранящейся на дисковых томах	89
10.4. Система автоматизированной подготовки текстовой информации	91
Глава 11.Система отображения готовых видеogramм	94
Глава 12.Диалоговая система просмотра и корректировки данных, организованных в виде последовательных файлов.....	103
Глава 13.Вариант диалоговой автоматизированной системы контроля исполнительской деятельности	107
Заключение	112
Список литературы	112

ВВЕДЕНИЕ

Важнейшими факторами, обеспечивающими повышение эффективности использования компьютерных систем в условиях современной научно-технической революции, является быстрая реализация новейших достижений в разработке средств электронно-вычислительной техники и соответствующих компонентов программного обеспечения, а также постоянное совершенствование на всех уровнях системы управления и на основе этого модернизация подсистем, составляющих весь комплекс решаемых задач. Основной тенденцией разработки комплексных автоматизированных управленческих систем является ориентация на обеспечение работы в реальном масштабе времени, использование безбумажной технологии обработки данных, диалоговые режимы взаимодействия конечных пользователей и ЭВМ.

В настоящее время периферийная техника ЭВМ становится все более дешевой и разнообразной. Поэтому применение ее для автоматизации трудоемких процессов сбора и переработки информации стало возможным на предприятиях любого масштаба независимо от характера производственного процесса. Использование систем телеобработки данных в управленческой деятельности становится реальным фактором повышения эффективности управления предприятием за счет оперативности контроля и управления, принятия более обоснованных решений, улучшения качества труда.

Системы телеобработки данных (СТД) представляют собой совокупность технических и программных средств. В состав технических средств входят мультиплексоры передачи данных (МПД), связные процессоры, модемы, каналы связи, телетайпы, дисплеи и т. д. К основным функциям программного обеспечения систем телеобработки данных относятся следующие: контроль и координация работы терминального и связного оборудования с сопряженными ЭВМ, диспетчеризация потоков передаваемых сообщений, редактирование вводимых (выводимых) данных, обеспечение независимости передачи данных от их обработки, обнаружение и исправление ошибок, организация очередей и т. д.

Использование режимов телеобработки данных позволяет по-новому подойти к решению проблемы взаимодействия человека и ЭВМ, принимать различные управленческие решения в реальном масштабе времени, что обеспечивает повышение эффективности использования систем автоматизированной переработки данных. Последнее достигается за счет значительного сокращения временного интервала между моментом ввода исходной информации и получением результатов; повышения актуальности, хранящейся в базе данных информации. Возможность оперативного получения пользователями информации по различным запросам и оперативного контроля исходной информации в режиме диалога наряду с возможностью обмена информацией по каналам связи между автоматизированными системами различных уровней управления не только улучшает информационное обслуживание пользователей, но позволяет также значительно сократить затраты машинного времени и бумаги на АЦПУ.

Непосредственно вслед за появлением дисплейной техники стали активно разрабатываться и использоваться программы и программные комплексы, предназначенные для отображения готовых видеogramм. При квалифицированной и оперативной подготовке видеogramм система их отображения может стать основой для формирования систем оперативного информирования предприятий. Для реализации подобных систем должны выполняться следующие условия:

— готовые к отображению видеogramмы (экранные страницы), хранящиеся в массиве, должны быть связанными в некоторые последовательные логические единицы (документы) и составлять в целом базу данных оперативного информирования руководства;

— управление терминалом для доступа к документам должно быть предельно простым, а время реакции системы — минимальным;

— массив машинных документов должен быть прост в обслуживании при минимальных затратах объемов памяти прямого доступа;

— доступ различных групп пользователей к документам и средствам их экранного формирования должен быть независимым.

Предпосылки реализации указанных условий и перехода к безбумажной технологии функционирования систем автоматизированной переработки информации были созданы

современным развитием терминальной техники и аппаратуры передачи данных. Внедрение этой технологии предполагает частичное или полное исключение из процесса обработки информации трудоемких операций с бумажными документами. К этим операциям, в первую очередь, относятся: первичная подготовка данных путем заполнения специальных бланков; последующий ввод этих данных в ЭВМ; корректирование с использованием перфокарт; выдача результирующей информации пользователям в виде распечаток и обмен этими распечатками между подразделениями предприятия.

Реализация безбумажной технологии обработки данных может вестись с использованием следующих общих принципов: информационной обеспеченности всех уровней управления, индивидуальной ориентированности информации, прямого диалога, глобальной доступности информационной базы, разграничения доступа к информации. Реализация принципа информационной обеспеченности всех уровней управления позволяет устранить информационный дефицит, существующий при традиционном пакетном способе обработки данных, при котором частая выдача подробных распечаток слишком громоздка и экономически и технически нецелесообразна. Реализация принципа индивидуальной ориентированности информации предполагает, что информация, предоставляемая на терминал пользователя, должна быть сжата, отфильтрована и, по возможности, соответствовать стилю его работы. Принцип прямого диалога означает, что для эффективной управленческой деятельности пользователи на всех уровнях автоматизированного управления должны получать информацию непосредственно от ЭВМ, минуя промежуточные звенья (программист, оператор, секретарь, работники вспомогательных служб и т. д.), следствием чего должно явиться повышение оперативности при принятии управленческих решений. Реализовать указанный режим практически возможно только в рамках безбумажной технологии при наличии программных средств, обеспечивающих язык диалога, легко доступный пользователю-непрограммисту. Суть принципа глобальной доступности информационной базы состоит в том, что программно-технические средства систем безбумажной технологии обработки данных должны, при необходимости, обеспечивать пользователям вышестоящих уровней доступ к информации пользователей низших уровней. Это позволяет осуществлять проверку достоверности передаваемой информации и получать данные с любой степенью детализации, что повышает объективность принимаемых решений. Разграничение доступа к управленческой информации означает выделение для каждого пользователя определенного информационного спектра, соответствующего его функциям и уровню компетенции и исключение возможности несанкционированного доступа к другим разделам базы данных. Несоблюдение данного принципа может привести к неконтрольному тиражированию или умышленному искажению информации.

Режимы диалогового взаимодействия между пользователями и ЭВМ и сетевого взаимодействия между удаленными ЭВМ, обеспечивающими обработку информации на различных уровнях управления, являются основными в системах безбумажной технологии обработки данных.

Пользователи систем оперативного информирования, как правило, не являются специалистами в области вычислительной техники, и поэтому весьма важна простота диалога между пользователями и системой. Различаются следующие типы диалога: активный, полуактивный и пассивный. При активном диалоге предполагается, что пользователь хорошо знает системные команды и атрибуты информационных запросов. Он непосредственно сам, пользуясь клавиатурой терминала, формирует любой требуемый информационный запрос. При полуактивном диалоге система выдает на терминал наименования атрибутов запроса, а пользователь задает их значения и последовательность выдачи ответа. В результате диалога с пользователем система осуществляет формирование информационного запроса из введенных атрибутов и начинает его обработку. Пассивный диалог предполагает предоставление пользователю определенного списка значений атрибутов запроса. Пользователь идентифицирует нужный атрибут, например, используя фотоселектор или отмечая его специальным символом, или курсором. Для облегчения диалога используются специальные команды-подсказки, помогающие ориентироваться в системных командах и атрибутах запросов. В ряде информационных систем используются диалоговые языки, позволяющие

адаптировать информационную систему к стилю работы пользователей, имеющих различный опыт работы за терминалом.

Большое значение при создании диалоговых систем оперативного информирования имеет организация информационной базы и обеспечение возможности взаимодействия в реальном масштабе времени. Основная совокупность получаемой пользователями информации образуется в результате поиска информации в базе данных по фиксированным или произвольно сформулированным запросам, или же представляется в виде фиксированных видеogramм из архивных массивов системы. Перечисленные три режима функционирования отличаются друг от друга методами поиска информации, что отражается прежде всего на времени реакции системы. Очевидно, что наибольшее время ответа будет в режиме формирования произвольно сформулированных запросов, а наименьшее — в режиме выдачи фиксированных, предварительно сформированных дисплейных видеogramм. Поэтому более гибкими и эффективными будут те системы, в которых осуществляется совмещение режимов, так как в этом случае у пользователя будет возможность как оперативного доступа к уже подготовленной информации, так и возможность получения различных вариантов, являющихся результатом многоаспектного поиска. Использование режима выдачи предварительно сформированных видеogramм для оперативного информирования имеет как отрицательные, так и положительные стороны. К недостаткам его применения можно отнести то, что в ряде применений дублируются одни и те же данные, что увеличивает объем архива видеogramм. Однако фиксированные видеogramмы идентичны образцам документов, с которыми работает управленческий персонал. Эти формы соответствуют стилю работы конкретных пользователей, а привычное расположение атрибутов в видеogramме экономит время просмотра интересующей информации. Кроме того, по сравнению с запросом произвольной формы, запрос на выдачу нужной видеogramмы осуществляется простым набором ее имени, что не требует затрат времени пользователя на формирование запросов. Анализ диалоговых систем показывает большое разнообразие способов ведения диалога. К числу основных из них относятся: посылка простых запросов, использование мнемонических кодов, ведение диалога на языке, близком к естественному, последовательный просмотр записей, метод «телефонного справочника». Способ посылки простых запросов не предполагает организации дерева диалога. Использование мнемонических кодов служит для упрощения процесса диалога в случаях использования сложных составных запросов. Способ мнемонических сокращений целесообразен в случае, если их легко запомнить. Он позволяет избежать ошибок пользователя, возникающих при наборе параметров запроса. В то же время при большом количестве вариантов мнемоник пользователи либо путают мнемонические сокращения, либо просто их забывают. В последнем случае полезным является организация служебного перечня всех действующих мнемонических сокращений, выдаваемого при необходимости на экран дисплея по специальному запросу. Использование способа задания запросов на языке, близком к естественному, предполагает наличие определенной подготовки со стороны пользователей. В некоторых случаях пользователи могут получать неправильные ответы, потому что употребляют слово или фразу, которая не содержится в словаре ЭВМ. Иногда приходится слышать мнение, что инициатором диалога всегда выступает пользователь, т. е. человек посылает запрос, а ЭВМ отвечает. Однако часто эффективней оказывается последовательность действий, когда ЭВМ начинает диалог, а человек отвечает.

Внедрение систем телеобработки данных в автоматизированных системах можно условно разделить на три этапа:

—выдача сводок на терминалы пользователей, подготовленных при решении задач в пакетном режиме и не предполагающих сложную обработку больших массивов информации (например, автоматизированный контроль исполнения приказов);

—сбор и накопление оперативных данных о ходе производства, корректировка базы данных в режиме телеобработки, реализация поисковых запросов к базе данных с выдачей информации на терминалы;

—повсеместное применение диалоговых методов в управлении предприятиями, реализация сетей ЭВМ и межмашинного обмена информацией между автоматизированными системами переработки данных различных уровней и назначений.

Задачи первого этапа достаточно просты для внедрения, но они не могут оказать существенного влияния на качество системы управления и эффективность ее функционирования. Задачи второго этапа связаны с обработкой больших объемов информации в диалоговом режиме и требуют организации информационной базы по принципу банка данных. Задачи третьего этапа предполагают широкое внедрение локальных, региональных, отраслевых и других сетей ЭВМ; систем распределенной обработки данных, вычислительных центров коллективного пользования.

Имеющиеся в настоящий момент технические и программные средства отечественного производства позволяют решать задачи первого, второго и третьего этапов. Недостатком имеющегося технического обеспечения автоматизированных систем является разнородность используемых средств, причем не только в масштабах каждой из отраслей, но зачастую даже в рамках отдельных предприятий. Здесь имеется ряд проблем, связанных с совместимостью программ телеобработки данных с системами управления базами данных (СУБД); использованием этих средств с различными типами терминалов; нехваткой внешних запоминающих устройств для совместного функционирования режима телеобработки, СУБД и прикладных программ; отсутствием достаточного количества квалифицированных специалистов на предприятиях.

Настоящая книга посвящена рассмотрению вопросов разработки и практического использования универсальных программных комплексов систем телеобработки данных, ориентированных на эксплуатацию на вычислительных средствах ЭВМ Единой Системы. Основная часть материала книги содержит описание оригинальных программных комплексов, разработанных при непосредственном участии авторов — пакеты программ ОБЬ, ФОРМАТОР, ТЕРМЕС, ЛИСТЕР, КОНТРОЛЬ, ПДТ, ПЛТ. Комплекс КДОМ является модернизированным вариантом системы диалоговой отладки программ КДО. Весь материал книги можно условно разделить на три части. В первой части (гл. 1,2) кратко рассмотрены общие положения технологии безбумажного информирования пользователей ЭВМ; этапы истории создания систем телеобработки данных; характеристики современных технических средств теледоступа; назначение и основные функции систем сетевого взаимодействия между удаленными вычислительными комплексами, в том числе локальных вычислительных сетей. Вторая часть книги (гл. 3—9) начинается с описания возможностей распространенных в нашей стране базовых мониторов телеобработки данных КАМА и ПРИМУС. Значительное внимание уделено описанию архитектуры, функциям и технологии эксплуатации относящегося к классу мульти-терминальных систем распределенной обработки данных телемонитора ОБЬ, который внедрен на более чем двухстах пятидесяти предприятиях СССР и включен в экспортный фонд НПО Центрпрограммсистем. Заключительная часть книги (гл. 10—13) посвящена описанию функциональных пакетов программ телеобработки данных.

ИСТОРИЯ РАЗВИТИЯ СРЕДСТВ ТЕЛЕОБРАБОТКИ ДАННЫХ

Значение мультитерминальных систем распределенной обработки данных в автоматизированных системах переработки информации велико. Этот факт отмечается практически во всех работах последнего пятнадцатилетия, посвященных проблеме создания автоматизированных систем обработки данных (АСОД), например [1,4—9]. Повышенная роль, которую играют мультитерминальные системы, объясняется тем, что функционирование АСОД и информационно-справочных систем тесно связано с необходимостью повышения оперативности передачи и обработки информации. Причем ее источники и потребители могут находиться на значительных расстояниях друг от друга. Это обстоятельство требует не столько простого увеличения в автоматизированных системах использования средств вычислительной техники и связи, которые должны быть отнесены к числу важнейших компонент технической базы таких систем, сколько создания новых форм и методов совместного использования этих средств, обеспечивающих максимально благоприятные условия взаимодействия источников и потребителей информации вне зависимости от их удаления друг от друга.

Автоматизация процессов управления и информационного обеспечения в значительной степени повышает их эффективность. Благодаря использованию ЭВМ увеличивается число и объем решаемых задач. Однако при организации и проектировании автоматизированных систем с учетом тенденций их развития значительно изменяется характер такого использования. В частности, при тех требованиях, которые предъявляются в настоящее время к ЭВМ, может довольно быстро наступить момент, когда одна ЭВМ даже высокой производительности окажется не в состоянии решать все задачи, стоящие перед конкретной автоматизированной системой. Очевидна необходимость создания многопроцессорных и многомашинных комплексов. С другой стороны, к системе могут быть предъявлены такие требования, которые вызовут необходимость географического распределения вычислительных мощностей.

Коллективное использование ЭВМ является важным фактором, определяющим прогресс в области вычислительной техники. Причем техническая и экономическая целесообразность этой формы использования особенно заметно стала проявляться после интеграции средств, создания методов и оборудования телеобработки данных. При этом появились дополнительные возможности организации взаимодействия абонент — ЭВМ, расширились функциональные возможности самой ЭВМ, изменился характер ее применения в автоматизированных системах.

Системы телеобработки данных уже имеют свою историю. В 1941 г. впервые информация с телеграфной перфоленты была перенесена на перфокарты для ввода в вычислительную машину. В 1954 г. с помощью трансивера (терминала, подключаемого к телефонной линии) данные с перфокарт стали передавать непосредственно в машину. «Взрыв» в области дистанционного использования ЭВМ приходится на конец 50-х годов. Так, например, была создана система противовоздушной обороны SAGE [4, 7] в США, предназначенная для сбора преобразованной в цифровую форму радиолокационной информации, поступавшей с разнесенных на сотни километров пунктов, и последующей ее передачей десяткам вычислительных центров. Эти центры были объединены в полуавтоматическую общегосударственную сеть, протяженность линий связи которой составляла 2 млн. км, охватывая линии интерактивных дисплейных пультов. В 1962 г. начала действовать первая крупная система резервирования авиабилетов — работающая в режиме реального времени при непосредственном участии ЭВМ (в режиме ONLINE) система SABRE [4, 7]. В этой системе 1200 специализированных терминалов были связаны с главным центром обработки данных.

С тех пор наблюдается непрерывный прогресс в области телеобработки. Убедительным подтверждением тому служат: все увеличивающееся число пользователей, вводящих задачи в ЭВМ и получающих результаты, находясь на значительном расстоянии от процессоров; накопление огромного объема информации в базах данных, к которым имеют доступ сотни и даже тысячи людей; развитие диалоговых систем, позволяющих использовать удаленные вычислительные средства в интерактивном режиме для разработки и отладки новых программ

или процедур. Происходит постоянное совершенствование и расширение возможностей дистанционного использования рассредоточенных ресурсов для обработки данных.

В зависимости от класса задач, решаемых АСОД с помощью средств телеобработки, мультитерминальные распределенные системы, входящие в качестве подсистем этих АСОД, можно классифицировать следующим образом [7]: системы с диалоговым режимом работы; системы типа «запрос-ответ»; системы сбора данных; системы с пакетным режимом работы; системы с межпрограммной связью; телеметрические системы.

Указанные системы имеют различные характеристики, и к их архитектуре (спецификации взаимоотношений между частями системы) предъявляются различные требования.

Системы с диалоговым режимом работы. Эти системы характеризуются обработкой связанных друг с другом последовательностей достаточно коротких сообщений. Интенсивность трафика (направленного потока информации, передаваемого по каналу связи) примерно одна и та же в обоих направлениях. Нормальным режимом работы является ожидание ответа на каждое сообщение. Одним из примеров систем такого типа являются системы для отладки программ в диалоговом режиме. Некоторые простые системы типа «запрос-ответ» по режиму работы также приближаются к диалоговым. Типично диалоговой является и система резервирования мест на авиалиниях. Длина сообщения при этом обычно не превышает 100 символов. Поскольку длина сообщения мала и объем обработки, необходимый для получения каждого ответа, относительно невелик, накладные расходы как в системе передачи данных, так и в системе обработки данных должны быть низкими. Другими словами, скорость выполнения транзакций (последовательности сообщений в одном или обоих направлениях, которые вместе составляют законченную единицу работы) в значительной степени лимитируется этими накладными расходами. В системах с диалоговым режимом работы основным требованием является малое общее время реакции системы. Для поддержания достаточно высокого уровня активности человека за пультом время реакции системы должно быть не больше 5 с.

Следует отметить, что системы с диалоговым режимом работы в настоящее время составляют большую часть систем телеобработки. Эти системы находят большое применение как в АСОД (например, система ЭКСПРЕС для резервирования железнодорожных билетов в г. Москве), так и в автоматизированных системах ВЦКП (вычислительных центрах коллективного пользования)

Системы типа «запрос-ответ». Для этих систем характерны более длинные ответы на короткие запросы и большая независимость одного запроса от другого. Обычно ответ в четыре-пять раз длиннее запроса, и это соотношение может быть намного больше. В этом случае говорят, что трафик не сбалансирован. Запрос иногда бывает довольно сложным, что видно из следующих примеров: «НАЙТИ ВСЕ ЗАПИСИ ФАЙЛА ПО СХЕМА 1, У КОТОРЫХ ПОДРАЗДЕЛЕНИЕ РАВНО 0000003 И ДОЛЖНОСТЬ = 003 , ВЫВЕСТИ СПРАВКУ, В КОТОРОЙ СТРОКА СОДЕРЖИТ ЛИЧНЫЙ НОМЕР, ИМЯ, ДАТА РОЖДЕНИЯ»; «НАЙТИ ВСЕ ЗАПИСИ ПО СХЕМА1 И ВЫВЕСТИ ГИСТОГРАММУ ДЛЯ ПОДРАЗДЕЛЕНИЕ ».

В системах типа «запрос-ответ» количество обращений к памяти больше, а средства, предназначенные для обработки данных, более развиты. Следовательно, накладные расходы и время, затрачиваемые собственно на передачу данных, составляют меньшую долю общих накладных расходов и времени обработки. В данном случае важно, чтобы время реакции системы находилось в диапазоне от 5 до 30 с

Системы сбора данных. Ввод и сбор данных часто характеризуется относительно длинными входными сообщениями и очень короткими ответами. Объем обработки в ЭВМ в пересчете на одно сообщение минимален, но он может включать в себя проверки правильности, редактирование и преобразование форматов.

Системы с пакетным режимом работы. Эти системы характеризуются удаленным вводом задач и выводом больших объемов информации в одном или нескольких удаленных пунктах. Вводимая информация может быть ограничена параметрами, используемыми программой, уже ранее помещенной в память ЭВМ центрального узла сети. В других случаях объектом ввода является сама программа, которая затем обрабатывает данные, хранимые в центральном узле. Наконец, как программа, так и данные могут быть введены с удаленного пункта. Объем вводимой и выводимой информации может быть большим. Общее время, отсчитываемое с

момента ввода задачи и до момента получения результатов и измеряемое минутами или даже часами, является характерной величиной времени реакции систем с пакетным режимом.

Системы с межпрограммной связью. В этих системах на обоих концах линии связи имеются достаточные вычислительные ресурсы, и передача данных происходит под управлением специальных комплексов обслуживающих программ. Применение подобных систем расширяется по мере снижения стоимости средств обработки данных и реализации все большего числа функций. Им иногда присущи некоторые характеристики диалогового режима, режима «запрос-ответ» или пакетного режима в зависимости от того, как распределены функции между двумя процессорами. Требования к средствам автоматического обнаружения ошибок и восстановления работоспособности при сбоях весьма высоки и должны выполняться при минимальном вмешательстве оператора.

Телеметрические системы. Основное требование к таким системам — это малое время ответа, измеряемое с того момента, когда датчик или измерительное устройство выдает запрос на обслуживание процессором, и до момента — после того как процессор закончит обработку, связанную с запросом,— когда весь ответ поступит в исполнительное устройство. Сообщение может состоять всего из нескольких байтов и содержать данные от единственного датчика или из сотен байтов от многих датчиков. Поскольку доступ к таким запоминающим устройствам, как магнитные ленты или диски, связан с механическими перемещениями и требует много времени, информация должна храниться в высокоскоростных запоминающих устройствах. Для того чтобы общее время реакции измерялось миллисекундами, время, необходимое для прерывания процессора и его переключение на обработку запроса, должно быть весьма малым, и следует использовать высокоскоростные линии передачи данных.

Требования к программному обеспечению систем телеобработки данных включают в себя возможность обслуживания большого числа разнородных терминалов, обеспечение высокой пропускной способности потока запросов и ответов и малого времени реакции системы, а также, как правило, жесткие ограничения на используемый объем оперативной памяти ЭВМ. Сочетание этих противоречивых требований делает проектирование и реализацию мультитерминальных систем весьма сложной задачей.

Наиболее удачная классификация проблем, которые связаны с программированием систем телеобработки приводится в [4]. Несмотря на значительный срок, прошедший с момента появления этой работы, актуальность изложенных в ней положений до сих пор сохраняется. Перечислим эти проблемы с учетом сегодняшнего состояния дел: динамическое планирование, распределение приоритетов, мультиобслуживание, очереди, перегрузки, управляющие программы, прикладные программы, СВЯЗЬ с оператором, высокая надежность, функционирование при наличии неисправности, отладка программ, управление развитием программ.

Динамическое планирование. В отличие от систем, работающих в пакетном режиме, который может быть подробно спланирован и рассчитан по времени программистом, в системах с непосредственной связью с пользователем дело обстоит по-другому. Сообщения в систему поступают в случайные моменты времени и имеют разнообразный характер. Как следствие этого они требуют индивидуального обслуживания, различных программ для своего выполнения. Поэтому операции к системе выполняются в незапланированной последовательности, что требует динамического планирования работы, т. е. изменения расписания выполнения работ в зависимости от изменяющихся требований вместо того, чтобы придерживаться фиксированного расписания, как это было бы в случае обычной пакетной обработки. В отличие от пакетной обработки последовательность работ может изменяться каждую секунду. Таким образом, в мультитерминальных системах необходимо планировать непредсказуемую последовательность операций так, чтобы информация обрабатывалась за минимальное время и возможности ЭВМ использовались как можно полнее. Это является функцией управляющей программы. Разработка и написание такой программы намного труднее, чем обычных программ обработки данных.

Распределение приоритетов. Вполне вероятно, что различные пользователи, работающие с терминалами системы, посылают сообщения в ЭВМ в один и тот же момент времени. В таком случае система имеет множество сообщений, оспаривающих право использования

центрального процессора одновременно, и эти сообщения могут быть различных типов. В мультитерминальных системах распределенной обработки данных требуется гибкая система управления приоритетами. Это связано с тем, что состав пользователей системы разнообразен. В него могут входить все группы управленческого персонала, а также непосредственные исполнители работ. Многоуровневая система приоритетов должна обеспечивать гибкий (по критерию времени) доступ к ресурсам.

Мультиобслуживание. Одновременная работа пользователей за терминалами системы требует параллельной обработки большого числа сообщений. Насколько эффективно осуществляется распараллеливание, настолько производительней становится работа всей системы в целом. Несмотря на то, что большое значение имеет мощность самой вычислительной установки, резервы в программировании здесь далеко не исчерпаны. Например, основной способ, предоставляемый БТМД ОС ЕС (базисный телекоммуникационный метод доступа) для одновременного обслуживания терминалов, заключается в управлении группой каналов связи с помощью одного блока управления данными (DCB). Тем самым, все терминалы, подключаемые к ЭВМ посредством каналов связи, входящих в эту группу, обслуживаются последовательно, т.е. пока не закончится цикл взаимодействия с одним терминалом связь с другими не производится. Поэтому имеет значение порядок подключения терминалов внутри группы. Время реакции на действия пользователя первого терминала отличается — иногда значительно — от времени реакции на действия пользователя последнего терминала в группе. В настоящее время именно такой подход применяется в системах телеобработки на ЕС ЭВМ. КАМА, СУИП, КВАНТ, PRIMUS, КДО, ЛЕС — вот далеко не полный перечень известных систем, работающих по этому принципу. Неравномерность обслуживания терминалов — особенно когда их количество велико — отмечается всеми пользователями этих систем. Проблема может быть решена программным путем — см. гл. 6, в которой изложены принципы взаимодействия компонентов системы ОБЬ.

Очереди. В большинстве систем телеобработки скорость, с которой поступают сообщения, меняется во времени. Вполне вероятно, что несколько сообщений поступят в ЭВМ в одно и то же время. Они не могут быть обработаны немедленно, и поэтому будут образовываться очереди. Для систем телеобработки данных, как правило, возникает необходимость в организации выходных очередей, а в общем случае и других очередей сообщений. Имеются разнообразные методы обработки очередей. Длина очередей является величиной вероятностной: в один момент они будут содержать много элементов, а в другое время они могут быть небольшими. В существующих системах очереди обычно находятся в оперативной памяти ЭВМ, но так бывает не всегда. Например, в ОТМД ОС ЕС (общем телекоммуникационном методе доступа) могут использоваться три способа построения очередей сообщений: только в оперативной памяти; только на внешнем запоминающем устройстве прямого доступа; как в оперативной памяти, так и на внешнем устройстве. Обслуживание очередей является функцией управляющей программы, и принятый в ней алгоритм обслуживания очередей сильно влияет на работоспособность системы.

Перегрузки. Так как поступление задач или сообщений является случайным, то система время от времени будет оказываться перегруженной. Если случится так, что все пользователи используют свои терминалы одновременно, очереди новых сообщений могут значительно вырасти. ЭВМ будет опасно не хватать оперативной памяти или, возможно, времени на обработку. Каналам ввода-вывода или механизмам выбора дорожек на дисках может быть задано слишком много работы, что приведет к созданию чрезмерных очередей к этим устройствам. Возможно, что ЭВМ начнет обработку нескольких сообщений и окажется не в состоянии завершить их, потому что попытка сделать это вызовет перегрузку. Перегрузки также являются обычным, хотя и нежелательным явлением во многих системах обработки информации, работающих в интерактивном режиме. Существуют различные методы управления при перегрузках в системах телеобработки. Один из возможных методов состоит в изменении состава обслуживаемых системой ресурсов в процессе работы. То есть оператор имеет возможность уменьшить количество ресурсов, управляемых системой. Это позволяет снизить нагрузку.

Управляющие программы. При обсуждении всех этих проблем становится очевидным, что на

управляющие программы в системах телеобработки ложится выполнение большого объема работы. Они, действительно, являются сложными программами, и в некоторых работающих системах на их создание было затрачено время порядка 20 человеко-лет. Очень важно, чтобы управляющие программы были хорошо спланированы и правильно составлены. Иногда специальные публикации посвящаются технологии быстрой разработки программной системы, где внимание уделяется не самой системе, а процессу ее написания. Во многом функционирование системы зависит от квалификации разработчиков. Знание структурного подхода и принципа модульности является необходимым условием для программиста, участвующего в разработке управляющей программы. Методы, положенные в основу разработки мультитерминальной системы распределенной обработки данных ОБД [2, 3], позволили сократить время написания управляющей программы до 10 человеко-лет.

Прикладные программы. Несмотря на всю важность управляющей программы основную целевую функцию в системе телеобработки несут прикладные программы, которые в широком смысле соответствуют обычным программам обработки данных в вычислительных системах пакетной обработки. В мультитерминальных системах распределенной обработки данных в качестве прикладных программ могут выступать системы телеобработки разных классов: диалоговые, запросно-ответные и др. Прикладные программы обычно составляются совершенно отдельно от управляющей программы, поэтому возникает проблема интерфейса с ней.

Связь с оператором ЭВМ. Системы телеобработки должны проектироваться настолько автоматическими и независимыми от оператора, насколько это возможно. Однако неизбежны ситуации, когда требуется помощь оператора, и эта помощь должна оказываться со всей возможной быстротой и эффективностью. Определение ситуации, требующей вмешательства оператора, является важной функцией управляющей программы.

Высокая надежность. Для систем телеобработки данных необходимо, чтобы они были более надежными, чем пакетные вычислительные системы. Для этого имеются две причины.

Первая: более высокая степень автоматизации. Сообщения, поступающие в систему, обрабатываются и возвращаются на соответствующие терминалы без всякого вмешательства человека. Если в течение этого цикла появится ошибка, она не может быть определена до тех пор, пока не станет очевидной пользователю, работающему за клавиатурой терминала. Появление ошибки, которая искажает массивы внешних ЗУ (запоминающих устройств), может быть особенно катастрофичным, поскольку она может пройти незамеченной. Система обычно продолжает работать с поврежденными или неправильными записями. Вычислительная техника в настоящее время обладает такой надежностью, что обнаруживаемые ошибки, вызываемые отказами оборудования, являются очень редкими. Поэтому, когда встречаются ошибки в работе системы, они обычно являются ошибками программирования. Надежность программ, таким образом, является еще одной проблемой программирования для систем телеобработки. В программах должны предусматриваться проверки на правдоподобие.

Вторая причина, по которой необходима высокая надежность в системах телеобработки, вытекает из природы этих систем — прекращение функционирования ведет обычно к большим неприятностям — это особенно относится к телеметрическим системам. Ярким примером высокой надежности является использование вычислительных машин для обеспечения полетов человека в космос. В критические секунды после запуска корабля ЭВМ определяет, выйдет ли он на орбиту. Если нет, то немедленно должна быть произведена посадка в безопасной зоне.

Функционирование при наличии неисправности. Для мультитерминальных систем распределенной обработки данных необходимо, чтобы отказ части системы не приводил к выходу из строя всей системы. Если выходит из строя какое-либо периферийное оборудование, обслуживаемое системой, — например, терминал, — то система должна продолжать работать с другими устройствами. Это касается всех объектов управления системы, в том числе и прикладных программ. В распределенной среде отключение одной ЭВМ не должно повлечь за собой отказ других ЭВМ. Более того, в управляющей программе должно быть предусмотрено возобновление обслуживания объекта после его восстановления. Например, после ремонта терминала пользователь может продолжить работу в системе.

Отладка программ. Из-за высокого уровня надежности, который необходим в программах

телеобработки, очень важной является тщательная отладка программ. Однако при отладке программ для мультитерминальных систем возникает намного больше проблем, чем при отладке программ для пакетной обработки. Здесь необходимы более совершенные методы, многие из которых изложены в [1, 4]. Трудности отладки программ телеобработки вызваны несколькими факторами, отсутствующими в обычных системах. Прежде всего это связано с использованием терминалов и линий связи для ввода-вывода. Часто непрактично, а иногда и невозможно использовать реальные терминалы и линии связи для отладки программ. Поэтому устройства ввода-вывода должны моделироваться. Но проблема корректного моделирования не проста сама по себе. Так, для моделирования алгоритма межмашинного обмена СНХ1 (синхронный обмен по выделенным каналам связи) в системе ОБЬ потребовалось разработать имитатор, который в итоге по своей сложности превзошел модуль, осуществляющий передачу информации между ЭВМ по каналам связи СНХ1. Сложность универсальных систем телеобработки создает дополнительные трудности. Там, где много программ взаимосвязаны и должны составлять единое целое, необходимо тщательно отслеживать их совместную работу. Решения этих проблем, основанные на разработке программ, которые предназначены для помощи во время отладки системы, подробно описаны в [4]. Но эти программы должны быть готовы к моменту, когда они необходимы. Их разработка может занять значительное время. Поэтому желательно конструировать программное обеспечение таким образом, чтобы минимизировать интерфейсы между различными частями системы, так как именно в интерфейсах заложена наибольшая вероятность появления ошибок.

Управление развитием программ. Опыт создания систем телеобработки подтверждает, что отклонения внедряемой системы от первоначального проекта весьма значительны. Эти отклонения могут иногда привести к катастрофическим результатам. Поэтому в программное обеспечение следует закладывать возможность развития системы. Конечно, необходимы уже стандартные методы управления и контроля, основанные на высоком уровне документирования и тесной координации работ. Но эти методы могут оказаться недостаточными, если в само программное обеспечение не была заложена возможность развития и модификации.

Резюмируя, надо отметить следующие особенности, по которым программирование для мультитерминальных систем распределенной обработки информации является более сложным, чем обычное программирование: управление многочисленными операциями ввода-вывода; разнообразие типов сообщений и требований по их обработке; управление при случайных перегрузках; ввод сообщений с различными приоритетами; обработка очередей; мультиобработка; функционирование с ухудшенным качеством при наличии неисправности; контроль ошибок; сложные управляющие программы; взаимодействие управляющих и прикладных программ; трудности в отладке программ; повышенные требования к надежности; проблемы проектирования и использования моделирования.

В настоящее время является общепринятой следующая классификация программ, составляющих программное обеспечение управляющих систем телеобработки данных: прикладные программы; управляющие программы; сервисные программы.

В таких системах, как КАМА, КВАНТ, PRIMUS, ОБЬ, программное обеспечение четко подразделяется на эти категории.

Прикладные программы. Это программы, которые выполняют содержательную обработку сообщений. Они соответствуют программам обработки данных в обычных системах и уникальны для каждой из них. В мультитерминальных системах распределенной обработки данных в качестве прикладных программ могут выступать системы телеобработки разных классов: диалоговые, запросно-ответные, пакетные, сбора данных и др. Прикладные программы (системы) функционируют в среде, предоставляемой управляющей программой универсальной системы телеобработки. Иногда вместо термина «среда» вводят понятие «виртуальная машина», на которой выполняется программа. Виртуальная машина состоит частично из аппаратуры, а частично из программного обеспечения. Виртуальная машина, используемая прикладной программой, на самом деле является иерархической системой виртуальных машин. Первые два уровня иерархии, аппаратная вычислительная машина и операционная система, составляют виртуальную машину, обеспечивающую работу мультитерминальной системы распределенной обработки данных. Работу прикладных программ поддерживает трехуровневая

виртуальная машина — добавляется уровень виртуальной машины управляющей программы. Как правило, второй и третий уровни виртуальных машин сильно различаются между собой, что является серьезным неудобством при включении в систему телеобработки уже действующих прикладных программ, работающих под управлением операционной системы. Виртуальная машина, предоставляемая прикладным программам системами КАМА или СУИП — адаптациями американской системы CICS, — имеет мало общего с виртуальной машиной операционной системы ОС ЕС. Это, например, привело к тому, что для включения в качестве прикладной программы в эти системы СУБД СИОД, ориентированной для работы в ОС ЕС, пришлось переписать почти все модули СУБД (ППП ТЕЛЕСИОД и его последующая модификация ППП ТЕЛЕДОСТУП). Аналогичным образом обстояло дело с ППП ТЕЛЕСПРАВКА, где было разработано два варианта модулей доступа к информации в базах данных. Один вариант для работы на виртуальной машине ОС ЕС, другой — для работы на виртуальной машине КАМА. Перечень примеров мог бы быть и продолжен.

Одно из основных положений, на котором основывается проектирование системы ОБЬ, заключается в том, чтобы минимизировать различия между виртуальными машинами операционной системы и управляющей программы. Преимущества такого подхода очевидны:

- может быть обеспечена независимость программ прикладного и управляющего типов;
- упрощается процедура включения ранее разработанных программ в мультитерминальную систему распределенной обработки данных;
- отпадает необходимость изучения правил составления прикладных программ для виртуальной машины управляющей системы.

Средства, которые управляющая система обязана предоставить программам прикладного уровня, а именно доступ к объектам управления системы: терминалам, данным пользователя, устройствам документирования информации и др., должны быть максимально приближены по форме к аналогичным средствам, предоставляемым операционной системой.

Управляющие программы. Как уже было отмечено, управляющие; программы являются сложными программными системами, на разработку которых уходит много времени. О них было сказано выше.

Сервисные программы. В ряде случаев разработка сервисных программ может потребовать больше труда, чем создание управляющей системы. Список сервисных программ для универсальной системы телеобработки включает программы: настройки (генерации) и инициирования работы системы; возобновления работы; диагностики; обеспечения функционирования при наличии неисправностей; ведения архива системы; генерации данных для отладки; загрузки массивов; моделирования управляющей системы при отладке; средства отладки программ; средства распечатки памяти; распечатки информационных массивов.

Наличие в настоящее время мощных операционных систем позволяет значительно сузить круг сервисных программ, которые требуется разрабатывать. Многие необходимые программы уже имеются в составе операционной системы и могут использоваться для нужд системы телеобработки. Особое место в классе сервисных программ занимают программы настройки или генерации системы на конкретный состав обслуживаемых ресурсов. Сервисные программы настройки требуются не только системам телеобработки, но и необходимы системам других типов, например системам управления базами данных (СУБД). Если для СУБД пока не удастся обойтись без программ, осуществляющих ее генерацию, из-за особенностей архитектуры современных систем, то для систем телеобработки такие программы могут быть исключены. В системе ОБЬ не требуется сервисных программ генерации. Ее настройка на конкретный состав обслуживаемых ресурсов осуществляется автоматически, в момент инициирования работы. Тем самым значительно сокращаются сроки освоения и адаптации системы к конкретным применениям.

ГЛАВА 2

ОБЕСПЕЧЕНИЕ ВВОДА-ВЫВОДА В ОС ЕС

2.1. Основные понятия ввода-вывода в ОС ЕС

Проблемная программа, работающая в среде операционной системы ОС ЕС, непосредственно выполнением операций ввода-вывода не занимается. Это связано как с необходимостью обеспечить разделение доступа к данным (в связи с чем команды ввода-вывода являются привилегированными, т. е. требующими для своего исполнения состояния «Супервизор» в слове состояния программы PSW), так и с большим объемом сложного однообразного программирования выполнения операций ввода-вывода. Выполнение операций ввода-вывода в ОС ЕС осуществляется по запросам от проблемных программ к операционной системе. Совокупность программ системы, осуществляющих инициализацию и завершение программ канала, обработку прерываний от ввода-вывода, действия по исправлению ошибок ввода-вывода, отслеживание состояния внешних устройств и т. п., называется супервизором ввода-вывода.

Супервизор ввода-вывода является главной составной частью широкого класса программ операционной системы, объединяемых под названием Управление Данными. Программы Управления Данными обеспечивают логический интерфейс проблемной программы с внешними устройствами. Основные этапы обработки данных следующие:

—подготовка к обработке данных (OPEN) — программы Управления Данными проверяют санкционированность доступа к данным и настраивают системные управляющие блоки и таблицы для работы с данными на внешних устройствах;

—выполнение операций ввода-вывода — по запросам проблемной Программы Управление Данными строит программы канала, инициирует их выполнение супервизором ввода-вывода, определяет характер завершения операций ввода-вывода, оповещает проблемную программу об этом завершении;

—окончание обработки данных (CLOSE) — система управлений вводом-выводом логически отключается от проблемной программы.

Вся деятельность программ Управления Данными базируется на фундаментальном для ОС ЕС понятии метода доступа. Под методом доступа понимается совокупность метода организации данных и программного способа доступа к данным. В ОС ЕС используются три типа методов доступа:

- базисные методы доступа;
- методы доступа с очередями;
- метод доступа EXCP.

При использовании базисных методов доступа отпадает необходимость в написании канальных программ. Обмен данными осуществляется на уровне физических блоков, при этом программист должен в своей программе организовать управление буферами, объединение данных в блоки, выделение их из блоков и т. д. Например, базисный последовательный метод доступа BSAM характеризуется следующим:

—последовательной организацией данных — записи данных следуют одна за другой и порядковый номер записи является ее единственным поисковым признаком;

—программным способом доступа к последовательно организованным данным — с помощью функций метода доступа BSAM READ и WRITE проблемная программа может обрабатывать записи одну за другой.

Следует отметить такую важную характеристику метода доступа BSAM, как независимость от устройства. Последовательно организованные данные могут размещаться на любых устройствах единичных записей (перфокарточных, перфоленточных, печатающих) и устройствах внешней памяти (магнитных дисках, магнитных лентах). Однако в проблемной программе специфика внешнего устройства, вообще говоря, не отражается, и одна и та же программа может использоваться для обработки данных на различных устройствах.

Методы доступа с очередями позволяют обеспечить обмен данными на уровне логических записей. Все операции, связанные с буферизацией данных, объединением их в блоки и выделением из блоков, а также управлением очередями, выполняют программные средства методов доступа.

Несмотря на наличие развитых методов доступа ОС ЕС, в некоторых случаях может возникнуть необходимость непосредственного составления программы канала в проблемной программе. Эта необходимость возникает в следующих случаях:

—при желании программиста повысить эффективность обработки данных со стандартной для ОС ЕС организацией;

—при организации данных некоторым нестандартным способом;

—при обслуживании нестандартных внешних устройств.

Во всех этих случаях может быть применен специальный метод доступа EXCP. При использовании этого метода доступа в проблемной программе формируется программа канала, которая предоставляется для выполнения супервизору ввода-вывода. Обработка завершения канальной программы осуществляется супервизором и может быть продолжена и расширена проблемной программой. Все исключительные ситуации выполнения программы канала могут быть обработаны специальными программами-аппендиксами, которые представляют собой расширение супервизора ввода-вывода и включаются в систему системным программистом. Рассмотрим основные этапы выполнения операций ввода-вывода при использовании метода доступа EXCP.

— в проблемной программе формируется блок управления данными DCB, в котором указывается тип метода доступа, организация данных, имя оператора DD, способы обработки ошибок, имена аппендиксов;

— запускается функция Управления Данными OPEN, при этом по указанному блоку DCB проверяется санкционированность доступа к данным, загружаются требуемые программы-аппендиксы и строится защищенное от модификации проблемной программой расширение блока DCB — блок экстенгов данных DEB, в котором указываются границы внешней памяти на магнитном диске, адрес внешнего устройства, ключ защиты памяти;

— в проблемной программе формируется программа канала, составленная из любых команд канала, допустимых для данного внешнего устройства;

—выполняется функция EXCP: проблемная программа формирует и передает супервизору ввода-вывода блок ввода-вывода JOB, в котором указаны адреса канальной программы и блока DCB; после этого проблемная программа должна дождаться завершения операции ввода-вывода и проверить ее результаты, для этого в блок JOB копируется слово состояния канала CSW;

— выполняется системная функция CLOSE, разрушающая связь проблемной программы с супервизором ввода-вывода (уничтожается блок DEB, удаляются программы-аппендиксы).

Использование метода доступа EXCP делает практически невозможной независимость от устройства и требует от программиста достаточных знаний о функционировании конкретных устройств ввода-вывода. Тем не менее этот метод доступа часто применяется для программирования операций ввода-вывода, особенно для таких сложных в обслуживании устройств, как устройства прямого доступа и телеобработки.

2.2. Обеспечение ввода-вывода для диалоговых устройств

Рассмотрим некоторые вопросы программирования для устройств телеобработки, которые относятся к диалоговым внешним устройствам. Обеспечение диалога базируется на синхронизации действий пользователя терминала и ЭВМ, основанной на обработке сигналов внимания. При локальном подключении терминалов к ЭВМ (непосредственно к интерфейсу ввода-вывода или через контроллер) нажатие специальных клавиш на терминале приводит к возникновению прерываний от ввода-вывода. Эти прерывания могут интерпретироваться проблемной программой как признак готовности пользователя, т. е. разрешение ЭВМ производить операции ввода-вывода на терминале. Сами эти операции (писать, читать) достаточно стандартны и особенных трудностей в программировании не представляют.

Значительно сложнее ситуация при удаленном (через аппаратуру передачи данных) подключении терминала к ЭВМ. В этом случае нажатие клавиши «Внимание» не вызывает прерывания в ЭВМ, а только переводит терминал в некоторое специальное состояние. Для обнаружения этого состояния, а значит, и сигнала «Внимание», проблемная программа должна вести непрерывный опрос терминала с помощью выполнения специальных команд канала на аппаратуре передачи данных. Операции чтения и записи тоже осуществляются с помощью достаточно сложных канальных программ. При этом возникает большое число ошибок ввода-вывода, как на терминалах, так и в аппаратуре передачи данных. Предоставляемые

операционной системой телекоммуникационные методы доступа не всегда могут удовлетворительно обработать и исправить возникающие ошибки, что заставляет обращаться к методу доступа ЕХСР. В этом случае ориентированность программы на конкретный тип терминала позволяет наладить и более корректное исправление ошибок ввода-вывода.

Обычно под программным обеспечением систем телеобработки данных ОС ЕС понимают методы теледоступа, телемониторы (например: КАМА, PRIMUS, ОБЬ) и функциональные прикладные программы, ориентированные на решение конкретных терминальных задач.

Для обеспечения обмена информацией между ЭВМ и удаленными терминальными станциями ОС ЕС предоставляет в качестве базисного метода доступа БТМД и в качестве метода доступа с очередями — ОТМД.

2.3. Базисный телекоммуникационный метод доступа

БТМД предназначен для реализации функций обмена данными между ЭВМ и удаленными станциями как в стартстопном, так и в синхронном режимах передачи. При этом могут использоваться как коммутируемые, так и некоммутируемые каналы связи.

Программные средства БТМД обеспечивают выполнение следующих основных функций:

- идентификация системы телеобработки;
- ее активизация и деактивизация;
- перекодировка передаваемых данных;
- организация буферных пулов;
- управление каналами связи;
- неавтономное тестирование и исправление обнаруженных ошибок.

Во время генерации операционной системы необходимо описать технические средства системы телеобработки данных (типы мультиплексоров передачи данных, каналов передачи данных, удаленных станций), что обеспечивает при работе с описанными устройствами использование нужных модулей БТМД.

Если используемые каналы связи имеют сходные характеристики, то целесообразно объединить их в группы, что позволит использовать для их обслуживания один и тот же модуль БТМД. Группы каналов связи задаются либо во время генерации операционной системы при помощи макрокоманды UNITNAME, либо на этапе ее выполнения с помощью оператора DD.

При объединении каналов в группы должны соблюдаться следующие условия:

- синхронные и стартстопные каналы не могут находиться в одной группе;
- все каналы группы должны иметь одинаковую конфигурацию (например, для стартстопного режима они должны быть или коммутируемыми, или некоммутируемыми, а для синхронного режима все каналы должны быть либо некоммутируемыми односточными, либо некоммутируемыми многоточечными, либо коммутируемыми односточными).

Как известно, операционная система ОС ЕС строит для каждого набора данных блок управления данными DCB, в котором содержатся все сведения о наборе данных. Блок управления данными строится путем задания соответствующей макрокоманды DCB. Для того чтобы получить доступ к набору данных, блок DCB нужно открыть при помощи макрокоманды OPEN. При открытии набора данных в блок DCB заносится дополнительная информация из оператора DD, описывающего набор данных, и метки набора данных. Соответствие между набором данных, используемым в программе, и оператором DD, описывающим это набор данных, устанавливается при помощи блока DCB.

В системах телеобработки данных блок DCB строится для каждой группы каналов связи. Макрокоманда DCB позволяет определять тип канала связи, вид буферизации; в ней задаются процедуры исправления ошибок.

Для идентификации абонента каждой удаленной станции должен быть присвоен свой уникальный номер или идентификатор. Все идентификаторы станций заносятся в специальные абонентские списки. Для некоммутируемых каналов связи строится два типа списков: списки опроса и списки выборки. Для задания абонентского списка используется макрокоманда DFTRMLST. В случае неудачного открытия набора данных предусмотрена возможность повторной попытки открыть этот набор данных с помощью макрокоманды LOPEN. По завершении работы с группой каналов связи следует выдать макрокоманду CLOSE.

При передаче данных по каналам связи возникает необходимость в создании буферов. Буфером называется область оперативной памяти, временно резервируемая для хранения сообщений. В БТМД предусмотрены два способа организации буферизации: программный и динамический. При программной буферизации перед каждой операцией чтения необходимо запросить требуемое количество буферов, которые после завершения операции по приему сообщения необходимо освободить. При динамической буферизации буферы выбираются из буферного пула автоматически.

Построение буферного пула может осуществляться либо с помощью макрокоманд GETMAIN, BUILD и DCB, либо путем использования макрокоманд GETPOOL и DCB. Для получения буфера из буферного пула применяется макрокоманда REQBUF, а для его освобождения после выполнения операций чтения или записи — макрокоманда RELBUF.

В ЕС ЭВМ для передачи информации по каналам связи используются семибитовый код КОИ-7 и пятибитовый телеграфный код МТК-2. В связи с этим возникает необходимость преобразования информации из кода передачи во внутренний код ЭВМ ДКОИ. В БТМД для такого преобразования используются макрокоманды ASMTRTAB и TRNSLATE. С помощью макрокоманды ASMTRTAB задается тип таблицы перекодировки, соответствующей применяемым удаленным станциям. В случае использования нестандартных кодов в БТМД предусмотрена возможность организации новой таблицы. Посредством макрокоманды TRNSLATE осуществляется перевод из кода ЭВМ в код передачи. Для задания и приема сообщений от удаленных станций используются макрокоманды WRITE и READ.

При передаче сообщений посредством макрокоманды WRITE выполняются следующие операции:

- установить соединение и передать блок сообщения;
- передать следующий блок сообщения;
- передать блок сообщения и закончить передачу.

При приеме сообщений от терминальных станций при помощи макрокоманды READ можно:

- установить соединение и читать первый блок сообщения;
- передать подтверждение и читать следующий блок сообщения;
- передать отрицание приема, указывающее на необходимость чтения блока сообщения.

Задание различных типов макрокоманд WRITE и READ вызывает выполнение различных канальных программ. Для идентификации, установления соединения и окончания процесса передачи используются специальные служебные символы КТМ и КП.

2.4. Общий телекоммуникационный метод доступа

ОТМД обладает большим количеством функциональных возможностей, предоставляемых пользователям. В ОТМД наборами данных, используемыми обрабатывающими терминальными программами, являются очереди сообщений, полу чаемых или передаваемых терминальными станциями.

Управление сообщениями в ОТМД обеспечивает программа управления сообщениями ПУС, основными функциями которой являются:

- активизаций и отключение каналов связи;
- получение сообщения от терминалов;
- динамическая буферизация сообщений;
- приоритетное управление сообщениями;
- редактирование поступающих и выходных сообщений;
- постановка сообщений и ответов в соответствующие очереди;
- формирование сообщений об ошибках и т. п.

Рабочие варианты программы управления сообщениями создаются (генерируются) в зависимости от конкретных конфигураций системы телеобработки данных, имеющих в вычислительных центрах. Для создания ПУС существует набор специальных макрокоманд ОТМД.

Сообщение, передаваемое средствами ОТМД, состоит из заголовка и текста. Заголовок содержит служебную информацию, необходимую для правильной его обработки (приоритет, тип очереди и т. д.).

В ОТМД существует пять способов организации очередей сообщений:

1) очереди на однократно используемом накопителе на магнитных дисках — они заполняются, пока имеется доступная память, после чего выдается сообщение о переполнении и очередь необходимо переформатировать; сообщение хранится в очереди до тех пор, пока оно не будет выбрано;

2) очереди на повторно используемых магнитных дисках — очередь организована таким образом, что после заполнения доступной области памяти запись поступающих сообщений производится в начале очереди на место сообщений, прибывших в очередь первыми;

3) очереди в Оперативной памяти — при такой организации оперативность передачи информации резко возрастает, однако в этом случае становится экономически нецелесообразно обеспечивать хранение сообщений в течение длительного промежутка времени в связи с необходимостью использования больших объемов оперативной памяти;

4) очереди в оперативной памяти с регулярной перезаписью сообщений на однократно используемый МД;

5) очереди в оперативной памяти с перезаписью на повторно используемый МД.

Обычно программа управления сообщениями состоит из пяти секций: активизации, деактивизации, определения наборов данных, управления абонентскими пунктами и каналами связи, обработки сообщений и программ выходов пользователей.

Секция активизации ПУС включает следующие макрокоманды: INTRO, OPEN, READY. Макрокоманда INTRO является первой макрокомандой ПУС, ей передается сообщение от операционной системы. В макрокоманде INTRO указывается имя ПУС, определяются размеры буферов, обеспечивается включение средств отладки. Макрокоманда OPEN инициализирует наборы данных очередей сообщений. Макрокоманда READY является последней макрокомандой секции активизации ПУС.

Если в системе телеобработки данных применяются прикладные программы, то для завершения работы ПУС используется макрокоманда MCPCLOSE. Если прикладных программ нет, то для завершения работы ПУС выдается команда оператора SYSCLOSE.

Секция определения наборов данных состоит из макрокоманд, описывающих наборы данных групп каналов связи и очередей сообщений. Программная секция управления абонентскими пунктами и каналами связи обеспечивает создание таблиц, необходимых ПУС для управления терминальными станциями. Основу ПУС составляет обработчик сообщений (МН), являющийся набором программ, определяющих вид обработки сообщений, при помощи анализа управляющей информации, содержащейся в их заголовках. МН обеспечивает также редактирование входных и выходных сообщений, проверку кодов передачи, отправку сообщений и т. д.

Секция программ выходов пользователей является средством, предоставляющим пользователю возможность написать собственную терминальную программу, получающую управление от ПУС.

Обработку сообщений, поступающих от терминалов или посылаемых им, осуществляют прикладные терминальные программы, которые выполняются асинхронно по отношению к ПУС. При этом ПУС имеет более высокий приоритет. Очевидно, что если требуются различные виды обработки сообщений, то для каждого из них должна быть написана своя прикладная терминальная программа, управление которой будет передаваться после анализа заголовка сообщений.

Общий телекоммуникационный метод доступа предоставляет пользователям значительно большее количество возможностей, чем БТМД. Однако следует отметить, что при этом требуется расход значительных объемов оперативной памяти (до 200 Кбайт). Недостатком использования ОТМД является также необходимость регенерации ПУС в случае изменения конфигурации системы телеобработки данных.

Для управления разветвленными терминальными сетями и организации межмашинного обмена данными между удаленными вычислительными комплексами более целесообразным, по сравнению с БТМД и ОТМД, является использование телемониторов.

СИСТЕМА ТЕЛЕУПРАВЛЕНИЯ ДАННЫМИ КАМА

3.1. Основные возможности системы КАМА

Система КАМА обеспечивает работу прикладных программ в режиме телеобработки и удаленный доступ к базам данных. Система позволяет организовать одновременное обслуживание в режиме телеобработки большого числа заданий, которые могут поступать в произвольные моменты времени с различных терминалов, а также обеспечить совместный доступ различных прикладных программ к данным.

Пользователем системы телеобработки данных на базе системы КАМА является оператор терминала, вводящий запрос на обработку и получающий на свой или на другие терминалы результаты обработки данных по этому запросу. Единицей работы в системе КАМА является транзакция. Выполнение транзакции связывается с созданием задачи для ее обработки. Задача обычно связывается с терминалом, с которого был введен запрос на обработку.

Система КАМА имеет статус обычного задания в операционной системе ОС ЕС В других разделах основной памяти параллельно могут выполняться другие задания. Однако программы системы КАМА обычно имеют высокий приоритет и возвращают управление ОС только после обслуживания всех транзакций. Все порождаемые транзакциями задачи выполняются в разделе системы КАМА. Система распределяет ресурсы между ними и осуществляет защиту памяти. Она обеспечивает одновременное выполнение нескольких транзакций различных абонентов и совмещение операций ввода-вывода с обработкой.

В отличие от работы в пакетном режиме бесконтрольное аварийное завершение задачи в разделе памяти системы КАМА приводит с неизбежностью к нарушению нормального функционирования системы и к потере части вводимых с терминалов данных. Для возобновления работы системы необходим ее повторный запуск.

Система КАМА совместно с управляющими программами ОС реализует следующие функции:

- опрос и анализ состояний линий связи;
- опрос терминалов и ввод-вывод сообщений;
- учет и планирование ресурсов;
- запуск задач по запросам с терминалов;
- выявление ошибок ввода-вывода;
- установление связи терминалов с программами;
- организация очереди запросов к ресурсам системы телеобработки данных;
- динамический вызов программ в основную память и удаление их после выполнения;
- выявление и устранение блокировок системы;
- выявление и устранение закливания программ.

Система КАМА находит свое применение в таких областях, как удаленный ввод заданий для организации пакетного режима; управляемый ввод-вывод в различных диалоговых системах и системах сбора данных; коммутация сообщений в различных справочных системах. На основе системы КАМА могут разрабатываться автоматизированные системы управления и информационные системы. Реализация этих возможностей осуществляется разработкой прикладных программ с помощью набора макрокоманд системы КАМА и генерацией (настройкой) системы.

Система КАМА является пакетом общего назначения и имеет модульную структуру. Настройка системы на конкретное применение позволяет включать или исключать отдельные функции и компоненты, расширять их, изменять. Конфигурация системы КАМА (терминалы, наборы данных, очереди), прикладные программы, транзакции составляют рабочую среду системы, которая описывается с помощью таблиц. Генерация системы включает как генерацию программных средств, так и генерацию таблиц.

3.2. Взаимодействие компонентов системы КАМА

Функциональные средства системы подразделяются на три группы: управляющие

программы; управляющие таблицы; служебные программы.

К управляющим программам системы КАМА относятся программы управления задачами, управления памятью, управления программами, управления программными прерываниями, управления временем, управления дампом, управления терминалами, управления файлами, управления транзитными данными, управления временной памятью.

Управляющие и служебные программы используют данные из системных управляющих таблиц, к ним относятся: таблица управления программами, таблица управляющих программ, таблица управления терминалами, таблица управления файлами, таблица инициализации системы.

Служебные программы предоставляют пользователю такие дополнительные возможности, как идентификация оператора терминала и обеспечение функции главного терминала для динамического управления системой. С их помощью оператор главного терминала может оптимизировать режим работы системы телеобработки данных, а именно:

- изменять состояние системы и значения параметров, от которых зависит режим работы;
- обслуживать другие терминалы системы;
- вести динамическую запись системной статистики;
- выявлять аварийные условия, не обрабатываемые операционной системой;
- динамически открывать и закрывать наборы данных пользователя при работе системы в реальном масштабе времени;
- прекращать работу системы.

Наборы данных, необходимые для функционирования системы КАМА, описываются и размещаются во внешней памяти центральной ЭВМ перед началом генерации системы и в процессе генерации. Эти наборы данных подразделяются на системные наборы данных и наборы данных пользователя. В состав системных наборов данных входят: библиотека загрузочных модулей системы, набор данных дампа, внутрисистемный набор данных, набор данных временной памяти.

В библиотеку загрузочных модулей системы КАМА входят все сгенерированные системные и прикладные программы, включая управляющую часть системы и некоторые таблицы. Блок управления программами вызывает необходимые модули из этой библиотеки в основную память.

Набор данных дампа является последовательным набором данных, в который программа управления дампом выводит дампы выполняемых задач.

Внутрисистемный набор данных и набор данных временной памяти используются для организации очереди данных.

С точки зрения пользователя работа системы КАМА представляет собой обмен данными между его терминалом и центральной ЭВМ. Пользователь может сам с помощью прикладных программ управлять форматом данных. Вводимые в систему данные преобразуются в код ДКОИ и передаются для обработки прикладным программам.

Система КАМА выполняет две основные функции: функцию управления и системного обслуживания и оперативном режиме и обработку, не относящуюся к оперативному режиму — пользовательскую функцию.

В оперативном режиме выполняются следующие функции: управление процессом обработки (супервизорные функции) и управление данными. К супервизорным относятся функции, выполняемые программами управления задачами, управления памятью, управления программами, управления программными прерываниями, управления временем, управления дампом. Функции управления данными реализуются программами управления терминалами, управления файлами, управления транзитными данными и управления временной памятью.

Большинство программ системы КАМА имеет дополнительные выходы, дающие возможность пользователю разрабатывать программы, расширяющие функции системы.

3.2.1. Функции управляющих программ. *Программа управления задачами КСР обеспечивает многозадачный режим работы системы. Порожденная транзакцией задача выполняется в соответствии с назначенным ей приоритетом. Ограничения на количество активных задач в системе связаны с размером доступной основной памяти и с максимальным числом одновременно выполняемых задач, задаваемым при генерации. К функциям программ*

управления задачами относятся:

- диспетчеризация задач;
- контроль максимального числа задач;
- выявление и устранение блокировки системы, связанной с перегрузкой выделенного раздела основной памяти;
- выявление и устранение заикливания задач;
- порождение задачи;
- завершение задачи;
- синхронизация задач;
- изменение приоритета;
- синхронизация ресурсов.

Программа управления памятью SCP контролирует динамическую часть основной памяти, выделенной системе КАМА. Память выделяется задаче или освобождается по макрокомандам, которые выдаются системными или прикладными программами. Программа SCP ставит в очередь и удаляет из нее запросы на выделение памяти, обеспечивает выделение памяти, ее инициализацию и освобождение.

Программа управления программами PCP осуществляет:

- динамический вызов и загрузку программ;
- удаление неиспользуемых программ из памяти при перегрузке памяти;
- интерфейс с прикладными программами на языках высокого уровня;
- связь между прикладными программами;
- нормальное и аварийное завершение исполнения программ.

Программа управления программными прерываниями PIP является обобщенным средством обработки прерываний, возникающих в разделе системы.

Программа управления временем осуществляет на основе задаваемых при инициализации системы интервалов времени выполнение следующих функций:

- выявление заикливания в прикладных программах;
- выявление и устранение самоблокировки системы вследствие перегрузки памяти;
- управление интервалами времени выхода из раздела системы при отсутствии готовых к выполнению задач;
- синхронизация выполняемых задач;
- получение текущего времени суток;
- автоматическое инициирование задач повремени.

Функции синхронизации задач, получения времени суток и автоматического инициирования задач выполняются по макрокомандам из прикладных программ. С помощью синхронизации задач организуются режимы ожидания и оповещения. Автоматическое инициирование задач обеспечивает выполнение прикладной программы в заданное время суток или по истечении заданного интервала времени.

Программа управления дампом DCP осуществляет вывод заданных областей памяти в системный набор данных по макрокоманде выдачи дампа. Имеется возможность использования двух наборов данных дампа.

Программа управления терминалами TCP осуществляет связь прикладных программ с терминалами. Она выполняет функции управления каналами связи и терминальными устройствами. Информация о каналах связи и терминалах задается в таблицах управления терминалами.

Прикладные программы могут связываться посредством макрокоманд с программой TCP для выполнения следующих функций:

- получения и освобождения памяти терминала;
- определения длины введенных данных;
- инициирования задач по запросу в автоматическом режиме;
- перекодировки данных;
- опроса и анализа состояния каналов связи;
- выполнения программ метода доступа;
- синхронизации выполнения задач;

- установления ошибок ввода-вывода;
- завершения ввода-вывода.

Программа управления файлами FCP обеспечивает управление базами данных, используя таблицы управления файлами и средства метода доступа. Программа FCP осуществляет следующие функции:

- прямой поиск в наборе данных;
- последовательный поиск (просмотр) данных в наборе;
- поиск данных по косвенной адресации;
- сегментирование записей и объединение сегментов в наборы данных;
- снятие блокировки с данных при прямом методе доступа;
- монопольное управление операциями корректировки и добавления записей;
- защиту от одновременной корректировки записи набора данных несколькими задачами.

При выполнении макрокоманд обслуживания файлов баз данных Программа FCP имеет приоритет запрашивающей прикладной программы. Поэтому после выполнения запроса управление передается следующей команде этой программы.

Программа управления транзитными данными осуществляет передачу и получение сообщений. В системе организуется очередь сообщений, которые определяют пункты назначения. Пункты назначения могут быть внутренними и внешними. Идентификаторы пунктов назначения описываются в таблице управления пунктами назначения DCT.

Программа управления транзитными данными выполняется как служебная программа системы с приоритетом запрашивающей задачи.

Для внутрисистемных пунктов назначения возможно автоматическое инициирование задачи. Инициирование задачи производится, когда количество записей в очереди достигает заданного для пункта назначения уровня. Это средство позволяет прикладной программе коммутировать сообщения с терминала на терминал.

Программа управления временной памятью TSP осуществляет функции, необходимые для временного хранения и поиска данных в основной памяти и на устройствах прямого доступа:

- получение временной памяти;
- размещение данных во временной памяти;
- освобождение временной памяти.

Использование временной памяти удобно при подготовке видеформатов для дисплеев.

3.2.2. Функции служебных программ системы КАМА. Служебные программы системы КАМА являются дополнительными обрабатываемыми программами. При работе системы в реальном масштабе времени они выполняют следующие функции:

- Включить-исключить — разграничение доступа и защита в системе;
- Главный терминал — управление с терминала работой системы;
- Супервизорный терминал — изменение режима работы терминала;
- Терминал оператора — управление режимом и статусом своего терминала;
- Системная статистика — сбор статистических данных;
- Аварийные условия в системе — анализ аварийных условий;
- Аварийные условия в терминале — анализ аварийных условий в терминальном оборудовании, каналах связи и режимах их работы;
- Завершение работы системы — контролируемое завершение работы;
- Процессоры ввода-вывода асинхронных заданий — группируют задания и связанные с ними данные в пакеты для их асинхронной обработки и вывода результатов;
- Трасса — слежение за состоянием основной памяти системы;
- Открыть-закрыть динамически — возможность динамически открывать и закрывать наборы данных в процессе работы системы в режиме реального времени;
- Управление — управление работой системы по времени суток.

Средства подготовки и обеспечения работы системы КАМА также относятся к дополнительным средствам. Это средства генерации и инициализации системы, процессор языка высокого уровня и программы управления дампом.

3.2.3. Основные характеристики системы КАМА. Система КАМА позволяет подключать, используя метод доступа БТМД, через некоммутируемые каналы связи следующие терминалы:

в режиме стартстопной передачи данных — абонентские пункты АП-1 (ЕС-8501, АП-61 (ЕС-8561), АП-63 (ЕС-8563) и АП-70 (ЕС-8570); в режиме с двойной синхронизацией — АП-11 (ЕС-8511).

Через коммутируемые каналы связи могут подключаться: в режиме стартстопной передачи данных — АП-1; в режиме с двойной синхронизацией — АП-11.

С помощью методов доступа ГМД или БТМД может подключаться локальный комплекс ЕС-7906. В поздних версиях системы существует возможность подключения локальных и удаленных комплексов ЕС-7920-и работы с методом доступа ОТМД.

Система КАМА функционирует под управлением операционной системы ОС ЕС (режимы MFT, MVT, SVS). Необходимы такие дополнительные средства ОС ЕС, как интервальный таймер и управление режимом многократного ожидания.

Система КАМА поставляется в виде исходных модулей на языке Ассемблер. Для исходных и генерируемых библиотек необходима память прямого доступа до 2,5 Мбайт. Минимальная основная память для системы с несколькими однотипными стартстопными терминалами составляет 64 Кбайт (режим MFT).

3.3. Прикладное программирование в системе КАМА

3.3.1. Основные требования к программам. При проектировании системы телеобработки данных определяются задачи, обеспечивающие функционирование системы. Каждая задача представляет собой выполнение одной или нескольких программ. Характеристики задач хранятся в таблице управления программами РСТ. Этими характеристиками являются: идентификатор задачи (код транзакции), приоритет, ключ защиты, статус задачи при возникновении ошибок на терминале и блокировании системы.

Задача может инициироваться следующими способами: передачей сода транзакции с терминала, вызываться системой, инициироваться в заданное время суток, вызываться из программы другой задачи.

Информация об обрабатываемых программах заносится в таблицу обрабатываемых программ РРТ. Одна и та же программа может входить в несколько задач, выполняемых независимо.

Для обращения к системным средствам в прикладные программы включаются макрокоманды системы КАМА, при выполнении которых происходит прерывание и управление передается требуемой управляющей программе. Чтобы программа могла параллельно использоваться в нескольких задачах, она должна быть квазиреентерабельной. Это означает, что перед прерыванием все команды и поля памяти, которые были изменены в ходе выполнения программы, должны приводиться в исходное состояние. Этого можно достичь, если все величины, зависящие от конкретного применения задачи, размещать в ее рабочей области или в специальных областях памяти, адреса которых при прерываниях сохраняются в рабочей области задачи.

Прикладные программы для обращения к системным программам используют макрокоманды системы КАМА. Система выделяет задачам области ввода-вывода и рабочие области, а программа при выполнении динамически связывается с этими областями.

Имя макрокоманды системы КАМА имеет вид DFHxx, где xx — код управляющей программы, которая вызывается макрокомандой. Действия, опрашиваемые макрокомандой, определяются операндом TYPE = <тип макрокоманды>. Например, макрокоманда

```
DFHSC TYPE = GETMAIN
```

запрашивает действия, аналогичные макрокоманде GETMAIN супервизора ОС ЕС.

Прикладная программа, работающая под управлением системы КАМА, должна отвечать следующим требованиям:

- она должна быть связана с системными областями памяти;
- промежуточные результаты, необходимые при возвращении к программе после прерывания, должны сохраняться в рабочей области задачи;
- должна осуществляться динамическая связь с областями ввода-вывода;
- программа может использовать только регистры общего назначения 0—11, 15 (адрес

точки входа в программу содержится в регистре 14);

—в прикладных программах на языках Кобол, ПЛ/1 нельзя использовать операторы ввода-вывода этих языков; вместо них следует кодировать макрокоманды системы КАМА;

—при программировании на ПЛ/1 нельзя использовать средства мультизадачности и обращения к внешним процедурам;

Каждая прикладная программа должна быть оформлена как главная процедура.

Система КАМА автоматически сохраняет и восстанавливает содержимое общих регистров при переходах к подпрограммам, прерываниях и возврате в программу.

3.3.2.Связь с системными областями памяти. Макрорасширения макрокоманд в системе КАМА содержат ссылки на поля системных областей памяти. Системные области памяти подразделяются на управляющие области, рабочие области и области ввода-вывода. Для обеспечения квазиреентерабельности все вводные и выводные данные находятся в динамически размещаемых областях памяти, адреса выделенных областей ввода-вывода хранятся в системной области.

Управляющие области используются системой, назначение полей данных строго определено. Разделение рабочих областей на поля определяется при разработке прикладной программы.

При инициировании системы КАМА выделяется общесистемная область памяти CSA, к которой имеют доступ все программы системы. В управляющей области содержатся данные, необходимые для работы системных программ, адреса управляющих программ и таблиц, общесистемные константы.

За областью CSA размещается общая рабочая область CWA, которая также выделяется при инициировании системы. Длина ее определяется при генерации системы.

При образовании задачи выделяется область управления задачей TCA. В полях этой области размещаются все характеристики задачи и управляющие переменные. За областью TCA располагается рабочая область задачи TWA. Размер области задается при генерации системных таблиц. Каждая прикладная программа разделяет область TWA на поля и устанавливает их имена. При образовании задачи адрес области управления задачей заносится в регистр 12. Поэтому в прикладной программе должна быть фиктивная секция, имеющая базовый регистр 12 и описывающая поля связи в TCA. Копирование такой секции выполняется командой

```
COPY DFHTCADS
```

после чего записываются команды DS, описывающие поля TWA.

3.3.3.Связь с управляющими программами. *Управление памятью.* Прикладная программа обычно принимает сообщения от терминалов, подготавливает ответные сообщения и выводит их на терминалы. Для вводного сообщения область памяти выделяется системой. Для размещения промежуточных результатов обработки вводного сообщения и для формирования выводного сообщения могут потребоваться отдельные области памяти. Специально выделенная область памяти, в которой формируется выводное сообщение, после его подготовки к выводу передается от задачи прикладной программы в программу управления терминалами. Область памяти, выделенная задаче системой для вводного сообщения, после обработки сообщения должна быть освобождена, иначе будет происходить быстрое заполнение памяти.

Запросы на области основной памяти и освобождение ранее запрошенных областей осуществляются макрокомандами управления памятью. Запрос на основную память осуществляет макрокоманда

```
DFHSC TYPE = GETMAIN
```

в которой задается требуемый объем памяти, назначение выделяемой памяти (ввод-вывод на терминал, транзитные данные, данные во временной памяти), код инициализации выделяемой памяти. Освобождение ранее полученной памяти осуществляется макрокомандой

```
DFHSC TYPE = FREEMAIN
```

Адрес освобождаемой области перед выдачей макрокоманды должен быть помещен в поле TCASCSA области управления задачей. При освобождении всей памяти, выделенной для

областей ввода-вывода данной задачи, используется операнд RELEASE = ALL и не требуется заполнять поле TCASCSA.

Управление задачами. При образовании задачи строится область управления задачей ТСА, которая включается в очередь готовых задач. Управление задачами в системе КАМА осуществляется управляющей программой DFHKCP. Функции управления задачами запрашиваются прикладными программами через макрокоманды.

Макрокоманда

DFHKC TYPE = ATTACH

образует новую задачу. В макрокоманде указываются код транзакции задачи и символический адрес области управления устройства, с которым связывается новая задача. По макрокоманде формируется область управления новой задачей ТСА. В соответствии с кодом транзакции находится описание задачи в таблице управления программами PCT и определяется имя первой выполняемой программы.

Приоритет задачи определяется суммой приоритета транзакции в таблице управления программами, приоритета связанного с задачей терминала и приоритета оператора, работающего на терминале. В ходе выполнения задачи ее приоритет можно изменить макрокомандой

DFHKC TYPE = SNAP

При необходимости можно вызвать прерывание активной задачи для активизации задачи более высокого приоритета с помощью макрокоманды

DFHKC TYPE = WAIT

Для выдачи запросов на монопольное использование общих ресурсов системы (областей памяти, устройств ввода-вывода и программ) применяется макрокоманда

DFHKC TYPE = ENQ

Ресурс идентифицируется либо адресом в памяти, либо символическим именем. Запрос на освобождение ресурса осуществляется макрокомандой

DFHKC TYPE = DEQ

Если ресурс, запрашиваемый задачей, занят, она приостанавливается и включается в очередь к этому ресурсу.

Использование макрокоманд типа ENQ и DEQ может приводить к взаимной блокировке задач, когда каждая бесконечно долго ждет ресурса, занимаемого другой задачей. В системе КАМА возможны два способа преодоления этого состояния. Первый способ заключается в том, что при генерации можно установить предельное время пребывания системы в состоянии блокировки. Второй способ — установление для каждой задачи статуса при состоянии блокировки, т. е. определяется, можно удалять задачу или нет. Изменить статус задачи можно и в процессе выполнения с помощью макрокоманды DFHKC с типом PURGE (задачу можно удалить в случае блокировки) или NOPURGE (удалять задачу запрещается).

Управление связями. Оно осуществляет динамический вызов программ и организацию межпрограммных связей. В системе КАМА функции управления связями осуществляются программой DFHPCP, которая вызывается макрокомандой DFHPC. Для динамического вызова программ применяется макрокоманда

DFHPC TYPE = UNK/XCTL

Если используется параметр LINK, то осуществляется переход к программе, находящейся на следующем, более низком уровне управления. По окончании работы вызываемой программы управление передается команде, следующей за макрокомандой DFHPC. Для перехода к программе, находящейся на том же уровне управления, что и вызывающая программа, используется параметр XSTE. В этом случае после выполнения макрокоманды DFHPC вызываемая программа может быть удалена из памяти, возврат управления переходит к программе более высокого уровня управления.

В вызываемой программе для возврата должна кодироваться макрокоманда

DFHPC TYPE = RETURN

Эта макрокоманда используется также для завершения задачи. Если за терминалом не закреплена постоянная задача в таблице управление терминалами, то при возврате управления можно изменить имя задачи; которая будет выполняться на этом терминале с помощью дополнительного параметра TRANSID.

Система КАМА обеспечивает через макрокоманды связь между программами только по передаче управления. Передача параметров в программу может быть осуществлена только через общую рабочую область CWA или через рабочую область задачи TWA.

Управление терминалами. Прикладная программа может выдать запрос на ввод данных с терминала или на вывод данных. Система строит очередь таких запросов.

При образовании задачи создается связь программы с определенным терминалом. Задача может быть связана только с одним терминалом. Обмен данными с терминалом осуществляется с помощью макрокоманды

DFHPC TYPE = (список функций)

В качестве функций могут использоваться следующие:

WRITE — вывод записи из области ввода-вывода терминала (TIOA) на терминал и разблокирование клавиатуры;

WRITEL — то же, что и WRITE, но без изменения состояния клавиатуры;

READ — ввод записи с терминала в выделенную системой или прикладной программой область TIOA (если перед этой операцией клавиатура была заблокирована, то она разблокируется);

READL — то же, что и READ, но после чтения сообщения клавиатура блокируется;

ERASE — используется совместно с функциями WRITE или WRITEL; очищает экран дисплея перед выводом данных и устанавливает курсор в начало первой строки;

WAIT — переводит программу в состояние ожидания до завершения операции ввода-вывода;

SAVE — по завершении операции ввода-вывода область TIOA сохраняется за прикладной программой;

DISCONNECT — разъединение коммутируемых каналов связи;

RESET — отключение от линии связи;

COPY — читается содержимое всего буфера терминала.

Некоторые функции часто используются в определенных комбинациях. Для таких комбинаций применяются специальные обозначения:

GET — (READ, WAIT)

PUT — (WRITE, WAIT)

CONVERSE — (WRITE, READ, WAIT)

PAGE — (ERASE, WRITE, READ)

Обмен информацией с устройствами комплекса ЕС-7920. Групповая дисплейная станция ЕС-7920 состоит из устройства управления и подключенных к нему алфавитно-цифровых дисплеев и печатающих устройств. Комплекс ЕС-7920 может подключаться к селекторному или мультиплексному каналу ЭВМ или соединяться с ЭВМ через мультиплексор передачи данных или телепроцессор. Для обмена данными в ОС ЕС используется базисный или общий телекоммуникационный метод доступа.

В системе КАМА существуют дополнительные базисные средства форматирования сообщений для терминалов комплекса ЕС-7920. При этом система КАМА вставляет управляющую информацию в выходные сообщения и позволяет иметь доступ непосредственно к полям данных для входных сообщений.

С помощью средств форматирования системы КАМА можно строить таблицы форматов для макетов входных и выходных сообщений, осуществлять доступ к полям данных в области ввода-вывода терминала. Возможность использования средств форматирования задается при генерации программы управления терминалами значением параметра BMS = YES.

Существует специальная макрокоманда форматирования ввода-вывода DFHBMS, которую можно использовать в прикладных программах вместо обычной макрокоманды управления терминалами.

При форматировании экран дисплея делится на поля, для каждого поля устанавливаются такие характеристики, как относительное расположение и длина поля, уровень яркости знаков, возможность указания пол световым пером, выравнивание данных в поле по левой или правой границе поля, защищенность поля от занесения новой информации (пропуск поля при перемещении курсора по экрану).

Таблица форматов задается с помощью последовательности макрокоманд DFHMDI и DFHMDF. После ассемблирования этих макрокоманд редактирования связей таблица записывается в библиотеку загрузочных модулей. Описание таблицы начинается макрокомандой DFHMDI, в качестве обязательных операндов используются TYPE, MODE, TERM. Тип таблицы определяется операндом TYPE, его значения могут быть MAP и DSECT. Если указано значение MAP, то создается таблица формата, включающая всю необходимую информацию, для записи в библиотеку загрузочных модулей. При значении DSECT строится фиктивная секция с описанием полей таблицы для последующего копирования в прикладную программу. Операнд MODE определяет назначение таблицы! MODE = IN — для входных данных и MODE = OUT — для выходных; Операнд TERM определяет тип терминала, его значение 7920.

Управление временной памятью. Она доступна всем задачам, выполняемым в системе КАМА. Существуют два уровня временной памяти: главная (первый уровень) и вспомогательная (второй уровень). Главная память выделяется в разделе основной памяти системы КАМА. Она используется для кратковременного хранения небольших записей длиной не более 256 байт. Вспомогательная память — это набор данных из записей фиксированной длины на устройстве прямого доступа. Этот набор данных при инициализации системы КАМА задается оператором DD с именем DFHTEMP.

Во временной памяти каждая запись имеет уникальное имя длиной до 8 символов, доступ к записям осуществляется по их именам. Обычно временная память используется для передачи информации в другие задачи, для временного хранения обрабатываемых данных.

Помещение данных во временную память осуществляется макрокомандой

```
DFHTS TYPE = PUT,STORFAC = MAIN/AUXILIARY
```

Здесь MAIN — память первого уровня (главная), AUXILIARY — память второго уровня (вспомогательная). Для данных длиной более 256 байт независимо от значения операнда STORFAC выделяется вспомогательная память.

Для получения данных из временной памяти используется макрокоманда

```
DFHTS TYPE = GET
```

для уничтожения — макрокоманда DFHTS TYPE = RELEASE

Управление временем. Программа управления временем системы КАМА позволяет прикладным программам получать значение времени суток в рабочее поле, переводить задачу в состояние ожидания до заданного момента времени, автоматически инициировать задачу в заданный момент времени. Обращение к программе управления временем осуществляется с помощью макрокоманды DFHIC.

Значениями операнда TYPE могут быть: GETTIME, WAIT, POST, INITIATE, PUT, CANCEL.

Значение TYPE = GETTIME используется для получения времени суток. При этом в команде можно задать дополнительные параметры: формат представления времени, адрес поля памяти, в которое помещается значение времени.

Значение TYPE = WAIT применяется, чтобы задержать выполнение задачи до истечения заданного интервала времени или до заданного времени суток. Для задания определенного интервала времени используется операнд INTRVAL, а для задания времени суток — операнд TIME.

Для синхронизации по времени выполнения двух или большего числа задач в ОС ЕС используется аппарат событий. Макрокоманда

DFHIC TYPE = POST

позволяет запланировать событие на конкретное время. При этом используются те же основные и дополнительные операнды, что и в случае TYPE = WAIT.

Задачу можно инициировать в заданный момент времени макрокомандой

DFHIC TYPE = INITIATE

используя операнды INTRVAL и TIME. Дополнительными операндами являются имя задачи (код транзакции) и имя терминала, с которым должна быть связана эта задача.

Значение TYPE = PUT используется при инициации задачи по времени с передачей ей группы данных через временную память. Дополнительно необходимо указать адрес памяти, по которому расположены передаваемые данные.

Макрокоманда

DFHIC TYPE = CANCEL

отменяет запросы к программе управления временем типа WAIT, POST,; PUT и INITIATE. При отмене запросов типа WAIT или POST система организует действия таким образом, как если бы ожидаемые события произошли и заданное время ожидания истекло. При отмене запросов типа INITIATE и PUT эти запросы удаляются из системы.

3.4. Работа оператора терминала в системе КАМА

3.4.1. Состояния терминала в системе. Терминалы в системе подразделяются на три группы. К первой группе относятся терминалы, которые постоянно связаны с определенной задачей. Код транзакции этой задачи записан в таблице управления терминалами. Любой ввод с терминала вызывает эту задачу, обращение к терминалу из других задач невозможно. За терминалами второй группы также закреплены определенные задачи, но к данному терминалу возможно обращение и из других задач. И наконец, к третьей группе относятся терминалы, за которыми не закреплена постоянная задача в таблице управления терминалами. Вводя код транзакции, оператор такого терминала может вызвать любую задачу.

При функционировании системы терминал может переводиться из одной группы в другую.

Терминал в системе имеет определенные состояния работы такие, как TRANSACTION (задача), TRANSCEIVE (приемопередача), RECEIVE (прием), INPUT (передача).

При состоянии работы TRANSACTION задачи инициируются только по вызову терминала. В этом случае терминал относится к первой или третьей группе.

При состоянии работы TRANSCEIVE задачи могут инициироваться как по вызову терминала, так и системой. Такое состояние возможно для всех трех групп терминалов.

Состояние работы RECEIVE разрешает только получение сообщений. Это состояние используется на терминалах второй группы при автоматическом инициировании задач.

Состояние INPUT разрешает только ввод сообщений и использует» па терминалах первой группы.

При генерации управляющих таблиц задается начальное состояние работы терминала. Оно может изменяться командами, посылаемыми с терминала, работающего в режиме главного терминала, или в результате работы системы.

Состояние терминала может быть активным (обслуживаемое) или пассивным (необслуживаемое). Терминал, находящийся в активном состоянии, выполняет ввод или вывод сообщений. Система не воспринимает сообщения, посылаемые с пассивного терминала, но если этот терминал имеет состояние работы TRANSCEIVE, то на него могут выводиться сообщения.

Для вызова задачи с терминала третьей группы, который имеет состояние TRANSACTION или TRANSCEIVE, необходимо передать с этого терминала инициирующее сообщение, в котором указываются код транзакции и данные. Для указанного вида терминала можно вызывать системные задачи, изменяющие состояние терминала, и запрашивать асинхронное выполнение задач.

Для проверки работы терминалов используется задача тестирования терминалов. С ее

помощью реализуются два режима проверки терминала: режим ввода всех символов терминала и режим дублирования вводных сообщений.

3.4.2. Оператор главного терминала. Главный, или супервизорный, терминал используется для оперативного управления процессом телеобработки. Главным терминалом может быть любой терминал со статусом TRANSACTION или TRANSCEIVE. Для перевода такого терминала в состояние главного терминала с него вводится код транзакции CSMT, инициирующий необходимую системную задачу.

Задача главного терминала осуществляет следующие функции управления:

- получение или изменение параметров таблицы инициализации системы;
- получение или изменение параметров таблицы управления терминалами;
- получение или изменение предельной длины очереди внутреннего пункта назначения;
- открытие и закрытие наборов данных для файлов баз данных или внешних пунктов назначения;
- работа с наборами данных для вывода дампов;
- управление средствами трассировки;
- принудительное завершение задачи любого терминала;
- получение характеристик прикладных программ;
- завершение работы системы.

Задача главного терминала при ее вызове выполняет какую-то одну запрашиваемую функцию, после чего она завершается.

3.4.3. Асинхронная обработка транзакций. Асинхронная пакетная обработка используется в системе КАМА для решения задач с большим объемом вводимых или выводимых данных или для задач, требующих значительного объема вычислений. Каждое задание в пакете заданий имеет свой код транзакции. Задания в пакете выполняются по мере освобождения требуемых ресурсов системы. По запросу с терминала результаты выполнения заданий, помещенные в выходной набор данных, могут выводиться на указанный в запросе терминал. Возможность такой фоновой асинхронной обработки транзакций повышает эффективность использования ЭВМ.

Любая прикладная или системная задача может выполняться в режиме асинхронной обработки. Для обеспечения этого режима при генерации системы КАМА задается макрокоманда

```
DFHSG TYPE = ATP
```

а в макрокоманде

```
DFHSG TYPE = INITIAL
```

должен быть задан параметр ATP = YES.

Для ввода пакета сообщений инициируется системная задача ввода с кодом транзакции CRDR:

```
CRDR NAME = имя, DELIM = ограничитель
```

В параметре NAME задается имя пакета. Ограничитель, определенный в параметре DELIM, используется для разделения отдельных заданий в пакете.

3.4.4. Сбор статистических данных. Для определения эффективности работы системы КАМА в конкретных условиях используются средства сбора статистики о запросах, поступающих в систему за определенные интервалы времени, о степени используемости программных и технических средств.

В прикладные программы могут быть включены операторы, которые посылают сведения о всех ключевых событиях в специальные наборы данных или другие пункты назначения. Существуют и специальные средства позволяющие накапливать значения определенных характеристик за время сеанса работы системы. К таким характеристикам относятся:

- общее время работы системы КАМА;
- реальное максимальное количество одновременно выполняемых задач;
- число реализованных запросов на получение областей памяти размер освобожденной

памяти;

- число вызовов транзакций;
- число вызовов каждой программы;
- количество выданных дампов;
- статистика обращений к файлам;
- статистика работы с пунктами назначения;
- статистика использования временной памяти.

Количество удовлетворенных запросов на получение и освобождение памяти показывает динамичность использования памяти. Если число запросов на получение памяти больше, чем на освобождение памяти, то это означает, что существуют дефектные прикладные программы, которые не освобождают память при завершении задачи.

В статистику обращений к файлам входит целый ряд характеристик число операций чтения, обновления и добавления записей, число записей, прочитанных из областей переполнения индексно-последовательных файлов, имена выбираемых сегментов для файлов с сегментированными записями.

К статистике работы с пунктами назначения относится количество входных и выходных записей для каждого пункта назначения, а также количество автоматически инициированных задач при достижении предельных длин очередей в пунктах назначения.

Вся системная статистическая информация по завершении работы системы КАМА помещается в системный пункт назначения CSSL в виде несблокированных записей переменной длины.

Для использования системных статистических средств при генерации системы КАМА необходимо включить в таблицу обрабатываемых программ PPT программы DFHSTKC, DFHSTTD и DFHSTTR. Задачу сбора статистики следует описать в управляющей таблице PCT с кодом транзакции CSTT.

Вызов задачи с терминала осуществляется сообщением

CSTT тип запроса, пункт назначения, тип информации

Тип запроса задает вариант получения статистической информации: AOR — получение полной информации, три пробела — получение сокращенной информации. В запросе можно указать несколько типов статистической информации.

3.4.5. Средства отладки. В системе КАМА имеются такие средства отладки, как моделирование терминалов устройствами с последовательным доступом, дампы областей основной памяти, трассировка.

При представлении терминалов устройствами с последовательным доступом терминал моделируется парой устройств, которая описывается как один терминал в таблице управления терминалами TCT. Моделирующей парой устройств могут быть устройство чтения с перфокарт и печатающее устройство, входной и выходной последовательные наборы данных на магнитных лентах и магнитных дисках. Вывод дампа памяти осуществляется по макрокоманде DFHDC. При этом содержимое областей памяти задачи выводится в последовательно организованный набор данных на магнитной ленте или на магнитном диске. Набор данных для вывода дампа должен быть описан в операторе DD с именем DFHDMPA при инициализации системы КАМА.

Вид макрокоманды управления дампом.

DFHDC TYPE = вид памяти, DMPCODE = идентификатор дампа

В операнде TYPE определяется вид области памяти, которая включается в дамп. Если включаются все области памяти, которые относятся к задаче, то используется значение TRANSACTION. Для дампа управляющих программ и таблиц используется значение КАМА. COMPLETE — выдача полного дампа, являющегося объединением двух предыдущих типов дампа. Если используется значение PARTIAL, то запрашивается дамп областей памяти, которые перечислены в дополнительном параметре LIST. В операнде LIST может быть указано:

TERMINAL — дамп областей ввода-вывода терминала (TIA);

PROGRAM — дамп области прикладных программ данной задачи;

TRANSACTION — дампы всех областей основной памяти задачи, связанных в цепочку, начало которой хранится в области TCA;

SEGMENT — дампы той области основной памяти, информация о которой помещена в области TCA перед выполнением макрокоманды.

Программа управления трассировкой фиксирует определенные события при работе системы КАМА в специальной таблице трассировки Включение и выключение программы управления трассировкой осуществляется макрокомандой DFHTR. Если задача аварийно завершается или выдается макрокоманда дампа DFHDC, таблица трассировки переписывается в набор данных дампа.

ГЛАВА 4

СИСТЕМА КОЛЛЕКТИВНОГО ДОСТУПА PRIMUS

Система PRIMUS является наиболее широко распространенным телемонитором, используемым на отечественных ЕС ЭВМ для организации коллективного доступа к ресурсам ЭВМ, работающих под управление ОС ЕС в режимах MVT и SVS. Ее популярности способствовал тот факт, что это первая универсальная диалоговая система для ЕС ЭВМ, разработанная в стране в то время, когда в качестве альтернативы были американские программы CICS, CRJE и TSO, которые с трудом адаптировались к отечественным вычислительным средствам. Требовалась собственная разработка, которая бы учитывала особенности функционирования ЕС ЭВМ и ее терминальных устройств.

Первая версия системы PRIMUS появилась в середине 70-х годов. В последующие годы она интенсивно развивалась, и в настоящее время по написанному в ее рамках прикладному программному обеспечению она превосходит все остальные системы данного класса, эксплуатируемые на ЕС ЭВМ в нашей стране.

В данной главе описываются основные сведения о назначении и возможностях версии 2.5 системы PRIMUS, взятые из программной документации, датированной 1985 г.

Основные функции системы заключаются в обеспечении одновременной и независимой работы группы пользователей, получающих доступ к ресурсам вычислительной системы через алфавитно-цифровые дисплеи. Она позволяет обслуживать одновременно до 256 абонентов. Каждый абонент, получивший доступ к системе, может использовать вычислительные ресурсы в режиме разделения времени или в пакетном режиме.

В режиме разделения времени пользователь может инициировать загрузку и выполнение произвольной программы. Она может быть выполнена в интерактивном режиме — при непосредственном взаимодействии с пользователем — путем распределения традиционных операций ввода-вывода на экран дисплея. Система также предоставляет специальные средства ввода-вывода, управляющие обменом информацией с дисплеем, предназначенные для написания диалоговых программ на различных языках программирования.

Пакетный режим работы обеспечивается стандартными средствами ОС. При этом система предоставляет возможность оформления и передачи заданий на выполнение в пакетном режиме, слежения за прохождением задания в ОС ЕС и получения на экран дисплея результатов его выполнения.

В состав системы входит стандартный набор диалоговых программ, называемый функциональными программами (ФП). Каждой ФП поставлена в соответствие команда, с помощью которой она может быть вызвана для выполнения. Полный перечень команд и соответствующих им ФП образует список команд системы. Команды предоставляют средства ввода и редактирования программ и данных, управления личными наборами данных, запуска программ. Этот список является открытым. Он допускает расширение со стороны пользователей, что дает возможность ориентировать стандартные средства системы на конкретную область применения с учетом специфики работы вычислительной установки.

Одной из основных концепций, лежащих в основе построения системы, является максимальное использование стандартных средств ОС ЕС. Это определяет ее относительную простоту и компактность.

Универсальная диалоговая система с коллективным доступом должна обеспечивать

независимую работу большого числа пользователей, создавая у них иллюзию монопольного использования ресурсов ЭВМ. В системе PRIMUS это достигается за счет предоставления каждому пользователю самостоятельного программного процесса, реализованного в форме подзадачи ОС ЕС. Распределение ресурсов центрального процессора (ЦП) между программными процессами пользователей достигается за счет стандартного средства квантования времени ЦП в ОС ЕС.

Система PRIMUS не имеет собственных средств разделения основной памяти. Эта проблема решается в SVS, обеспечивающей управление виртуальной памятью.

В рамках собственного программного процесса абонент может инициировать выполнение как диалоговой, так и вообще произвольной программы. На структуру программы и на средства их взаимодействия с ОС не накладывается никаких ограничений. Если программа является диалоговой, то для обеспечения ее взаимодействия с терминалом пользователя предоставляются специальные средства ввода-вывода, освобождающие разработчика программы от необходимости учитывать конкретные особенности устройства. Для произвольной программы имеется средство распределения операций ввода-вывода на терминал, базирующихся на методах доступа ОС ЕС BSAM или QSAM. Это дает возможность выполнять любые программы в интерактивном режиме.

В действительности программа взаимодействует не с физическим, а с некоторым виртуальным терминалом (VT), имеющим уникальный логический номер и поддерживаемым средствами ядра системы. Функции отображения VT на реальный терминал и обратно реализуются программами управления сообщениями (ПУС), являющимися частью ядра системы. Многообразие типов обслуживаемых терминалов может потребовать включения в ядро нескольких ПУС, каждая из которых обслуживает терминалы определенного типа.

По аналогии с терминалами система обеспечивает управление выходными потоками информации, направленными на различные устройства вывода и внешние программные процессы. Для этих целей система поддерживает виртуальные файлы вывода (ВФВ). Их обработка выполняется с помощью специальных программ управления выводом (ПУВ), которые, как и ПУС, входят в ядро системы, Набор ПУВ может пополняться с целью расширения класса обслуживаемых физических устройств и внешних программных процессов.

Для обеспечения гибких средств редактирования информации каждому пользователю, вступающему в контакт с системой, присваивается так называемый «рабочий набор данных» (РИД). Этот набор данных имеет специальную организацию и предназначен для промежуточного хранения информации абонента. РИД может быть также использован для обеспечения информационной связи между последовательно выполняющимися ФП.

Помимо РИД каждый пользователь имеет доступ к наборам данных ОС ЕС. Стандартный набор ФП предоставляет возможность создания, уничтожения и переименования наборов данных. Имеются также средства просмотра наборов данных и средства организации обмена информацией между ними и РИД. Потенциально каждый абонент может получить доступ к любому набору данных, расположенному на произвольном запоминающем устройстве. Подобная свобода действий не всегда является допустимой, поскольку может повлечь за собой конфликтные ситуации.

В связи с этим в состав системы включена необязательная подсистема управления доступом пользователей к ресурсам. Подсистема может быть активизирована в момент запуска PRIMUS. Она позволяет регламентировать доступ пользователей к системе в целом, к ее различным функциональным возможностям, доступным через команды, а также к наборам данных. Центральным элементом подсистемы является справочник. Он представляет собой специальный набор данных и содержит информацию о всех абонентах, имеющих право доступа к системе, а также о наборах данных, приписанных этим абонентам.

Подсистема управления доступом различает три категории абонентов: пользователь, администратор, системный программист. Они отличаются правами и возможностями, которые им предоставляет система.

Одновременная работа пользователей в режиме разделения времени требует специальных средств учета и контроля используемых ресурсов ЭВМ. В связи с этим в системе имеется подсистема сбора статистической информации. Она может быть активизирована в момент

запуска РКIMUS.

Подсистема сбора статистической информации позволяет отслеживать работу каждого абонента и накапливать собираемые данные на одном из внешних запоминающих устройств. В ее состав также входят программы автономной обработки накопленной информации и выдачи итоговых протоколов работы системы. Формат создаваемых статистических данных совпадает с форматом записей системной мониторинной программы (СМП) ОС ЕС. Это дает возможность использовать ее специальные наборы данных SYS1.MANX и SYS1.MANY для хранения и последующей обработки статистической информации о работе системы.

Система PRIMUS функционирует в рамках самостоятельного одношагового задания ОС ЕС. Она позволяет поддерживать связь с другими заданиями, которые выполняются как отдельные ФП или специальные мониторы ФП.

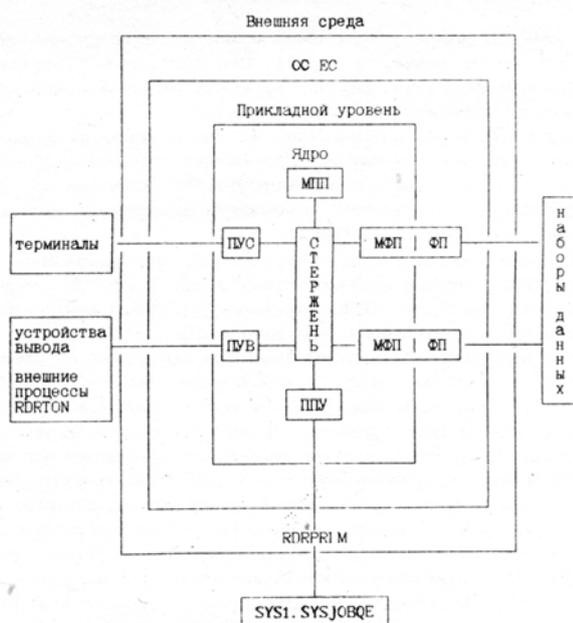


Рис. 1. Общая структура систем

Программное обеспечение системы представляет собой совокупность программных процессов, функционирующих как подзадачи ОС. Общая структура системы изображена на рис. 1. Четко разделяются два уровня программного обеспечения: ядро и прикладной уровень.

Ядро инициализируется в момент начальной загрузки системы и занимает постоянный объем памяти в течение всего времени выполнения. Размер ядра зависит от состава обслуживаемых терминалов и печатающих устройств, а также от наличия необязательных возможностей управления доступом пользователей и сбора статистической информации. Минимальный размер ядра составляет 40 Кбайтов.

Объем основной памяти, занимаемой прикладным уровнем системы, определяется размерами программ, вызываемых пользователями для выполнения. К числу таких программ относятся стандартные ФП, личные диалоговые программы (ДП), а также произвольные программы, выполняемые в режиме разделения времени. При обслуживании шестнадцати терминалов и выполнении только стандартных ФП размер прикладного уровня системы составляет около 150 Кбайтов.

Структура ядра системы. Ядро системы представляет собой совокупность программных процессов, выполняющих определенные функции. К их числу относятся управления: программными процессами всей системы; программами пользователей; терминалами; выходными потоками данных; заданиями пакетной обработки.

В ядро входят следующие компоненты: монитор программных процессов (МПП); монитор функциональных программ (МФП); программы управления сообщениями (ПУС); программы управления выводом информации (ПУВ); планировщик первого уровня (ППУ).

Для объединения параллельных процессов ядра в единую систему используется специальное средство синхронизации. Оно называется «стержень» и реализовано в виде дополнительного

программного процесса. Принцип синхронизации, реализуемой стержнем, основан на обмене сообщениями между программными процессами. Он состоит из набора структур данных, программных модулей и диспетчера, управляющего их выполнением. Стержень активизируется всякий раз, когда к нему обращается некоторый программный процесс. Цель обращения заключается в передаче сообщения с указанием пункта назначения. Результатом работы стержня является изменение состояния его структур данных, а также, возможно, регистрация некоторого события, приводящего к активизации другого программного процесса. Наличие в стержне механизма управления внутренними работами обеспечивает совмещение обработки нескольких сообщений, одновременно поступающих от различных программных процессов. Это позволяет устранить эффект влияния их друг на друга при взаимодействии со стержнем.

Главным процессом ядра является МПП. Он выполняется головной задачей шага задания и порождает остальные программные процессы, обрабатывает ситуации аварийного завершения подзадач и восстанавливает работоспособность процессов.

Все программные процессы ядра порождаются в момент инициализации системы. Процессы пользователей динамически создаются в момент начала сеансов работы и уничтожаются в конце сеансов.

Управление программами пользователей. В рамках программных процессов пользователей функционирует компонент ядра системы — МФП. Он является реентерабельной программой. В связи с этим одна копия МФП, загруженная в оперативную память, используется во всех программных процессах абонентов. В момент создания программного процесса ему приписывается виртуальный терминал (ВТ). МФП может вступать в диалоговое взаимодействие с пользователем путем применения стандартных средств обмена сообщениями с ВТ.

В начальный момент своей работы МФП вступает в контакт с пользователем в целях получения его имени и прав доступа к системе. Последняя функция выполняется лишь в случае, когда активизирована подсистема управления доступом. После успешного завершения фазы представления пользователя МФП переходит в режим интерпретации команд. После получения очередной команды МФП обращается к стандартному списку команд с целью определения имени соответствующей ФП. Если команда указана правильно, то загружается соответствующая программа в оперативную память и ей передается управление. Вызванная ФП может вступить в диалоговое взаимодействие с пользователем, используя интерфейс ВТ. Она может также потребовать создания виртуального выводного файла. Аварийное завершение ФП обрабатывается средствами МПП, который позволяет восстановить программный процесс пользователя.

Управление терминалами. Функции отображения виртуальных терминалов на физические устройства осуществляются программами управления сообщениями. ПУС выполняется в рамках самостоятельных программных процессов ядра.

В каждом варианте системы для конкретной вычислительной установки состав ПУС описывается с помощью специальной таблицы. Она используется в момент инициализации системы для загрузки в оперативную память соответствующих ПУС.

В состав стандартного программного обеспечения входят ПУС, написанные с помощью графического и базисного телекоммуникационного методов доступов. Каждая ПУС предназначена для обслуживания группы однотипных терминалов. Поддерживаются: локальные дисплейные станции ЕС-7906; локальные и удаленные комплексы ЕС-7920; дисплейные станции ЕС-7970, ЕС-7910; абонентские пункты ЕС-8564, ЕС-8563, ЕС-8562, ЕС-8534. Обеспечивается связь с ЭВМ ЕС-1010 и СМ-4

Интерфейс взаимодействия ПУС со стержнем формализован, что позволяет включать в систему новые программы управления сообщениями для других типов терминалов. Эти ПУС могут использовать любые средства ОС ЕС. В частности, допускается написание ПУС с применением общего телекоммуникационного метода доступа (ОТМД). В этом случае все многообразие программ управления сообщениями может быть заменено единственной ПУС, обслуживающей все типы терминалов. Однако большие затраты оперативной памяти, свойственные ОТМД, приводят к необходимости проведения специальных оценок целесообразности использования такой ПУС.

Управление выводными файлами. Под внешней по отношению к PRIMUS средой понимается совокупность программных и аппаратных средств, параллельно работающих на вычислительной установке.

Для ввода информации во внешнюю среду используются виртуальные файлы вывода. Они составляют выходные очереди. Каждая выходная очередь соответствует одному выходному классу системы PRIMUS.

Выходные классы обозначаются буквами латинского алфавита и обслуживаются программами управления выводом. Соответствие между классом и ПУВ устанавливается с помощью специальной таблицы в момент запуска системы.

Каждая ПУВ, в зависимости от своего назначения, может выводить данные из предписанного ей класса на конкретное устройство (АЦПУ, магнитную ленту, том прямого доступа) или некоторой программе, выполняющейся вне Системы PRIMUS. В последнем случае ПУВ используется как средство обмена информацией между системой и внешним программным процессом. Примером может служить передача с помощью ПУВ заданий пользователей программе системного вывода RDRTON.

ФП может потребовать построить для нее виртуальный файл вывода, который представляет собой динамически создаваемую цепочку последовательных наборов данных на устройстве прямого доступа. Во время работы ФП заполняет ВФВ и по завершению вывода указывает выходной класс, в который следует его направить. Затем соответствующая ПУВ выполнит вывод информации на конкретный носитель или передаст ее в другую программу. Если был задан класс «А», то ВФВ помещается в системную выходную очередь ОС ЕС с классом «R». Во всех других случаях используются внутренние очереди системы PRIMUS.

В программное обеспечение системы входят ПУВ, обеспечивающие работу стандартных АЦПУ и печатающих устройств ЕС-7934, входящих в состав локальных комплексов ЕС-7920. Допускается включение в систему дополнительных ПУВ, обслуживающих другие типы печатающих устройств. Общее число одновременно работающих ПУВ определяется в момент создания таблицы ПУВ.

Управление пакетными заданиями. В системе реализовано два способа запуска заданий на выполнение в пакетном режиме.

Первый способ предпочтительнее для вычислительных установок с малым объемом оперативной памяти. В этом случае заявки, поступающие от ФП на выполнение заданий в пакетном режиме, обрабатываются планировщиком первого уровня. ППУ является самостоятельным программным процессом ядра и обеспечивает передачу заданий, подготовленных пользователями, на выполнение в пакетном режиме под управлением ОС ЕС. Он поддерживает внутреннюю очередь заданий, называемую очередью первого уровня и представляющую собой специально организованный набор данных. Элементы очереди обрабатываются в порядке поступления. Задания, извлекаемые из очереди, передаются средствами ППУ планировщику заданий ОС ЕС путем запуска специальной процедуры системного ввода RDRPRIM. ППУ следит за тем, чтобы в системе одновременно выполнялось не более одной такой программы системного ввода.

ППУ не является обязательным компонентом ядра. Его наличие или отсутствие определяется в момент загрузки системы. В случае, когда он отсутствует, задания накапливаются в очереди первого уровня.

Второй способ запуска заданий более быстрый. Он основан на использовании специальной резидентной программы системного ввода RDRTON, выполняющейся в рамках самостоятельной системной процедуры. Подготовленные для выполнения задания образуют выходной поток ВФВ, направленных к программе RDRTON — внутренний класс вывода «H».

Система PRIMUS с точки зрения прикладного программиста. На прикладные программы в системе PRIMUS не накладывается никаких ограничений по использованию средств ОС ЕС. По способу организации взаимодействия с пользователем прикладные программы могут быть условно разделены на два класса: диалоговые и пакетные. Диалоговые программы рассчитаны на выполнение в режиме непосредственного взаимодействия с пользователем и характеризуются тем, что алгоритм их работы существенно зависит от вводимой информации, а содержание вводимой информации не может быть определено заранее и зависит от результатов

предыдущих взаимодействий пользователя с программой. Диалоговые программы могут быть выполнены только в интерактивном режиме. Пакетные программы отличаются тем, что содержание вводимой информации может быть с достаточной степенью вероятности определено до начала работы программы, потому они успешно работают в пакетном режиме. Однако в некоторых случаях целесообразно выполнять пакетные программы в интерактивном режиме, например при решении многовариантных задач, когда вмешательство человека может существенно сократить общее время решения. Система PRIMUS обладает средствами, позволяющими эффективно выполнять в интерактивном режиме оба указанных класса прикладных программ. Эти средства обеспечивают:

- распределение операций ввода-вывода, использующих стандартные методы доступа BSAM и QSAM, на экран дисплея;
- выполнение одношагового задания в режиме разделения времени;
- вывод информации во время работы системы;
- выполнение командных процедур.

В зависимости от класса прикладной программы ее выполнение может инициироваться с помощью команд LINK или RUN. Команда LINK предназначена для выполнения под управлением системы PRIMUS диалоговых программ. При вызове прикладной программы ей передаются параметры системы PRIMUS и обеспечиваются средства обмена информацией с экраном дисплея. В целях облегчения написания диалоговых программ существует набор макросредств для языков ассемблера и ПЛ/1, позволяющих осуществлять доступ к экрану и наборам данных пользователя. Выполнение в интерактивном режиме прикладных программ, предназначенных для пакета, может быть осуществлено с помощью команды RUN, которая обеспечивает загрузку в оперативную память и передачу управления указанному модулю. Вызванной программе, по желанию пользователя, могут быть переданы параметры в соответствии с соглашениями о связях ОС ЕС. Обеспечение интерактивного взаимодействия с программой осуществляется системой PRIMUS с помощью средств закрепления за экраном операций ввода-вывода, использующих стандартные методы доступа BSAM и QSAM. Предварительно пользователь должен определить DD-предложения, предназначенные для описания ввода-вывода, с помощью команд DD и DLST.

Прикладной интерфейс PRIMUS является достаточно развитым. Это объясняется и тем, что на прикладной уровень вынесены многие базовые средства системы такие, как: манипулирование входной и выходной информацией; управление наборами данных, программами и заданиями пользователей. Эти и другие средства выполняются с помощью прикладных программ, составляющих стандартный набор функциональных возможностей системы. Макросредства, предназначенные для взаимодействия с ресурсами системы PRIMUS (терминалами, печатающими устройствами, рабочими и личными наборами данных) довольно разнообразны. Весь набор предлагаемых макрокоманд разбивается на следующие группы средств:

- стандартного оформления программ (PMBEGIN, PMRET, PMEND);
- управления обменом информацией с терминалом (PMPUT, PMGET, PMOUT, PMIN, PMSCINIT, PMSCFORM, PMSCSTR, PMSCALL, PMSCFREE);
- работы с личными наборами данных (PMALLC, PMFREE);
- работа с рабочими наборами данных (OPENWDS, PUT, GET, CLOSEWDS);
- работы с печатающими устройствами (OPENPRNT, PUTPRNT, CLSEPRNT, TESTCLSS).

В системе PRIMUS имеется возможность организации обмена информацией с пользователем терминала с помощью макрокоманды CALL, в которой указываются адреса различных модулей системы. Для программ, написанных на ПЛ/1, средства доступа к ресурсам PRIMUS основаны на операторе CALL.

ГЛАВА 5

ВОЗМОЖНОСТИ СИСТЕМЫ ОБЬ

Система ОБЬ является универсальной программной системой, предназначенной для решения широкого класса задач в мультитерминальном интерактивном режиме с распределенной средой

хранения и обработки информации. На ее базе могут быть реализованы, например, информационно-справочные и информационно-поисковые системы, автоматизированные рабочие места, подсистемы сбора данных.

Система ОБЬ принадлежит классу управляющих программ. К ее объектам управления относятся: прикладные программы, терминалы, устройства документирования информации, аппаратура сопряжения ЭВМ. Кроме того, имеются средства управления системой во время работы, которые позволяют изменять конфигурацию обслуживаемых периферийных устройств и состав прикладных программ без ее перезапуска. Объектом управления системы ОБЬ является также вспомогательный набор данных на томе прямого доступа, так называемый файл работ.

Прикладные программы разрабатываются пользователями (абонентами) системы, и их назначение может быть произвольным. Система ОБЬ обеспечивает:

- разнообразные вызовы программ с терминалов;
- мультитерминальный доступ к прикладным программам;
- передачу программам входной информации, которая может быть подготовлена с помощью средств управления данными системы и храниться в файле работ в виде текстов-запросов пользователей;
- прием выходных данных от прикладных программ и их запись в файл работ в виде текстов-ответов абонентам терминалов;
- различные способы чтения и записи данных с терминалов;
- непосредственное управление терминальными устройствами со стороны программ.

Система ОБЬ позволяет использовать в прикладных программах все средства ОС ЕС. Они могут быть написаны на любом языке программирования. Взаимодействие между системой и прикладными программами реализуется с помощью операторов CALL, операторов ввода-вывода языков программирования (GET, PUT, READ, WRITE), причем нет ограничений на использование в них спецификаторов форматов, а также по стандартным соглашениям передачи параметров через поле PARM оператора EXEC языка управления заданиями ОС ЕС.

Прикладные программы могут работать в зоне оперативной памяти системы ОБЬ и выполняться как самостоятельные задания ОС ЕС. В течение одного сеанса работы системы программа может запускаться в зоне системы и вне нее. Режимы выполнения программ управляет оператор. Для пользователя обращение к программе, работающей в зоне памяти системы ОБЬ, не отличается от вызова этой программы, выполняющейся как задание ОС ЕС.

Диалог, который абонент терминала ведет с системой, позволяет ему:

- формировать и корректировать данные, которые могут содержать произвольную символьную информацию и быть оформлены в виде текстов. Они хранятся в файле работ и используются для передачи прикладным программам, абонентам терминалов или распечатываются на специальных устройствах;
- управлять вызовом прикладных программ и документированием информации;
- получать информацию о доступных ресурсах;
- изменять режимы работы и форматы отображения данных на терминал*.

Устройства документирования информации, обслуживаемые системой ОБЬ, предназначены для распечатки произвольных символьных данных, представленных в виде текстов. Процедуру документирования инициирует пользователь терминала или прикладная программа. Обеспечение распределенной среды обработки данных в системе ОБЬ осуществляется с помощью устройств сопряжения ЭВМ. По каналам связи между ЭВМ под управлением системы проходят следующие потоки данных: тексты-запросы к прикладным программам, работающим на других ЭВМ; тексты-ответы адресатам от программ; письма пользователям терминалов; циркулярные сообщения; служебные таблицы ресурсов, с помощью которых абоненты терминалов могут адресоваться к объектам других ЭВМ.

Файл работ системы ОБЬ предназначен для хранения текстов и служебной информации.

Система ОБЬ может функционировать на любой модели ЕС ЭВМ и требует хотя бы один накопитель на магнитных дисках для размещения библиотеки загрузочных модулей системы и файла работ.

В комплекс технических средств, обслуживаемых системой, входят: дисплейные комплексы

ЕС-7920 (локальные и удаленные варианты). ЕС-7910 и его аналог ЕС-8566; локальная дисплейная станция ЕС-7906; абонентские пункты ЕС-8501 и ЕС-8501.М (с любыми допустимыми способами подключения), ЕС-8504, ЕС-8534 (с коммутируемым и некоммутируемым одноточечным соединением), ЕС-8562 (с дисплеями VTS-56100 и ИЗОТ-7925), ЕС-8564, ЕС-8575.М (в режиме соперничества с автоконтролем); дисплеи СМ-1604, СМ-1611, СМ-7209, VDT-52100.С; телеграфный аппарат Т-63; минитерминал СМ-1605.М1; любые устройства печати, обслуживаемые базисным последовательным методом доступа (BSAM) и последовательным методом доступа с очередями (QSAM); устройства сопряжения ЕС ЭВМ, функционирующие по синхронному байт-ориентированному способу передачи информации по выделенным одноточечным каналам связи (CHX1); устройства сопряжения ЕС ЭВМ и шестнадцатиразрядных ПЭВМ типа IBM PC, работающие по стандарту RS-232-C в асинхронном режиме.

Прикладные программы должны находиться в библиотеках загрузочных модулей, доступных системе в процессе работы. Это могут быть любые библиотеки, которые пользователи подключают к системе во время работы, сцепленные библиотеки задания или шага задания, библиотека SYS1.LINKLIB или любая из библиотек, перечисленных в разделе LNKLST00 библиотеки SYS1.PARMLIB.

Системе ОБЬ через поле PARM оператора EXEC или на этапе инициализации указывается идентификатор списка прикладных программ. В этом списке, подготавливаемом службой сопровождения системы, находятся имена программ и их характеристики (пакетная, терминальная, доступная для абонентов других ЭВМ программа и др.). На основании информации из этого списка система обеспечивает пользователей терминалов соответствующим доступом к прикладным программам. Оператор системы имеет возможность изменять состав и характеристики программы во время работы системы.

Состав прикладных программ, доступных пользователю терминала на данный момент времени, отображается в виде таблиц ресурсов (локальных и удаленных). Их содержимое выводится абоненту в ответ на команды получения информации о состоянии ресурсов. Система обеспечивает достоверность этих таблиц на любой момент времени. Наряду с прикладными программами в них находятся идентификаторы терминалов, устройств печати, ЭВМ, операторов.

Пользователи терминалов взаимодействуют с системой по определенному алгоритму. Он включает в себя такие понятия, как листинг, фрейм, программа.

Листинг — это режим накопления информации о диалоге с абонентом за некоторый промежуток времени. Для дисплеев он реализуется выводом сообщений в определенную область экрана. В нее записываются принятые команды, ответы, сообщения от прикладных программ, пользователей, операторов и других компонентов системы. Размер области зависит от типа дисплея. Для печатающих терминалов область листинга не поддерживается, так как вся предыстория диалога сохраняется на рулоне бумаги пишущей машинки.

Фрейм — это режим работы с данными, к которым относятся:

- тексты запросов, ответов, писем во время их создания, просмотра или корректировки;
- справочная информация о состоянии текстов;
- таблицы локальных, удаленных ресурсов и задач пользователя.

Фрейм включает в себя буфер оперативной памяти, который выделяется в момент перехода в этот режим и освобождается во время выхода из него. На терминале выделяется область, через которую происходит обмен данными с абонентом. Управление фреймом, т. е. изменение его содержимого, выполняется с помощью подкоманд команды, которая перевела терминал в режим фрейма.

В режиме программы абонент работает по алгоритму терминальной прикладной программы. После завершения диалога происходит возврат терминала системе и перевод его в режим листинга. Пользователь имеет возможность перевести устройство в режим программы без ее вызова. В этом случае терминал отключается от системы и становится доступным для распределения другим программам, работающим параллельно с системой ОБЬ в ОС ЕС. Абонент может запросить возврат терминала системе через заданное время.

В режимах фрейма и программы система запоминает в файле работ все сообщения,

адресованные пользователю. Они выводятся на терминал после перехода устройства в режим листинга. В режимах листинга и фрейма команды и подкоманды вводятся в соответствующее поле после появления разрешающих сообщений. Начало поля отмечается курсором для дисплеев или положением каретки для печатающих терминалов.

Система ОБЬ является самонастраиваемой программой. Персоналу, ответственному за ее эксплуатацию, необходимо лишь:

- отвести память для файла работ;
- подготовить списки прикладных программ, процессов на многоточечных каналах связи, имен операторов DD для методов доступов системы ОБЬ;
- составить задание или процедуру на запуск системы.

На основании операторов DD, указанных в задании, и определенных списков система автоматически построит необходимое управление данной конфигурацией ресурсов.

Память для файла работ отводится стандартными способами, принятыми в ОС ЕС для выделения памяти под последовательные наборы данных на накопителях магнитных дисков. Параметры DCB в метке, описывающей этот набор данных, должны быть следующие: DSORG = PS, RECFM = F, BLKSIZE = X, где X может принимать любые значения от 86 байтов до размеров дорожки используемого накопителя.

Список прикладных программ записывается в библиотеку загрузочных модулей системы ОБЬ в результате ассемблирования и редактирования связей ОС ЕС. Он имеет линейную структуру. Каждый элемент списка описывает прикладную программу. Длина элемента составляет 18 байтов и содержит имя программы (8 байтов); имя оператора DD, определяющего библиотеку, в которой находится программа (8 байтов); тип программы (1 байт); максимальное число одновременного доступа (1 байт).

Список процессов на многоточечных каналах связи (МКС) требуется, если система обслуживает периферийное оборудование, подключаемое к ЭВМ посредством многоточечных каналов связи (например, удаленный комплекс ЕС-7920). Необходимость списка обусловлена тем, что язык управления заданиями ОС ЕС не позволяет описывать непосредственное обращение к устройствам, расположенным на МКС, а позволяет описывать в операнде UNIT оператора DD только сам канал связи. Непосредственное обращение к устройствам, расположенным на МКС, осуществляется в самой программе с помощью списков опроса (для чтения) и выборки (для записи), где указываются адреса этих устройств на канале связи.

Для обеспечения асинхронной работы устройств, подключенных к таким каналам, в системе ОБЬ применяются списки процессов на МКС. Список предназначен для указания системе конфигурации устройств, которые необходимо обслуживать на МКС, описанном оператором DD.

Список имен операторов DD, зарезервированных за методами доступов системы ОБЬ (последовательным и терминальным), применяется для более гибкого управления распределением ресурсов прикладным программам, работающих в зоне оперативной памяти системы и использующих ее методы доступа через стандартные операторы ввода-вывода языков программирования (GET, PUT, READ, WRITE). Он имеет линейную структуру, где каждый элемент содержит имя оператора DD. На основании информации из списка система на этапе инициализации распределяет его элементы методам доступа. Во время работы оператор имеет возможность перераспределять операторы DD с помощью команд динамической реконфигурации. Они могут указываться и непосредственно в задании на запуск системы, но применение списка позволяет сократить количество операторов DD в нем.

Процедура или задание на запуск системы ОБЬ состоит из одного шага — одного оператора EXEC. В поле PARM этого оператора могут быть указаны идентификаторы списков прикладных программ, процессов на МКС и имен операторов DD методов доступов.

Требуется один оператор DD для описания файла работ.

Каждое периферийное устройство, которое предполагается использовать в системе, и которое не подключено к МКС, описывается одним оператором DD. Через его имя системе передается тип ресурса (терминал, устройство печати, канал связи с ЭВМ) и код устройства, например, локальный дисплей ЕС-7920. Через поле операндов оператора DD ей сообщается адрес устройства (операнд UNIT); запрос на отложенное обслуживание (операнд DUMMY);

интервал времени периодического опроса состояния готовности оборудования (операнд LABEL); пакет автоматически выполняемых команд после включения терминала (операнд DSNAME).

Необходим один оператор DD для описания МКС. Его имя должно совпадать с соответствующим значением в списке процессов на МКС, идентифицированном при запуске системы. Через поле операндов системе передается запрос на обслуживание всей конфигурацией устройств, описанных в списке МКС под данным именем оператора DD (UNIT); указание на отложенное обслуживание (DUMMY); требование работать со всеми устройствами конфигурации по состоянию готовности (LABEL) или необходим индивидуальный подход к каждому устройству (специальные указания в МКС); пакет команд всем терминалам (DSNAME) или определенным устройствам (МКС).

ГЛАВА 6

ПРИНЦИПЫ ПОСТРОЕНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ СИСТЕМЫ ОБЪ

Основная задача, которая ставилась перед разработкой системы ОБЪ, аналогична задаче, решаемой другими известными системами телеобработки такими, как КАМА, PRIMUS, ОТМД и др. Кратко ее можно сформулировать так: построить управляющую систему, способную обслуживать одновременно большое количество разнообразных терминалов, с помощью которых можно было бы взаимодействовать с прикладными программами самого широкого профиля. Причем это взаимодействие должно распространяться на всю распределенную систему обработки информации, т. е. прикладная программа может работать на одной ЭВМ, а обращение к ней может происходить со всех связанных с системой ЭВМ.

Вместе с основной задачей выдвигались дополнительные требования:

- независимость прикладных программ от телемонитора;
- простота включения новых программ в систему;
- легкость подготовки к работе на определенную конфигурацию ресурсов;
- возможность расширения системы, как путем включения новых типов периферийных устройств, так и путем наращивания ее функциональных возможностей;
- сохранение работоспособности при выходе из строя каких-либо компонентов (например, отказ терминала) с возможностью повторного запуска;
- малое время реакции на команды абонентов;
- компактность управляющей системы.

В решении данной задачи использовался принцип адаптивности программного обеспечения (ПО) мультитерминальных систем распределенной обработки информации. Под адаптивностью ПО понимается его способность самонастраиваться на конфигурацию конкретного вычислительного комплекса. При этом из множества возможностей мультитерминальной системы выбирается некоторое подмножество. Программные компоненты, реализующие эти возможности, должны обладать относительной независимостью, допускающей расширение возможностей системы, а также модификацию программных блоков без изменений смежных модулей. Модифицируемость отдельных компонентов без последующей переделки системы в целом — на всех этапах разработки и после ее завершения — должна рассматриваться как одно из основных требований к ПО мультитерминальных систем.

Основу алгоритмического построения ПО системы ОБЪ составляют положения об управлении процессами и ресурсами.

Система ОБЪ реализована в виде совокупности асинхронных процессов связи с оператором системы, обслуживания терминалов, управления прикладными программами, печати, передачи данных по каналам связи между ЭВМ, сбора статистических данных, файла работ.

Процесс — логическое понятие, близкое к понятию «задача» (в терминологии ОС ЕС), но не тождественное ему. Он создается для выполнения некоторой работы на этапе инициализации системы и логически независим от других процессов. Любой процесс, за исключением процессов обслуживания оператора и файла работ, может иметь одно из трех состояний: активное, пассивное и резервное. Он может переходить из одного состояния в другое во время

работы системы.

Состояние процесса определяет его отношение к объекту управления. Изменение его состояния выполняется по командам оператора или автоматически при получении определенных условий, например, готовность устройства. Для терминальных процессов дополнительно можно указать последовательность (пакет) любых команд пользователя. Она будет автоматически выполняться процессом после его перевода в активное состояние. Тем самым, можно задать работу процесса по определенному направлению. Например, так спланировать обслуживание терминала, что его абонент будет иметь возможность работать только с конкретной прикладной программой.

В активном состоянии процесс выполняет работу, для которой он создавался. Если это терминальный процесс, то он ведет диалог с пользователем. Если это процесс управления прикладной программой, то он принимает запросы со стороны других процессов и, если они корректны, запускает их выполнение; определяет характер завершения работы по данному запросу и возвращает результаты обработки процессу, от которого пришел вызов. Все активные процессы имеют задачи или иерархии задач и конкретные объекты управления: терминал, программа, принтер и др.

В пассивном состоянии процесс не выполняет «полезную» работу, но конкретный объект управления ему распределен. Если это процесс, который обслуживает периферийное устройство, то оно распределено системе и поэтому не может быть выделено другим заданиям, выполняющимся параллельно с системой ОБЬ в ОС ЕС. О существовании пассивных процессов знает только оператор системы. Для пользователей терминалов они недоступны — отсутствуют в таблицах ресурсов.

В любой момент времени оператор или сама система автоматически может запустить — перевести в активное состояние — пассивный процесс. Тогда он станет известен — появится в таблицах ресурсов — всем абонентам терминалов распределенной среды. Исключение составляют прикладные программы, доступные только для местного использования. О их наличии оповещаются только пользователи терминалов ЭВМ, где она выполняется.

В резервном состоянии процесс не выполняет работу и конкретного объекта управления ему не выделено; о процессе известны только его тип (терминальный, программный и т. д.) и какой объект он обслуживал последний раз (например, дисплей локального комплекса ЕС-7920).

Оператор имеет возможность перевести процесс из резервного состояния в пассивное, указав конкретный объект управления. Если этот процесс, предназначенный для обслуживания периферийного устройства, то оператор указывает его физический адрес, который «захватывается» системой. Если это программный процесс, то он может задать имя, место хранения и характеристики программы.

Классификация процессов системы ОБЬ приведена в табл. 1.

Таблица 1

Состояние процесса	Идентификатор процесса	Объект управления	Способ управления
Активный	Адрес устройства [.суффикс из списка МКС] или имя прикладной программы	Устройство или прикладная программа	Задача или иерархия задач
Пассивный	Адрес устройства [.суффикс из списка МКС] или имя прикладной программы	Устройство или прикладная программа	Нет или задача проверки устройства
Резервный	Последние 5 байтов имени оператора DD устройства [.суффикс из списка МКС] или имя прикладной программы	Тип ресурса	Нет

Диаграмма изменения их состояний изображена на рис 2. Здесь

(1) —,[ИДПН], [DDNAME], [PARM], [EP], [T] [.USER = KHC]
[,COM = ГКП]

(2) — [,USER = КНС] [,COM = ГКП]	
A, M, S, U, P, R	— команды оператора;
ИДП	— идентификатор процесса;
ИДПН	— новый идентификатор процесса;
DDNAME	— имя оператора DD, описывающего периферийное устройство или библиотеку, в которой находится прикладная программа;
PARM	— параметры процесса (списки опроса и выборки, тип и счетчик одновременного использования прикладной программы и др.);
EP	— имя модуля, выполняющего ввод-вывод (только для процессов, обслуживающих периферийные устройства);
T	— интервал времени периодического опроса устройства для автоматического запуска процесса по состоянию готовности оборудования;
КНС	— команда начала сеанса абонента;
ГКП	— группа команд пользователя терминала.

Сформулируем *положения об управлении процессами*:

— процессы создаются в соответствии со специфицированной средой — операторы DD в задании на запуск системы, списки прикладных программ и процессов на многоточечных каналах связи (МКС);

— активные процессы взаимодействуют между собой с помощью сообщений, пересылаемых процессом обслуживания файла работ;

— синхронизация работы процессов достигается с помощью средств ОС ЕС — WAIT/POST, ENQ/DEQ — и системных управляющих таблиц.



Рис. 2. Диаграмма изменения состояния процессов системы ОБЪ

Система управления процессами включает в себя аппарат управления (средства синхронизации и управляющие таблицы) и набор типов и форматов сообщений, которыми процессы обмениваются между собой. За счет разделения целей (сообщения межпроцессного взаимодействия) и средств управления их можно модифицировать независимо друг от друга.

Это означает, что можно усложнять форматы сообщений для получения новых функциональных возможностей системы на существующих средствах управления. И, наоборот, можно оптимизировать аппарат доставки сообщений, не меняя типов сообщений. Это позволяет получить удовлетворительные показатели по гибкости, надежности и оперативности работы всей системы в целом.

Любой активный процесс, разрушенный в результате сбоя оборудования или других причин, может быть восстановлен по командам оператора или автоматически системой. Таким образом, надежность системы управления процессами определяется надежностью процесса оператора — старшего в системе. Он всегда находится в активном состоянии, его задача является главной задачей шага задания, и он спроектирован с повышенной степенью надежности — все это возможно при использовании средств восстановления STAE ОС ЕС.

Среднее время пересылки сообщения между процессами зависит от функционирования процесса обслуживания файла работ. В его реализации большое внимание было уделено сокращению времени прохождения сообщений через файл работ.

Положения об управлении ресурсами системы ОБЪ формулируются следующим образом:

— все ресурсы системы разбиваются на типы, и каждому типу назначается в качестве управления свой функциональный блок;

— каждая единица ресурса определенного типа имеет автономное обслуживание в виде

процесса;

—совокупность признаков, характерных для ресурсов одного типа, составляют логическую часть, управление которой выполняется программой логической единицы;

—признаки, не вошедшие в общую (логическую) часть, определяют физические характеристики объектов данного типа ресурсов. Им назначается управление в виде программ физических единиц.

Ресурсы системы ОБЬ по типам делятся на прикладные программы, терминалы, принтеры, каналы связи между ЭВМ, устройства накопления статистической информации, пульт оператора, файл работ. Их детализация до логического и физического уровней, выполненная в системе, существенно отличается для разных типов ресурсов.

Такой подход к расслоению программного обеспечения позволяет получить удовлетворительные характеристики системы по адаптивности к расширению перечня ее возможностей; независимости ПО от количественного состава обслуживаемых ресурсов; компактности ПО.

Архитектура системы ОБЬ состоит из управляющих и функциональных блоков, соответствующих типам обслуживаемых объектов. Каждому объекту назначается процесс. Процессы подразделяются на управляющие и функциональные. Управляющих процессов — два: верхнего уровня — оператор системы — и нижнего уровня — файл работ. Функциональные процессы группируются по типам: терминальные, программные, транспортные, принтерные и статистические. Структура множества процессов системы ОБЬ изображена на рис. 3. Как видно, в ней отсутствуют прямые связи между функциональными процессами, которые определяют основное назначение системы. Взаимодействие между ними происходит через служебные управляющие процессы оператора и файла работ. Тем самым, значительно повышается автономность каждого блока, что позволяет, в дальнейшем, достаточно легко наращивать возможности системы как в плане модификаций внутри блока, так и в плане добавления новых блоков, решающих другие задачи.

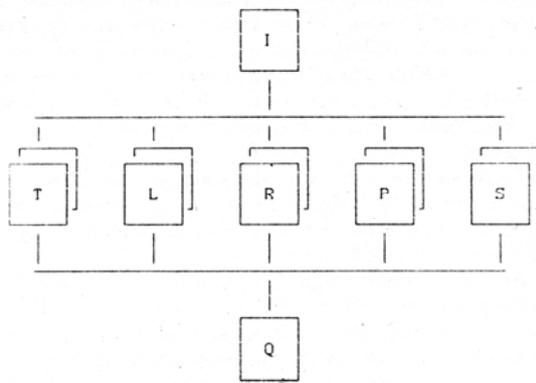


Рис. 3. Структура множества процессов системы ОБЬ:

I — управляющий процесс верхнего уровня; T - процессы обслуживания терминалов; L — процессы управления прикладными программами; R - процессы управления связью с ЭВМ; P — процесс документирования данных; S — процесс сбора статистической информации; Q - управляющий процесс нижнего уровня

Управляющие процессы оператора и файла работ присутствуют в системе при любом режиме эксплуатации. Необходимость появления других процессов определяется на этапе инициализации системы в соответствии с указанной средой.

Назначение процесса оператора состоит в построении среды функционирования системы в соответствии с заданной конфигурацией ресурсов и осуществлении ее управления во время работы. Процесс оператора — старший в системе. Его задача является первой задачей шага задания. Он всегда находится в активном состоянии и выполняется программой управления средой — логическая единица. Модуль физической единицы реализуется макрокомандами связи с консолью ОС ЕС.

Необходимая конфигурация ресурсов определяется операторами DD в задании на запуск системы и тремя списками: прикладных программ, процессов на МКС и имен операторов DD для программ, работающих в зоне памяти системы и использующих методы доступа ОС ЕС

BSAM и QSAM для обращения к терминалам и файлу работ. Идентификаторы этих списков передаются через поле PARM оператора EXEC или задаются оператором на этапе инициализации системы.

На основании этой информации определяется необходимое число функциональных процессов и строятся управляющие таблицы. Затем запускается управляющий процесс обслуживания файла работ с инициализацией или восстановлением файла — в зависимости от указания в поле PARM или оператора. После этого все остальные процессы переводятся в одно из состояний: активное состояние, т. е. запускаются соответствующие задачи, если явно не задан объект управления; пассивное состояние, если указан режим автоопроса, а соответствующее оборудование не включено, или для прикладной программы установлен признак, что она будет выполняться как отдельное задание ОС ЕС; резервное состояние, если объект управления явно не задан, т. е. для периферийных устройств оператор DD определен как DUMMY, а для программ установлен соответствующий признак в списке прикладных программ.

Во время работы системы процесс оператора выполняет команды: управления состояниями процессов, отображения состояний процессов, пересылки циркулярных сообщений, управления распределением операторов DD для прикладных программ, отображения состояния операторов DD, завершения работы системы. Кроме того, процесс оповещает оператора о разрушении функциональных процессов с указанием причины и рассылает необходимые сообщения межпроцессного взаимодействия.

Терминалы системы ОБЬ обслуживаются терминальными процессами. Они состоят из программы управления логическим терминалом и модулей ввода-вывода и автоопроса. Любой терминальный процесс выполняется программой логического терминала и соответствующим модулем ввода-вывода или автоопроса.

Команды диалога, который ведет с абонентом программа логического терминала, позволяют: формировать, корректировать и документировать тексты; вызывать прикладные программы; посылать сообщения пользователям и операторам; выполнять ряд сервисных функций, касающихся управления собственным терминалом.

Пользователи имеют возможность выводить на терминалы таблицы локальных и удаленных ресурсов. В этих таблицах присутствуют идентификаторы активных процессов. Они используются в командах для адресации к соответствующему объекту. Так, идентификатором процесса, обслуживающего прикладную программу, является имя программы, под которым она записана в библиотеку загрузочных модулей. Идентификатором процесса, работающего с периферийным устройством, является его адрес, дополненный, может быть, суффиксом из списка процессов на МКС. Если процесс терминальный и он ведет диалог с пользователем, то к идентификатору процесса добавляется, в скобках, идентификатор абонента.

Прикладная программа — ресурсы системы ОБЬ, обслуживаемые программными процессами. Каждая программа, указанная в списке прикладных программ, представлена функциональным процессом. Он реализуется универсальной программой связи — логическая единица — и, собственно, прикладной программой — физическая единица. Активные программные процессы — комплексные, т. е. представляют собой иерархию задач. Тип программы (пакетная, терминальная, локальная, доступная для обращения из других ЭВМ, работающая в самостоятельной зоне памяти и др.), а также число ее одновременного использования указаны в списке прикладных программ. Во время вызова программ передается информация по стандартным соглашениям передачи данных через поле PARM оператора EXEC. Также, система предоставляет последовательный метод доступа (ПМД) для чтения и записи текстов, хранящихся в файле работ, и терминальный метод доступа (ТМД) для обеспечения интерактивного взаимодействия пользователей с прикладными программами. Вместе с этим, в программах разрешается применение любых средств ОС ЕС

Связь между ЕС ЭВМ в системе ОБЬ обеспечивается транспортными процессами. Они выполняются программой управления логическим каналом и модулями ввода-вывода, автоопроса выделенного канала связи, работающего по алгоритму СНХ1. Система ОБЬ формирует по одному процессу на каждую ЭВМ, указанную при запуске системы.

Транспортный процесс управляет приемом и передачей текстов из файла работ,

коммутацией циркулярных сообщений и таблиц ресурсов, в которых указаны идентификаторы активных процессов, к которым разрешен доступ из других ЭВМ.

Если во время работы системы происходит изменение таблиц локальных ресурсов, затрагивающее объекты удаленного доступа, например, за каким-нибудь терминалом начался диалог с пользователем, то модифицированная таблица ресурсов немедленно будет передана на все связанные ЭВМ. Таким образом, поддерживается достоверность таблиц ресурсов на всех ЭВМ распределенной среды на любой момент времени. Транспортные процессы будут передавать данные по каналам связи только в том случае, если в соответствующей таблице ресурсов указан правильный адресат.

Процессы документирования информации на печатающих устройствах реализованы в соответствии с вышеизложенными принципами. Они распечатывают данные, оформленные в виде текстов — запросов и ответов — файла работ. Вывод информации на принтеры выполняется по разнообразным форматам. Тексты передаются от функциональных процессов — локальных и удаленных.

Процессы сбора статистической информации о работе системы принимают ее из файла работ и записывают в наборы данных. К статистической информации относятся: сообщения межпроцессного взаимодействия; команды оператора; наиболее важные сообщения, выдаваемые процессом оператора на консоль ОС ЕС.

Вспомогательный процесс файла работ функционирует под управлением задач активных процессов. Программа управления очередями — запросов, ответов, обмена сообщениями, свободного пространства и переполнения — составляет логическую единицу процесса. Эта программа выполняет функции обслуживания очередей файла: поиск, удаление, добавление и др. Модуль ввода-вывода — физическая единица — использует метод доступа EХСР ОС ЕС для перемещения физических блоков файла между оперативной памятью и накопителем на магнитных дисках.

Активные и пассивные с автоопросом устройств процессы непосредственно обращаются к процессу файла работ. Запросы выполняются в порядке поступления.

Если в системе есть активные статистические процессы, то все сообщения межпроцессорного взаимодействия оформляются процессом файла работ в виде статистик и направляются этим процессам.

Работа системы ОБЬ зависит от автоматически создаваемых на этапе инициализации и во время работы управляющих блоков и таблиц. Они описывают среду функционирования и обеспечивают работу в соответствии с заданными требованиями. Совокупность управляющих блоков и таблиц можно разбить на два типа: глобальные блоки и таблицы, блоки процессов.

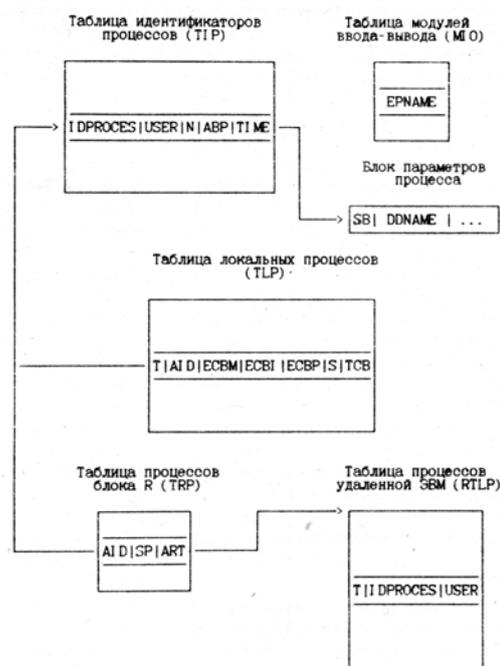


Рис. 4. Взаимосвязь глобальных управляющих таблиц

Глобальные управляющие блоки и таблицы формируются на этапе инициализации и во время работы. Они служат для управления процессами. На рис. 4 изображена их взаимосвязь.

На каждый процесс, который определяется во время запуска системы — кроме процесса управления файлом работ — отводится элемент в таблицах TLP и TTP. Размер таблицы MIO зависит от количества типов периферийных устройств, специфицированных при запуске или во время работы системы, и может меняться в процессе работы. Информация о типах устройств берется из имен операторов DD, списка процессов на МКС и команд оператора. Все транспортные процессы дополнительно собираются в таблицу TRP. В момент приема таблиц ресурсов от удаленных ЭВМ формируются таблицы RTLP, которые «связываются» с соответствующим элементом таблицы TRP.

При переводе процесса в активное или пассивное с автоопросом состояние ему передается блок параметров, доступ к которому выполняется по стандартным соглашениям передачи информации через поле PARM оператора EXEC.

Поля в элементах таблиц означают следующее (в скобках указаны их размеры в байтах):

IDPROCES(8) — идентификатор процесса;

T(1) — тип процесса (0 — прикладной, 1 — терминальный, 2 — принтерный, 3 — операторный, 4 — транспортный, 6 — статистический);

AID(3) — адрес элемента процесса в TTP;

ECBM(4) — блок управления событием ОС ЕС (ECB), в котором отмечается появление сообщений межпроцессного взаимодействия для данного процесса;

ECBI(4) — ECB, куда поступает сигнал от процесса оператора об окончании работы данного процесса;

ECBP(4) — ECB, где ОС ЕС отмечает завершение старшей задачи процесса;

S(1) — состояние процесса:

00..... — активный,

01..... — пассивный,

10..... — пассивный с автоопросом,

11..... — резервный,

..1..... — выполняется команда «R» (см. рис. 2),

...1.... — задана «KHC» (см. рис. 2),

....1... — выполняется команда пользователя «СТОП ВМ»;

TCB(3) — адрес блока управления задачей ОС ЕС (TCB) старшей задачи процесса;

USER(4) — идентификатор пользователя из команды начала сеанса для терминальных процессов или текущее число запросов и адрес списка процессов-инициаторов работ для программных процессов;

N(1) — номер элемента таблицы MIO — начиная с нуля, — где находится имя модуля ввода-вывода. Это поле используется только для процессов, обслуживающих периферийные устройства;

ABP(3) — адрес блока параметров процесса;

TIME(4) — интервал времени — в сотых долях секунды — периодической проверки готовности периферийного оборудования;

EPNAME(8) — имя модуля ввода-вывода;

SP(2) — число элементов в RTLP;

ART(3) — адрес RTLP;

SB (2) — количество байтов данных в блоке параметров процесса без длины поля SB;

DDNAME(8) — имя оператора DD, описывающего периферийное устройство, канал связи или библиотеку, в которой находится прикладная программа;

... — информация, зависящая от типа процесса: для программных процессов — это тип и максимальное число одновременного доступа к программе; для процессов, обслуживающих устройства, подключенные к ЭВМ через многоточечные каналы связи — это списки опроса и выборки и др.

Управляющие блоки процессов создаются в момент перевода процессов в активное или пассивное с автоопросом состояние. Они служат для обеспечения автономной работы

процессов. Каждый управляющий и функциональный блок имеет свой формат управляющего блока процесса. Во время работы процесса системы ОБЬ адрес его управляющего блока находится в десятом регистре. Форматы управляющих блоков процессов хранятся в библиотеке исходных модулей системы ОБЬ в виде фиктивных секций языка ассемблер.

Формат управляющего блока процесса оператора системы имеет следующий вид:

RCSICSA	DSECT		
	USING	*,10	
ICSIDP	DS	CL8	Идентификатор процесса
ICSDWORD	DS	D	Область параметров
ICSPROCA	DS	H	Текущ. число проц-ов с задачами
ICSZTLP	DS	H	Число элементов в TLP
ICSZTIP	EQU	ICSZTLP	Число элементов в TIP
ICSATLP	DS	A	Адрес TLP
ICSATRP	DS	A	Число элементов и адрес TRP
ICSATIP	DS	A	Адрес TIP
1CSAMI0	DS	A	Число элементов и адрес MIO
ICSAECBL	DS	A	Адрес списка ожидаемых событий
ICSEAQ	DS	A	Адр. входа в проц. файла работ
ICSCOM	DS	CL80	Буфер команд оператора
ICSAEND	EQU	*	

Список ожидаемых событий процесса оператора содержит: адрес ECB для связи с оператором через консоль ОС ЕС — поле ECB для связи с оператором консоли в управляющей части блока CSCB ОС ЕС, если передача команд выполняется с помощью команд главного планировщика ОС ЕС MODIFY, STOP, или поле ECBP элемента процесса оператора в TLP, если передача — через команду REPLY ОС ЕС; адрес ECB для приема сообщений от других процессов — поле ECBM элемента процесса оператора в TLP; адреса ECB, в которых отмечаются события окончаний старших задач функциональных процессов — поля ECBP элементов процессов в TLP — или адреса ECB, в которых появляются сигналы о поступлении сообщений для резервных и пассивных процессов — поля ECBM элементов процессов в TLP.

Процесс оператора системы ОБЬ отслеживает выполнение запросов к прикладным программам с помощью списков, адреса которых он помещает в поля USER соответствующих элементов программных процессов TIP. Новая информация добавляется в список после приема запроса к исполнению и удаляется из него по окончании обработки. Ведение списков позволяет в любой момент времени определять объем работы, выполняемый программным процессом.

Формат поля USER:

COUNT	DS	X	Число запросов
ADDR	DS	AL3	Адрес начала списка

Формат элемента списка:

NEXT	DS	A	Адрес следующего элемента
PROCESS	DS	A	Адрес элемента TLP
*			инициатора работы

Управляющий блок терминального процесса описывается фиктивной секцией языка ассемблер следующего вида:

RCSTCSA	DSECT		
	USING	MO	
TCSIDP	DS	CL8	Идентификатор процесса
TCSIDU	DS	CL4	Идентификатор пользователя
TCSFLAG	DS	X	Байт флажков:

*		.1.....	— «*ТЗ» — в пакете команд
*		..1.....	— нет связи с терминалом
*		...1....	—«КНС» передана оператором
*	1...	— работа с текстом-запросом
*	1..	— не выв. сообщения и ответы
*	1.	— вывод сообщений без ид.
*	1	— контроль доступа
TCSFRMSZ	DS	X	Размер фрейма (80 байт. блоки)
TCSZTLP	DS	H	Число элементов в TLP
TCSATLP	DS	A	Адрес TLP
TCSATRP	DS	A	Число элементов и адрес TRP
TCSATRPA	DS	A	Адрес элемента TRP
TCSATRPO	DS	A	Адрес отображаемой RTLP
TCSAECBL	DS	A	Адрес списка ожид. событий
TCSEAQ	DS	A	Адр. входа в проц. файла работ
TCSEAT	DS	A	Адрес входа в модуль ВВ
TCSACS10	DS	A	Адрес области управления ВВ
TCSPFLAG	DS	X	Флаги режима фрейма:
*	1...	— конец текста
*	1..	— передача сообщения
*	1.	— формирование нового текста
*	1	— передача фрейма
*	ORG	TCSFFIAG	
TCSAFBUF	DS	A	Адрес буфера фрейма
TCSAWOR	DS	A	Адрес области корр. текстов
TCSDWORD	DS	D	Область параметров
TCSRNAME	DS	CL8	Идентификатор процесса RTLP
TCSMSS	DS	CL80	Буфер команд и сообщений
TCSENQL	ENQ	(,TCSRNAME),MF = L	
TCSATMSS	DS	A	Адрес модуля терм. сообщений
TCST1DIO	DS	F	Спецпризнаки модуля ВВ
TCSAEND	EQU	*	

Список ожидаемых событий терминального процесса содержит три адреса: поле ECBI элемента процесса в TLP, в котором отмечается указание процесса оператора о завершении работы — код 0 — или переводе из режима автоопроса в активное состояние — код 4; поле ECB окончания операций ввода-вывода на терминале; поле ECBM элемента процесса в TLP, где отмечается появление сообщений от других процессов.

На каждую программу, заданную в списке прикладных программ, формируется программный процесс, который в активном состоянии имеет управляющий блок. Он состоит из префикса и основной части:

RCSLCSAP	DSECT		Формат префикса
LCSEAJBI	DS	A	Адрес входа в инициатор
LCSEAJBD	DS	A	Адрес входа в терминатор
LCSEAFUL	DS	A	Адрес ограничителя работ
LCSATIOT	DS	A	Адрес TIOT ОС ЕС
TCSADDUC	DS	A	Адрес таблицы DD ПМД и ТМД
LCSATCBP	DS	A	Адрес TCB защиты процесса
LCSIDRCS	DS	CL8	Ид. процесса оператора
LCSAPEND	EQU	*	
RCSLCSA	DSECT		Формат основной части
	USING	*,10	
LCSIDP	DS	CL8	Идентификатор процесса
LCSEAX	DS	A	Адрес завершителя задач ОС

LCSEAG	DS	A	Адрес входа в модуль-GET ПМД
LCSEAP	DS	A	Адрес входа в модуль-PUT ПМД
LCSEAM	DS	A	Адрес формирователя ответов
LCSEAQ	DS	A	Адр. входа в проц. файла работ
LCSEAW	DS	A	Адрес входа в модуль-WTO
LCSEATO	DS	A	Адрес входа в модуль-OPEN ТМД
LCSEATG	DS	A	Адрес входа в модуль-GET ТМД
LCSEATP	DS	A	Адрес входа в модуль-PUT ТМД
LCSEATC	DS	A	Адрес входа в модуль-CLOSE ТМД
LCSIAMIO	DS	A	Адрес поля ICSAMIO в RCSICSA
LCSATLP	DS	A	Адрес TLP
LCSAECBL	DS	A	Адрес списка ожидаемых событий
LCSECBPE	DS	F	ECU завершения запросов
LCSAJOBS	DS	A	Число запр. и адр. очереди работ
LCSMJOBS	DS	X	Максимальное число запросов
LCSFLAG	DS	X	Характеристики прикладной прогр.
LCSZTLP	DS	H	Число элементов TLP
LCSADCBL	DS	A	Адрес DCB библиотеки программы
LCSAEND	EQU	*	

Модули инициатора, терминатора, ограничителя работ, завершителя задач аналогичны планировщику ОС ЕС в том смысле, что решают ту же задачу, но по отношению к выполнению запроса на прикладную программу в среде системы ОБЬ.

Модули GET, PUT, WTO, OPEN, CLOSE составляют часть последовательного и терминального методов доступов системы, которые описываются в гл. 10, касающейся прикладного программирования в системе ОБЬ.

Список ожидаемых событий программного процесса содержит: адрес ECB, в котором отмечается указание процесса оператора о переводе в пассивное состояние — поле ECBI элемента процесса в TUP; адрес ECB, где указывается окончание работы прикладной программы — поле LCSECBPE; адрес ECB для приема сообщений от других процессов — поле ECBM элемента процесса в TLP.

Байт LCSFLAG содержит информацию о характеристиках программы. LCSMJOBS — максимальное число одновременного использования прикладной программы.

На каждый запрос, который процесс принял к исполнению, формируется блок управления работой. Они ставятся в очередь текущих работ процесса. Очередь организована в виде односвязанного списка. Адрес первого элемента находится в поле LCSAJOBS. Новый блок помещается в конец очереди. Максимальное число элементов в очереди не превышает значение, которое записано в LCSMJOBS. Блок управления работой предназначен для хранения информации, связанной с прохождением запроса. Его формат имеет следующий вид:

RCSLJOB	DSECT		
LJOBNEXT	DS	A	Адрес следующей RCSLJOB
LJOBATCB	DS	A	Номер запроса и адр. 1 -й TCB
LJOBALCS	DS	A	Адрес RCSLCSA
LJOBENDC	DS	F	Признаки сост. заверш. работы
LJOBAQSG	DS	A	Адреса блоков связи с процессом
LJOBAQSP	DS	A	файла работ GET и PUT ПМД.
LJOBEP	DS	CL4	Имя модуля ввода-вывода ТМД
LJOBEBAT	DS	A	Адрес входа в модуль ВВ ТМД
LJOBATCS	DS	A	Адрес области управления ВВ ТМД
LJOBPRM1	DS	A	Адрес LJOBPRMS или LJOBUFPS
LJOBPRM2	DS	A	Адрес LJOBUFPS
LJOBPRMS	DS	H	Количество символов в LJOBPRMB
LJOBPRMB	DS	CL80	Буфер данных PARM

LJOBUFPS	DS	H	Длина блока параметров процесса
LJOBUF	DS	CL80	Буфер обмена с ПМД и ТМД
LJOBCOM	DS	CL44	Команда инициатора работы
LJOBRESV	DS	XL4	Резерв
LJOBEND	EQU	*	

В буфере LJOBPRMB находятся данные, заданные, например, пользователем терминала в операнде PARM команды вызова прикладной программы. Блок параметров процесс-инициатора работы (см. рис. 4) передается в буфер LJOBUF, который, в дальнейшем, используется в последовательном и терминальном методах доступа. LJOBPRM1 содержит адрес LJOBPRMS, если есть поле PARM. В противном случае, адреса в LJOBPRM1 и LJOBPRM2 совпадают.

Запрос на прикладную программу реализуется в системе запуском задачи ОС ЕС. Первая область сохранения регистров задачи — адрес области содержится в поле TCBFSA блока TCB — используется системой ОБЬ для поддержания взаимодействия с программой через ее методы доступа. При передаче управления прикладной программе регистр 13 содержит адрес области сохранения. Формат ее — следующий:

RCSLNKSA	DSECT		
	USING	*,15	
LNKSTIME	POST	LNKTECB, (0)	Программа-выхода таймера RCSSTM
	BR	14	
	DROP	15	
LNKFORM	DS	A	Адрес след. обл. сохранения
	DS	F	Резерв
LNKAECBA	DS	A	Адрес ECB для отметки RCSTAI
LNKAIDP	DS	A	LINK SF = L
LNKATAI	DS	A	Адрес области RCSTAI
LNKAPAR	DS	A	CALL MF = L для ПМД и ТМД
LNKAABUF	DS	A	
LNKALENG	DS	A	
LNKPAR	DS	A	Адрес буфера обмена LJOBUF
LNKABUFS	DS	A	Адрес буфера программы с ТМД
LNKLENG	DS	II	Длина буфера программы с ТМД
LNKLBUF	DS	H	Длина буфера ТМД
LNKABUF	DS	A	Адрес буфера ТМД
LNKLGET	DS	H	Длина прочитанных данных ТМД
LNKLPUT	DS	H	Длина записанных данных ТМД
LNKASAD	DS	A	Адрес области RCSSAD
LNKTECB	DS	A	ECB для отметки по таймеру
LNKBINTV	DS	F	Интервал для таймера

Транспортные процессы управляют приемом-передачей информации между ЭВМ. Формат управляющего блока:

RCSRCSA	DSECT		
	USING	*, 10	
RCSIDP	DS	CL8	Идентификатор процесса
RCSDWORD	DS	D	Рабочая область
RCSZTLP	DS	H	Число элементов в TLP
RCSZRTLTP	DS	H	Число элементов в RTLTP
RCSATLP	DS	A	Адрес TLP.
RCSARTLP	DS	A	Адрес RTLTP
RCSATRPS	DS	A	Адрес элемента TRP процесса

RCSEAQ	DS	A	Адрес входа в проц. файла работ
RCSEAR	DS	A	Адрес входа в модуль ВВ
RCSACS10	DS	A	Адрес области управления ВВ
RCSAECBL	DS	A	Адрес списка ожидаемых событий
RCSFLAG	DS	X	Флаги режимов
RCSTYPE	DS	X	Тип приема-передачи по каналу
RCSZMSS	DS	H	Количество байтов в буфере
RCSMSS	DS	CL80	Буфер данных
RCSWORK	DS	CL160	Сообщение инициатора работы
RCSENQL	ENQ	(,,,8), MF = L	Список для работы с RTLP
RCSENQLE	EQU	*	
RCSGETML	GETMAIN	EC, MF = L	Список для формир. RTLP
RCSAEND	EQU	*	

Список ожидаемых событий аналогичен списку событий терминального процесса, только вместо адреса ECB окончания операций ввода-вывода на терминале, естественно, используется адрес ECB завершения операций ввода-вывода в канале связи.

Управляющий блок процессов распечатки текстов имеет следующий формат:

RCSPCSA	DSECT		
	USING	*, 10	
PCSIDP	DS	CL8	Идентификатор процесса
PCSATLP	DS	A	Адрес TLP
PCSAECBL	DS	A	Адрес списка ожидаемых событий
PCSEAQ	DS	A	Адрес входа в проц. файла работ
PCSEAP	DS	A	Адрес входа в модуль ВВ
PCSACSIO	DS	A	Адрес области управления ВВ
PCSZTLP	DS	H	Число элементов в TLP
PCSFLAG	DS	X	Флаги печати
PCSDWORD	DS	D	Рабочая область
PCSEPLDV	DS	CL8	Имя модуля-разделителя текстов
PCSMSS	DS	CL80	Буфер данных
PCSCOM	DS	CL80	Команда инициатора печати
PCSPARM	DS	CL80	Параметры распечатки
PCSAEND	EQU	*	

Список ожидаемых событий процесса содержит два адреса ECB, в которых отмечаются сигналы процесса оператора о завершении работы, переводе в активное состояние и сигналы о поступлении сообщений от других процессов.

Процессы сбора статистических данных о работе системы имеют управляющий блок следующего вида:

RCSSCSA	DSECT		
	USING	*, 10	
SCS1DP	DS	CL8	Идентификатор процесса
SCSATLP	DS	A	Адрес TLP
SCSAECBL	DS	A	Адрес списка ожидаемых событий
SCSEAQ	DS	A	Адрес входа в проц. файла работ
SCSEAS	DS	A	Адрес входа в модуль ВВ
SCSACSIO	DS	A	Адрес области управления ВВ
SCSDWORD	DS	D	Рабочая область
SCSZTLP	DS	H	Число элементов в TLP
SCSMSS	DS	CL80	Буфер данных
SCSAEND	EQU	*	

Список ожидаемых событий аналогичен списку ожидания процесса документирования данных (распечатки текстов).

Объектом управления вспомогательного процесса файла работ системы является его набор данных — файл работ. Он состоит из заголовка и пяти очередей: запросов, ответов, обмена сообщениями, свободного пространства и переполнения.

Заголовок файла содержит заголовки очередей, информацию, необходимую для идентификации запросов и ответов, пароль максимального приоритета выполнения запросов.

Очереди представляют собой упорядоченные двухсвязанные списки. Упорядоченность ускоряет поиск, а двухсвязанность повышает надежность хранения данных в очередях. Возникающая в списковых структурах данных проблема «мусора» и «висячих ссылок» решена в плане уменьшения вероятности появления «висячих ссылок».

Физически файл работ состоит из блоков фиксированной длины, составляющих набор данных на томе прямого доступа. Размер блока определяется при создании файла и может варьироваться от 86 байтов до размера дорожки тома прямого доступа.

Минимальной логической единицей в файле является запись. Она имеет фиксированную длину 86 байтов. Первые 80 байтов содержат данные — это заголовки и блоки текстов, а оставшиеся шесть байтов — указатели, с помощью которых запись помещается в различные списки: очереди, тексты.

Указатель в файле работ имеет длину три байта. Он представляет собой порядковый номер записи в наборе данных. Во время работы системы указатель преобразуется в адрес физического блока на томе прямого доступа, содержащего требуемую запись, и смещение в блоке до нее.

Заголовок файла работ находится в первом блоке набора данных. Его структуру можно описать фиктивной секцией языка ассемблера следующего вида:

RCSQDIR	DSECT		
	USING	*, 10	
QCRFREE	DS	XL3	Указатель на свобод. запись
	DS	X	Резерв
QCRFULL	DS	XL3	Адр. 1-й зап. очереди переполн.
	DS	X	Резерв
	DS	XL2	Число записей, взятых из нее
QCREQ	DS	XL3	Указ. на запрос с мин. номером
	DS	X	Флаги очереди запросов
	DS	XL3	Указ. на запрос с макс. номером
	DS	X	Резерв
	DS	XL4	Мин. номер текста в очереди
	DS	XL4	Макс. номер текста в очереди
QCREPLY	DS	XL3	Указ. на ответ с мин. номером
	DS	X	Флаги очереди ответов
	DS	XL3	Указ. на ответ с макс. номером
	DS	X	Резерв
	DS	XL4	Мин. номер текста в очереди
	DS	XL4	Макс номер текста в очереди
QCRMSS	DS	XL3	Адрес сообщ. с макс. приор.
	DS	X	Флаги очереди сообщений
	DS	XL3	Адрес сообщ. с мин. приор.
	DS	X	Резерв
	DS	X	Макс.приоритет в очереди
	DS	X	Мин. приоритет в очереди
PASSWORD	DS	CL8	Пароль
NUMCOUNT	DS	F	Счетчик номеров текстов
QDIREND	EQU	*	

Очередь свободного пространства (QCRFREE) представляет собой связанный стек. Очередь переполнения (QCRFULL) предназначена для поддержания работоспособности системы, когда все записи из очереди свободного пространства окажутся выбранными. Элементы очереди переполнения выделяются только для сообщений межпроцессного взаимодействия. Она не доступна для создания текстов запросов и ответов. По своей структуре очередь переполнения аналогична очереди QCRFREE. Размер очереди вычисляется во время инициализации файла работ, исходя из физического пространства, отведенного под набор данных на томе прямого доступа. После инициализации все записи находятся в очередях свободного пространства и переполнения.

Очереди запросов и ответов представляют собой двухсвязанные списки заголовков текстов, которые упорядочены по возрастанию их номеров.

Заголовок текста занимает одну запись и имеет формат:

RCSQHEAD	DSECT		
IDENTIF	DS	CL4	Идентификатор
NUMBER	DS	F	Номер
FLAG	DS	X	Флаги
BLOKFST	DS	XL3	Указ. на первый блок текста
BLOKLST	DS	XL3	Указ. на послед. блок текста
NUMSHAR	DS	X	Счетчик использования
REMOTIDP	DS	CL8	Идентификатор процесса
REMOTIDR	DS	CL4	Идентификатор и
REMOTNUM	DS	XL4	Номер текста из другой ЭВМ
	DS	XL48	Резерв
HEADFORM	DS	XL3	Указ. на предшествующий текст
HEADBACK	DS	XL3	Указ. на последующий текст

Текст запроса или ответа состоит из записей, которые объединены в двухсвязанный список, что позволяет достаточно просто осуществлять коррекцию текста.

Очередь обмена сообщения (QCRMSS) предназначена для обеспечения межпроцессного взаимодействия и служит также для обмена сообщениями между пользователями терминалов и передачи циркулярных сообщений оператора. Очередь организована в виде двухсвязанного списка, упорядоченного по убыванию приоритетов. Внутри одного приоритета упорядоченность выполняется по принципу «первый пришел — первый ушел». Совокупность форматов сообщений, которыми обмениваются процессы, может быть описана следующим образом:

RCSFMSS	DSECT		
FMSCOM	EQU	RCSFMSS	Команда
FMSIDADR	DS	CL8	Идентификатор адресата
FMSPRTY	DS	X	Приоритет сообщения
FMSFLAG	DS	X	Тип сообщения
FMSIDSEN	DS	CL8	Идентификатор отправителя
FMSCIDRQ	DS	CL4	Идентификатор текста
FMSCNMRQ	DS	XL4	Номер текста
FMSCQUE	DS	C	«3» — QCREQ, «0» — QCREPLY
FMSCRIDP	DS	CL8	Ид. процесса с другой ЭВМ
FMSCSTAT	DS	X	Флаги:
*		1... ..	— изменить список TIPUSER
*		.1.. ..	— команда «СТОП ВМ»
*		..1.	— команда «СТОП»
*		...1	— передать TIP на др. ЭВМ
*	 1...	— снять программу

*		... 1..	— «открыть» устройство
*	1.	— терминал «закрыт»
*	1	— команда «ЗА»
FMSCRWK	DS	XL3	Указатель на доп. текст
FMSCEND	EQU	*	
FMSAGE	EQU	RCSFMSS	Срочное сообщение
	ORG	FMSCIDRQ	
FMSGRIDP	DS	CL8	Ид. процесса другой ЭВМ
FMSGTXT	DS	CL54	Текст сообщения
FMSGEND	EQU	*	
FMSREPLY	EQU	RCSFMSS	Ответ
	ORG	FMSIDSEN	
FMSRTXT	DS	CL70	Текст ответа
FMSREND	EQU	*	
FMSLOG	EQU	RCSFMSS	Статистика
	ORG	FMSIDSEN	
FMSLPTR	DS	XL3	Указ. на текст статистики
FMSLEND	EQU	*	

Все сообщения межпроцессного взаимодействия делятся на четыре типа:
 команда (FMSFLAG = X'00');
 срочное сообщение (FMSFLAG = X'01');
 ответ (FMSFLAG = X'02');
 статистика (FMSFLAG = X'03').

Команды предназначены для запроса у процесса-адресата определенных услуг, например: выполнить прикладную программу; распечатать текст; передать информацию по каналу связи на другую ЭВМ. В поле FMSCRWK может быть указатель на запись дополнительного текста команды. Это — например: группа команд пользователя, передаваемая терминальному процессу на этапе инициализации; параметры запроса к прикладной программе; формат расположения распечатываемого текста. Сообщения типа команда являются основными в системе межпроцессного обмена.

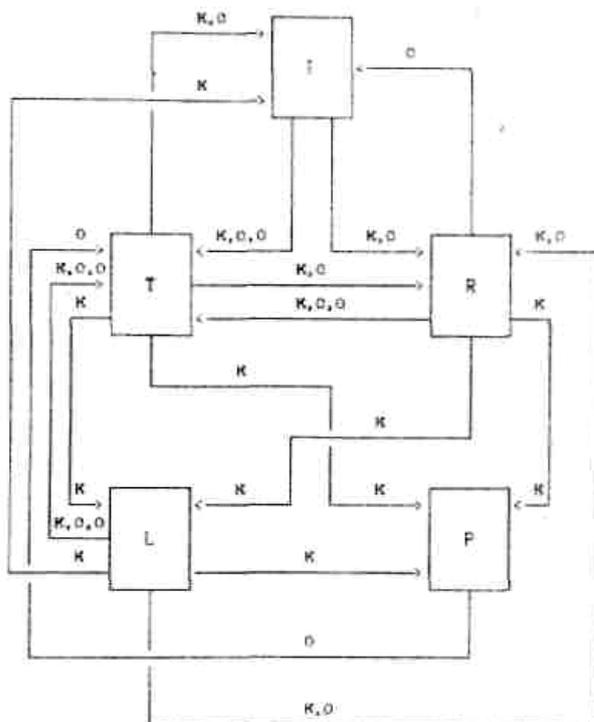


Рис 5 Структура потока сообщений межпроцессного взаимодействия: I - процесс оператора; Т - терминальные процессы; L - программные процессы; R - транспортные процессы; Р - процессы печати; К - команды; С - срочные сообщения; О - ответы

Срочные сообщения используются для обмена информацией между абонентами терминалов и операторов сети с помощью специальных команд.

Ответы служат для передачи процессу-отправителю данных о характере выполнения работы. Они реализуют обратную связь в системе взаимодействия между процессами. Структура потока сообщений в системе ОБЬ представлена на рис. 5. На нем отсутствуют статистические сообщения. Если в системе есть активные процессы сбора статистических данных, то процесс файла работ после записи в очередь обмена сообщениями команд, срочных сообщений, ответов формирует и записывает в эту же очередь статистику, которая состоит из заголовка и текста сообщения межпроцессного взаимодействия — всего две записи. Также процесс файла работ формирует статистику при специальном обращении.

ГЛАВА 7

ТЕХНОЛОГИЯ РАБОТЫ ПОЛЬЗОВАТЕЛЕЙ ЗА ТЕРМИНАЛАМИ СИСТЕМЫ ОБЬ

Абонент терминала системы ОБЬ запрашивает необходимые действия с помощью команд. Набор команд, разрешенных пользователю в течение сеанса, определяется в момент его подключения к системе. Процедура, регламентирующая доступ в систему пользователям, устанавливается персоналом, ответственным за ее эксплуатацию. Санкционированный доступ реализуется с помощью программы-выхода RCSTPASS. Она получает управление при подключении пользователя к системе. На входе ей передается в первом регистре адрес управляющего блока терминального процесса RCSTCSA. Его формат описан в предыдущей главе. В поле TCSMSS записана следующая информация: идентификатор абонента, пароль. Они переносятся в TCSMSS из команды начала сеанса. Программа RCSTPASS по окончании работы обязана передать системе ОБЬ через пятнадцатый регистр код возврата: 0 — пользователю разрешается работать в системе; 4 — недопустимый идентификатор абонента; 8 — неверный пароль. В программе может быть реализована любая процедура контроля, лишь бы при возврате управления пятнадцатый регистр не выходил за диапазон разрешенных значений. Если программа RCSTPASS два раза подряд отвергла команду начала сеанса, то система ОБЬ выводит оператору сообщение:

```
RCS000I XXXXXXXX: ПОПЫТКА НЕСАНКЦИОНИРОВАННОГО  
ДОСТУПА ВРЕМЯ = ЧЧ.ММ.СС
```

XXXXXXX — идентификатор терминального процесса. При наличии в системе активных статистических процессов сообщение записывается в соответствующие наборы данных.

Определенный порядок работы за терминалом может быть задан в операторе DD, описывающем устройство. Он имеет следующий вид:

```
//RCSTYYZZ DD UNIT = CUU/DUMMY[,LABEL = TTT]  
[,DSN = '[КНС]/[ГКП]']
```

Символы YY детализируют тип обслуживаемого устройства. Например, 2L — означает дисплей локального комплекса EC-7920.

Символы ZZ могут принимать произвольные значения и служат для создания уникальных имен операторов DD.

Если в качестве операнда оператора DD указано DUMMY, то это означает, что процесс, обслуживающий данный тип терминала, должен быть после запуска системы в резервном состоянии. Устройство не распределяется телемонитору. Во время работы возможен «захват» любого допустимого устройства и обслуживание его терминальным процессом.

В операнде LABEL специфицируется интервал времени в секундах периодического опроса терминала на предмет автоматического перевода процесса в активное состояние по готовности оборудования.

Через операнд DSN передается пакет команд абонента терминала. Он состоит из двух частей: команды начала сеанса (КНС) и группы команд пользователя (ГКП). Если КНС не задана, то

процесс после активизации перейдет на ожидание подключения абонента. ГКП будет исполняться после ввода команды начала сеанса. Если КНС указана, то процесс после запуска автоматически ее выполнит. В случае успешного завершения операции процесс перейдет на выполнение ГКП. В противном случае ГКП удаляется из системы. Остаток пакета удаляется также при переходе процесса в пассивное состояние. ГКП имеет следующий формат:

[*] КОМ1 [;[*]КОМ2]...

КОМ — любая команда пользователя терминала; «;» — разделитель команд в группе; «*» — признак «зацикливания» команды вызова прикладной программы или команды «ТЗ» (см. ниже) до тех пор, пока ее процесс находится в активном состоянии.

Если терминал подключен к ЭВМ с помощью многоточечных каналов связи (МКС), то аналогичная процедура может быть определена в списке процессов на МКС.

Команды пользователя терминала системы ОБЬ подразделяются на следующие: начала и конца сеанса; управления текстами; обработки запросов; получения информации о состоянии ресурсов; передачи сообщений; спецобработки.

Команда состоит из поля операции и поля операндов. Поля разделяются пробелами. При вызове прикладных программ с помощью команды «В» разрешается пропускать поле операции. В этом случае команда начинается с поля операндов.

Поле операции содержит мнемоническое обозначение команды или подкоманды; идентификатор пользователя и пароль; число.

Поле операндов может быть пустым или содержать позиционные и ключевые операнды. Они задают необходимую информацию, либо запрашивают некоторые действия системы.

Позиционный операнд является простым или составным именем, числом, последовательностью символов.

Ключевой операнд используется в командах обработки запросов. При вызове прикладных программ он может содержать параметры, как в операнде PARM оператора EXEC языка управления заданиями ОС ЕС. При распечатке текстов он может указывать класс выходной очереди ОС или задавать формат размещения данных на бумаге.

Операнды следуют друг за другом в указанном порядке. Сначала позиционные, потом ключевые. Один операнд от другого отделяется запятой. Если позиционный операнд отсутствует и не является последним в поле операндов, то вместо него указывается запятая. Если операнд единственный в поле операндов или опускается группа последних операндов, то завершающие запятые могут не указываться.

При описании формата команд соблюдаются следующие правила. Операнды, указанные в квадратных скобках «[», «]» необязательны и могут быть опущены по усмотрению абонента — в этом случае принимаются умалчиваемые значения операндов. Альтернативные значения операндов разделены наклонной чертой «/».

Команды начала и конца сеанса. Команда начала сеанса идентифицирует пользователя терминала и начинает сеанс.

Формат команды:

ИДПЛ, [ПАРОЛЬ]

ИДПЛ определяет идентификатор абонента. Он содержит от одного до четырех символов, за исключением запятых и пробелов; ПАРОЛЬ — содержит от одного до восьми символов. Требования, предъявляемые к операнду, зависят от процедуры идентификации пользователей в системе.

После приема команды начала сеанса оператору системы выводится сообщение:

RCS000I XXXXXXXX: НАЧАЛ РАБОТУ ПОЛЬЗОВАТЕЛЬ ИДПЛ ВРЕМЯ =
ЧЧ.ММ.СС

XXXXXXX — идентификатор терминального процесса. Команда завершения сеанса имеет формат:

КОНЕЦ

После ввода команды очищается листинг и появляется сообщение:

```
RCS200I ТЕРМИНАЛ СВОБОДЕН
```

В случае коммутируемого подключения терминала происходит разрыв линии связи. Оператору системы передается сообщение:

```
RCS000I XXXXXXXX: ЗАКОНЧИЛ РАБОТУ ПОЛЬЗОВАТЕЛЬ ИДПЛ  
ВРЕМЯ = ЧЧ.ММ.СС
```

Команды управления текстами. Они предназначены для их создания (команда «З»), просмотра (команда «И»), корректировки (команда «К») и удаления (команда «У»).

Текстом называется произвольный набор 80 байтных записей. Они хранятся в файле работ и подразделяются на запросы — находятся в очереди запросов — и ответы — записаны в очередь ответов. Любой текст определяется четырехсимвольным идентификатором и порядковым номером. Идентификатор задается в команде создания текста, а номер присваивается системой. В очередях запросы и ответы упорядочены по номерам, что ускоряет их поиск. Если идентификатор текста в команде не указан, то ему назначается идентификатор пользователя. Тексты могут быть временными и постоянными. Это соответствует аналогичным наборам данных ОС ЕС. Например, создается временный текст, распечатывается и автоматически удаляется системой из файла работ.

Технология работы пользователя по управлению текстами зависит от типа терминала.

Экран дисплея после ввода команды делится на три части: область фрейма; строка, содержащая подсказывающее сообщение; строка для ввода подкоманд. Область фрейма, которая занимает почти весь экран, предназначена для редактирования текстов. В строку сообщений выводится подсказывающая информация. Последняя строка служит для ввода подкоманд, с помощью которых, собственно, осуществляется манипулирование данными. Абоненты дисплеев, имеющих функциональную клавиатуру, большую часть подкоманд могут вводить в сокращенном виде с помощью функциональных клавиш.

Для печатающих терминалов тексты вводятся с клавиатуры пишущих машинок. Их можно создавать, просматривать и удалять. Корректировать — нельзя. Разрешается модифицировать записи текста во время создания, используя особенности конкретного терминала.

Команды обработки запросов. Эти команды используются для вызова прикладных программ и распечатки текстов. В поле операндов этих команд указываются идентификаторы процессов адресатов; атрибуты текста; приоритет обработки; параметры.

Идентификаторы процессов адресатов — это имя прикладной программы или адрес устройства печати, а если обращение выполняется к объекту другой ЭВМ, то и адрес ЭВМ. В последнем случае указывается адрес ЭВМ и в круглых скобках имя программы или устройства печати.

Атрибуты текста необходимы для печати и прикладных программ, которым требуются входные данные. К атрибутам текста относятся: идентификатор, номер, тип (запрос, ответ).

Приоритет обработки задается числом — от нуля до 254- Он определяет место запроса в очереди к указанному объекту. Очередь упорядочена по приоритетам. Наивысший приоритет с номером 254 защищен паролем, который хранится в файле работ и устанавливается обслуживающим персоналом.

Параметры для прикладных программ — это то, к чему программа имеет доступ по соглашениям передачи данных через операнд PARM оператора EXEC языка управления заданиями ОС ЕС. Для печати в поле параметра задается информация о способе размещения данных на бумаге или идентификатор выходной очереди ОС ЕС.

Пользователь терминала имеет возможность объединить процедуры формирования текста, выполнения запроса и просмотра ответа — в случае обращения к программе — с помощью одной команды (команда «ЗА»).

Команды получения информации о состоянии ресурсов. К ним относятся команды: просмотра локальных ресурсов (команда «ТЛ»); отображения доступных объектов другой ЭВМ (команда «ТУ»); вывода состояний текстов пользователей (команда «ПИ»).

К ресурсам абонента терминала относятся объекты, которым он может адресовать команды

обработки запросов. В результате выполнения команд «ТЛ» и «ТУ» на терминал выводятся таблицы, в которых находятся идентификаторы программ, терминалов, устройств печати, связанных ЭВМ. Следует отметить, что на самом деле количество ресурсов, управляемых системой, может быть большим. В таблицах записаны только те объекты, к которым разрешено обращение в данный момент времени. К остальным доступ закрыт. Управление разрешением и запрещением доступа к ресурсам осуществляется оператором системы.

Просмотр состояний текстов выполняется не только для определения их статусов таких, как: формируется, свободен, занят, корректируется, временный и т. д., но и для автоматизации работы с ними. В состав подкоманд этой команды входят команды просмотра и удаления текстов.

Команды передачи сообщений. Они применяются для обмена короткими и длинными сообщениями между абонентами терминалов.

В команде передачи коротких сообщений текст набирается в поле операндов после идентификатора адресата и имеет ограничение по длине — 54 байта.

В команде передачи длинных сообщений задаются атрибуты текста аналогично командам обработки запросов. Размер сообщений может быть произвольным.

Специальные команды. С помощью этих команд выполняются действия: отключение терминала на заданное время (команда «СТОП»); изменение формата вывода сообщений (команда «УФ»); переключение модулей ввода-вывода (команда «МВВ»); установка таблицы задач (команда «ТЗ»).

С помощью команды «СТОП» пользователь управляет отключением и подключением терминала в системе. Отключение терминала означает его доступность для распределения другим заданиям ОС ЕС.

Формат команды:

СТОП [ВРЕМЯ/ВМ]

ВРЕМЯ — интервал времени в минутах, на которое необходимо отключить терминал от системы (терминальный процесс абонента остается в активном состоянии); ВМ — код, означающий запрос на перевод процесса в пассивное состояние. Если система ОБЬ эксплуатируется в среде ОС ЕС на виртуальной машине СВМ ЕС, то выполняется попытка отсоединить диалоговое устройство пользователя от виртуальной машины.

Ввод команды «СТОП» без операндов приводит к переводу терминального процесса абонента в резервное состояние.

Команда *УФ» устанавливает определенные форматы отображения данных на терминале.

Формат команды:

УФ [Н/Ф],[О/З]

Н — требуется выводить сообщения с идентификаторами системы ОБЬ. Они формируются по следующим правилам: трехбуквенный код «RCS»; цифра, характеризующая компонент системы, пославший сообщение; двухзначный номер сообщения; однобайтный признак (I — информационное сообщение; A — сообщение, требующее ответа; W — сообщение, сигнализирующее об изменении состояния процесса);

Ф — требуется выводить сообщение с флагами. Флаг — это однобайтовый символ, который заменяет идентификатор. Информационным сообщениям системы предшествует знак минус («—»). Сообщениям, требующим ответа, — «*». Сообщениям от других пользователей и прикладных программ — «!»;

О — означает, что принимаются все сообщения;

З — означает, что выводятся только сообщения собственного терминального процесса. Сообщения от других компонентов системы не выводятся.

В результате обработки команды начала сеанса автоматически выполняется команда

УФ Н, О

а в конце сеанса —

УФ Н, З

По команде «МВВ» пользователь терминала имеет возможность переключаться с одного модуля ввода-вывода на другой. Эта возможность реально используется с терминалами, работающими по стандартному асинхронному алгоритму обмена. К ним относятся, например, дисплеи СМ-1604, СМ-1611, СМ-7209, VDT-52100С. Служба сопровождения системы на конкретной вычислительной установке может не знать тип терминала, подключенного к каналу связи, работающего в асинхронном режиме. В этом случае достаточно первоначально передать управление универсальному стартстопному модулю ввода-вывода, который есть в системе ОБЬ. Он предоставляет минимальные возможности абоненту, но может работать с любым типом терминала. Затем пользователь сам может переключиться на подходящий специализированный модуль, максимально реализующий возможности оборудования.

Формат команды:

МВВ [А/Т/ИМЯМОД]

А — требуется вывести имя текущего модуля ввода-вывода; Т — необходимо записать на терминал таблицу, в которой есть имена доступных модулей; ИМЯМОД — имя модуля (оно должно быть в таблице имен модулей ввода-вывода).

Таблица имен оформляется в виде программной секции ассемблера ОС ЕС. Она транслируется и редактором связей ОС ЕС записывается в библиотеку загрузочных модулей системы под именем RCSTCMTB. Таблица содержит произвольный текст. Единственное требование — это присутствие в нем имен доступных пользователям модулей ввода-вывода.

С целью повышения уровня автоматизации вызова прикладных программ пользователями терминалов и построения автоматизированных рабочих мест, реализована подсистема табличной организации доступа к прикладным программам. Она базируется на предварительно составленных обслуживающим персоналом таблицах, так называемых таблицах за дач, которые оформляются в виде загрузочных модулей и записываются в библиотеку системы под специальными именами. Абонент терминала помощью команды «ТЗ» вызывает необходимую таблицу задач и работает с ней. Работа с таблицей задач состоит в вызовах компонент (задач), описанных в ней, по их номерам или нажатием функциональных клавиш. По окончании работы с задачей система опять выводит таблицу на терминал. В качестве компонента может выступать другая таблица задач. Тем самым допускается реализация древовидной структуры связанных между собой таблиц задач. Максимальная длина ветви равна девяти. Есть возможность объединения процедуры, регламентирующей доступ в систему абонентам, и таблиц задач с целью организации автоматического и принудительного режима работы пользователя в рамках определенного для него подмножества задач.

Таблица задач строится как программная секция языка ассемблер, состоящая из констант. Сформированная таблица задач транслируется и записывается в библиотеку загрузочных модулей системы под именем RCSZSSSS. SSSS — это произвольный суффикс длиной от одного до четырех символов, который будет использоваться для адресации к таблице.

Таблица состоит из оглавления и текста.

Оглавление формируется в начале программной секции. Оно описывает элементы и содержит указатели на тексты. Оглавление имеет линейную структуру и состоит из строк фиксированной длины. Каждая строка включает пять адресных констант — всего двадцать байтов. В некоторых константах первый байт используется в качестве специального признака. Окончание оглавления распознается системой по нулевой адресной константе в начале строки. Элементом таблицы может быть задача, ссылка на другую таблицу задач или комментарий. По содержанию строки оглавления система может определить тип элемента. Формат строки следующий (в скобках указана длина поля в байтах):

AMSS(4),LID(1),AID(3),FSTART(1),ASTART(3),AEND(4), APARM(4)

AMSS — адрес 80-байтного текста, описывающего элемент таблицы. Если адрес равен нулю, то это означает конец оглавления;

LID, AID — длина и адрес идентификатора задачи или другой таблицы. Идентификатором задачи должен быть восьмибайтный идентификатор процесса прикладной программы системы ОБЬ. В качестве идентификатора таблицы указываются четыре последних байта имени раздела

библиотеки, в котором находится эта таблица. Если строка описывает комментарий, то LID и AID должны быть равны нулю;

FSTART — признак, указывающий на необходимость запроса подтверждения вызова задачи или другой таблицы у пользователя терминала. Если он равен X 'FF', то при каждом обращении система будет выводить переспрашивающее сообщение и ожидать действия абонента. Простое нажатие клавиши «ВВОД» будет восприниматься как положительное подтверждение. Другие действия трактуются системой как ввод новых данных. Если FSTART равен нулю, то это означает вызов без подтверждения;

ASTART — адрес 80-байтной строки, содержащей команду главного планировщика ОС ЕС, с помощью которой процесс прикладной программы данной задачи может быть переведен в активное состояние. Эту информацию система использует в том случае, когда абонент терминала запросил задачу, а ее управляющий процесс — не активен. Тогда система передает указанную команду главному планировщику и ожидает в течение 30 с активизации процесса. После этого выполняется обращение к задаче. Если поле ASTART равно нулю, то вызов задачи осуществляется только при активном состоянии ее процесса;

AEND — адрес 80-байтной строки, где записана команда главного планировщика, с помощью которой, как правило, терминальный процесс данной задачи выводится из активного состояния. Если поле AEND не равно нулю, то система инициирует выполнение адресуемой команды после обработки запроса пользователя и отсутствия других работ у обрабатываемого процесса;

APARM — адрес параметров. Эта информация аналогична данным, которые задаются в операнде PARM команды вызова прикладных программ. Она оформляется в виде записи переменной длины. Сначала идет двухбайтное поле, в котором содержится длина текста параметров в байтах, затем следует поле параметров.

Текстом в таблице задач называется совокупность символьных констант, на которые указывают адресные поля из оглавления. К тексту относятся: 80-байтные символьные константы, описывающие строки таблицы задач на терминале — это единственная часть таблицы, которая отображается на устройстве; 4- и 8-байтные символьные константы, задающие идентификаторы задач и суффиксы подчиненных таблиц задач; 80-байтные строки, содержащие команды главного планировщика ОС ЕС; записи переменной длины, которые передаются в качестве параметров соответствующим задачам в момент их вызова.

При составлении отображаемой части таблицы следует иметь в виду, что она должна полностью помещаться на экране дисплея. Если это не так, то ее необходимо разделить на несколько частей. Каждая строка оглавления в своем первом поле содержит адрес текста, который выводится в одну строку экрана дисплея. Порядок вывода текста соответствует порядку в оглавлении: сверху — вниз. Если строка — комментарий, то ее текст без изменений копируется на терминал. Если строка оглавления описывает задачу, то система выполняет поиск процесса прикладной программы с идентификатором, совпадающим с идентификатором задачи. В случае удачного поиска и возможности активизации процесса — если он не был в активном состоянии — система в первые два байта текущей строки экрана дисплея выводит порядковый номер, в третий байт — пробел, а в оставшуюся часть строки помещает текст. Если процесс в пассивном или резервном состоянии и нет команды его активизации, то в первую позицию строки выводится знак минус, во вторую — пробел, а с третьей позиции записывается текст. В остальных случаях текст копируется в строку экрана дисплея, начиная с первой позиции. И наконец, если строка оглавления описывает другую таблицу задач, то выполняются действия, аналогичные строке-задаче, процесс которой в активном состоянии.

Приведем пример двух таблиц задач:

RCSZST01	CSECT	
	DC	A(T1M01),4A(0)
	DC	A(T1M02),4A(0)
	DC	A(T1M03),X '8' ,AL3(T01103),3A(0)
	DC	A(T1M04),4A(0)
	DC	A(T1M05),4A(0)

	DC	A(T1M06),4A(0)
	DC	A(T1M07),4A(0)U
	DC	A(T1M08),X'8',AL3(T01I08),3A(0)
	DC	A(T1M09),X'8',AL3(T01I09),3A(0)
	DC	A(T1M10),X'8',AL3(T01I10),3A(0)
	DC	A(T1M11),4A(0)
	DC	A(T1M12),X'8',AL3(T01I12),3A(0)
	DC	A(T1M13),X'8',AL3(T01I13),3A(0)
	DC	A(T1M14),4A(0)
	DC	A(T1M15),4A(0)
	DC	A(T1M16),X'4',AL3(T01T16),X'FF',AL3(0),2A(0)
	DC	A(T1M17),X'4',AL3(T01T17),3A(0)
	DC	A(T1M18),X'4',AL3(T01T18),3A(0)
	DC	A(T1M19),4A(0)
	DC	A(T1M20),4A(0)
	DC	A(T1M21),4A(0)
	DC	A(T1M22),4A(0)
	DC	A(0)
T01I03	DC	C'KDO
T01I08	DC	C'TDV
T01I09	DC	C'TTV
T01I10	DC	C'COPY
T01I12	DC	C'JESSY
T01I13	DC	C'ARGUS
T01T16	DC	C'ST04
T01T17	DC	C'ST03
T01T18	DC	C'ST02
T1M01	DC	C' ДЕМОНСТРАЦИОНН
	DC	C' АЯ ТАБЛИЦА ЗАДАЧ
T1M02	DC	C' КОМПОНЕНТЫ, ОБЕСПЕЧИВАЮЩИЕ РЕШ
	DC	C' ЕНИЕ ЗАДАЧИ «АРМ ПРОГРАММИСТА»:
T1M03	DC	C' РЕДАКТИРОВАНИЕ БИБЛИОТЕК ИСХОДНЫХ МОДУЛЕ
	DC	C' Й;
T1M04	DC	C' ПОДГОТОВКА, ЗАПУСК И ПРОСМОТР РЕЗУЛЬТ
	DC	C' АТОВ РАБОТЫ ЗАДАНИЙ ОС ЕС;
T1M05	DC	C' ОБСЛУЖИВАНИЕ ОГЛАВЛЕНИЙ ТОМОВ ПРЯМОГО
	DC	C' ДОСТУПА И БИБЛИОТЕК;
T1M06	DC	C' СОЗДАНИЕ, УДАЛЕНИЕ И ПОДКЛЮЧЕНИЕ ПОСЛ
	DC	C' ЕДОВАТЕЛЬНЫХ И БИБЛИОТЕЧНЫХ Н/Д ОС ЕС;
T1M07	DC	C' ВВОД КОМАНД ОПЕРАТОРА СИСТЕМНОЙ КОНСО
	DC	C' ЛИ И ПРОСМОТР РЕЗУЛЬТАТОВ ИХ ВЫПОЛНЕНИЯ.
T1M08	DC	C' ПРОСМОТР, КОРРЕКТИРОВКА ДИСКОВЫХ ДАННЫХ
	DC	C' И ВОССТАНОВЛЕНИЕ СБОЙНЫХ УЧАСТКОВ
T1M09	DC	C' ПРОСМОТР ДАННЫХ НА МАГНИТНЫХ ЛЕНТАХ И ФО
	DC	C' РМАТИРОВАНИЕ ЛЕНТОЧНЫХ ТОМОВ
T1M10DC	DC	C' ЗАГРУЗКА РАЗДЕЛОВ БИБЛИОТЕК И ПОСЛЕДОВАТ
	DC	C' ЕЛЬНЫХ Н/Д В ФАЙЛ РАБОТ СИСТЕМЫ ОБЪ,
T1M11DC	DC	C' А ТАКЖЕ ОБРАТНАЯ ВЫГРУЗКА ТЕКСТОВ ИЗ
	DC	C' ФАЙЛА РАБОТ В Н/Д ОС ЕС.
T1M12DC	DC	C' РЕДАКТОР ТЕКСТОВ «JESSY»

```

DC      DC      C'
T1M13DC DC      C' СЕРВИС «ARGUS»
DC      DC      C'
T1M14DC DC      C' ВНИМАНИЕ! «JESSY» И «ARGUS» МОГУТ БЫТЬ В
DC      DC      C' БЫЗВАНЫ ТОЛЬКО С ЛОКАЛЬНЫХ ДИСПЛЕЕВ ТИПА
T1M15DC DC      C'      ЕС-7920 И ТРЕБУЮТ ПРЕДВАРИТЕЛЬ
DC      DC      C' НОГО ЗАПУСКА СПЕЦИАЛЬНЫХ ПРОЦЕДУР.
T1M16DC DC      C' ТАБЛИЦА СПЕЦИАЛИЗИРОВАННЫХ ЗАДАЧ
DC      DC      C'
T1M17DC DC      C' СПИСОК ИГРОВЫХ ПРОГРАММ
DC      DC      C'
T1M18DC DC      C' ПРОВЕРКА ПОСЛЕДОВАТЕЛЬНОГО И ТЕРМИНАЛЬНО
DC      DC      C' ГО МЕТОДОВ ДОСТУПА СИСТЕМЫ ОБЬ
T1M19DC DC      C' ПРИМЕЧАНИЕ: ВЫЗОВ КОМПОНЕНТА ВЫПОЛНЯЕТСЯ
DC      DC      C' ' ВВОДОМ ЕГО НОМЕРА, УКАЗАННОГО В ПЕРВЫХ
T1M20DC DC      C' ДВУХ ПОЗИЦИЯХ СТРОКИ. НОМЕР МОЖЕТ БЫТЬ 3
DC      DC      C' АДАН КУРСОРОМ ИЛИ КЛАВИШЕЙ: ПФ1 — ПФ12
T1M21DC DC      C' (1—12); ПД1 — ПД3 (13—15); ВЫЗОВ ТЕСТА (
DC      DC      C' 16). ПОДКОМАНДЫ «КОНЕЦ»,«К»,«Н»,«З»,«ПИ»
T1M22DC DC      C' ВЫПОЛНЯЮТСЯ АНАЛОГИЧНО ОДНОИМЕННЫМ
DC      DC      C' КОМАНДАМ ПОЛЬЗОВАТЕЛЯ ТЕРМИНАЛА.
END

```

RCSZST04 CSECT

```

DC      A(GM01),X '8' ,AL3(GI01),A(GS01),A(GE01),A(0)
DC      A(GM02),X '8' ,AL3(GI02),A(GS02),2A(0)
DC      A(GM03),X '8' ,AL3(GI03),3A(0)
DC      A(GM04),X'8' ,AL3(G104),3A(0)
DC      A(0)
GI01    DC      C'NATURAL'
GI02    DC      C'NATBATCH'
GI03    DC      C'GETPTS'
GI04    DC      C'MBM'
GS01    DC      CL80'S NATURAL'
GS02    DC      CL80' S NATBATCH'
GE01    DC      CL80 'F RCS,P NATURAL'
GM01    DC      C' РАБОТА С СИСТЕМОЙ NATURAL В РЕ
DC      C' ЖИМЕ ДИАЛОГА
GM02    DC      C' РАБОТА С СИСТЕМОЙ NATURAL В ПА
DC      C' КЕТНОМ РЕЖИМЕ
GM03    DC      C' КОПИРОВАНИЕ ФАЙЛОВ С МИНИДИСКОВ П Д О
DC      C'
GM04    DC      C' ПЕРЕДАЧА ТЕРМИНАЛА МОНИТОРУ С В М
DC      C'
END

```

Реализация санкционированного доступа к определенному подмножеству задач выполняется по правилам регламентированного доступа в систему, описанным в начале главы. Необходимо переименовать программу RCSTPAS1 в библиотеке загрузочных модулей системы ОБЬ в RCSTPASS

Таблица 2

Команда	Функция
	Команды управления сеансом
ИДПЛЗ, (ПАРОЛЬ) КОНЕЦ	идентифицирует пользователя терминала и начинает сеанс завершает сеанс
	Команды управления данными
З	вызывает начало формирования нового текста
П	употребляется для просмотра и обновления на месте текстов
К	используется для корректировки текстов
У	удаляет тексты из файла работ системы
	Команды обработки запросов
В	вызывает прикладные программы для исполнения
Р	распечатывает тексты
ЗА	применяется для объединения процедур формирования текста, вызова прикладной программы и последующего автоматического просмотра ответов
	Команды получения информации о состоянии ресурсов пользователя терминала
ТЛ	выводит на терминал таблицу локальных ресурсов системы
ОБЬ	
ТУ	отображает таблицу ресурсов удаленной ЭВМ
ПИ	показывает состояние текстов пользователя
	Команды передачи сообщений
С	посылает короткое сообщение (до 54 байтов) пользователям терминалов и операторов
СП	передает пользователям произвольные тексты
	Специальные команды
СТОП	вызывает отключение терминала от системы на заданное время
УФ	управляет форматом отображения информации на терминале
МВВ	переключает модули ввода-вывода
ТЗ	устанавливает таблицу задач

и составить список пользователей. Список строится как программная секция языка ассемблер и записывается в библиотеку системы под именем RCSUST01. Он имеет линейную структуру и состоит из 40-байтных строк, описывающих пользователей, которым разрешено работать с системой. Формат строки следующий (в скобках указана длина поля в байтах):

USER(8),PASSWORD(8),FAMILY(20),TASK(4)

USER — идентификатор пользователя, который задается в команде начала сеанса; PASSWORD — пароль; FAMILY — фамилия или любая другая информация, описывающая абонента; TASK — суффикс корневой таблицы задач, с которой предписано работать данному абоненту (если в этом поле указаны пробелы, то после ввода команды начала сеанса пользователь будет работать по обычному алгоритму диалога с системой).

Пример списка:

```
RCSUST01  CSECT
DC        CL8'US01',CL8'1',CL20 'ИВАНОВ И. И',CL4'ST01'
DC        CL8'US02',CL8'2',CL20 'ПЕТРОВ П. П.',CL4'ST02'
DC        CL8'US03',CL8'3',CL20'СИДОРОВ С С.',CL4' '
DC        CL8'US'04',L8'4',CL20'КУЗНЕЦОВ К. К.',CL4'ST04'
DC        8X'FF'
END
```

В табл. 2 приведен список команд пользователя терминала.

ПРИКЛАДНОЕ ПРОГРАММИРОВАНИЕ В СИСТЕМЕ ОБЬ

Пользователи системы ОБЬ имеют возможность составлять свои прикладные программы, к которым они могут обращаться с помощью команд обработки запросов. Включение готовых программ осуществляется персоналом, ответственным за эксплуатацию системы, формированием соответствующих элементов в списках прикладных программ. Списки строятся как программные секции языка ассемблер, затем они транслируются и с помощью редактора связей ОС ЕС записываются в библиотеку загрузочных модулей системы. Во время запуска указывается суффикс требуемого списка программ. Списки имеют линейную структуру, и на каждую программу в них отводится восемнадцать байтов: имя программы и место ее хранения (16 байтов), тип программы (1 байт), максимальное число одновременных обращений к программе со стороны пользователей терминалов (1 байт). Во время работы оператор может изменять состав и характеристики программ, доступных абонентам, с помощью команд динамической реконфигурации системы.

В прикладной программе можно использовать любые средства ОС ЕС, в том числе и средства взаимодействия с терминалом (например, БТМД). Дополнительно, система предоставляет два метода доступа:

—последовательный метод доступа (ПМД), для чтения и записи текстов, с которыми пользователь может работать с помощью команд управления текстами;

—терминальный метод доступа (ТМД), для организации интерактивного взаимодействия с абонентом без применения средств ОС ЕС, осуществляющих терминальный ввод-вывод в прикладной программе.

Программы могут быть пакетные и терминальные. Тип программы указывается в ее характеристиках в списке программ или в командах оператора.

Пакетные программы не ведут диалога с вызвавшими их пользователями. Они аналогичны пакетным заданиям ОС ЕС. Абонент вводит команду обращения к такой программе. Система сообщает ему, что вопрос принят или не принят к исполнению с указанием причины. Если запрос принят, то он ставится в очередь на обработку в соответствии с заданным приоритетом. После этого пользователь может вводить другие команды. По мере прохождения запроса в системе ему на терминал будут выводиться сообщения о начале обработки, окончании формирования программой текстов-ответов, конце обработки. Если к моменту поступления сообщения терминал занят, то оно будет записано, как только это будет возможно.

Терминальные программы ведут диалог со своими абонентами. Если пользователь обращается к такой программе, то после обработки команды вызова терминал передается под непосредственное ее управление. По окончании работы программы устройство возвращается системе и на него выводятся все сообщения, которые к этому моменту были приняты.

Любая программа может работать в зоне оперативной памяти системы ОБЬ и выполняться как самостоятельное задание ОС ЕС.

Прикладным программам, получающим управление от системы ОБЬ, через первый регистр передаются две адресные константы: адрес строки, в которой содержатся данные, заданные в операнде PARM; адрес строки параметров терминала.

Совпадение адресов означает отсутствие операнда PARM в команде вызова. Параметры терминала передаются всегда. Строки оформлены в виде записей переменной длины, выровненные на границу полуслова. Сначала идет двухбайтное поле счетчика данных, затем следуют данные. Например:

```
SAMPLE    CSECT
          USING      *,12
          STM        14,12,12(13)
          LR         12,15
          GETMAIN R, LV = 72
          ST         1,8(13)
          ST         13,4(1)
```

LM	13,1.8(13)	
LM	2,3,0(1)	Адреса строк параметров
CR	2,3	
BE	BRANCH	Операнд PARM не задан
LA	4,2(2)	Адрес начала PARM
LH	5,0(2)	Длина данных PARM
...		

Строка параметров терминала всегда в первых восьми байтах содержит имя оператора DD, описывающего устройство, с которого пришел вызов. Последующая информация зависит от типа терминала: списки опроса и выборки для многоточечного соединения, телефонный номер для коммутируемого сопряжения с автовызовом (АП-1) и др.

Приведем примеры строк параметров терминала;

```
RCST2LXX           дисплей ЕС-7920-01
RCSC25XX4130      дисплей ИЗОТ-7925 АП-62
RCSC1CXX413541312020585  комм. АП-1 с автовызовом
```

Параметры необходимы прикладным программам, которые взаимодействуют с терминалами с помощью методов доступа ОС ЕС, для открытия блока DCB. Такие диалоговые программы могут работать как под управлением системы ОБЬ, так и без нее.

Приведем в качестве примера фрагмент программы, написанной на ассемблере, которая может вести диалог только с пользователем дисплея локального комплекса ЕС-7920:

```
SAMPLE2L  CSECT
          USING      *,12
          STM        14,12,12(13)
          LR         12,15
          GETMAIN R,LV = 72
          ST         1,8(13)
          ST         13,4(1)
          LM         13,1,8(13)
          L          2,4(1)           Адрес парам.терминала
                                   0(2,2), = H'8'
          BH         ABEND           Длина парам. > 8
          CLC        5(3,2), = C'T2L'
          BNE        ABEND           Не дисплей ЕС-7927-01
          ...
          MVC        DCB+ 40(8) ,2(2)  Имя DD → в DCB ОС ЕС
          OPEN       MF = (E,OPENL)   Открыть DCB терминала
          ...
          WRITE      DECВ,TS,,BUF,1924,MF = E
          ...
ABEND     ABEND      5,DUMP
          END
```

Данная программа способна работать не только под управлением ОБЬ, но и в среде ОС ЕС (автономно). Для автономной работы задание на запуск SAMPLE2L, например, может выглядеть так:

```
//SAMPLE2L  JOB
//          EXEC   PGM = SAMPLE2L,PARM = RCST2L00
//RCST2L00  DD     UNIT = 7927-2
          ...
```

Адрес начала данных строки параметров терминала программа должна запомнить, если в ней используется ПМД или ТМД в явном виде (с помощью оператора CALL).

Для получения адресов начальных параметров в произвольном месте программы можно

воспользоваться оператором CALL с именем точки входа RCSPAR, у которой два аргумента. Это выровненные на границу слов два поля, куда RCSPAR пересылает адреса данных строк параметров. Оператор имеет следующий вид:

```
CALL RCSPAR, (PARM,PARS) — ассемблер  
CALL RCSPAR (PARM,PARS); — ПЛ/1.
```

PARM — адресная переменная, указывающая на параметры пользователя;

PARS — указатель на параметры системы.

Если абонент не задал параметры в команде вызова, то PARM = PARS.

Пример программы на ПЛ/1:

```
TESTPRM:   PROC   OPTIONS(MAIN);  
           DCL   BUFPRM CHAR (80) BASED (PARM),  
           BUFRCS CHAR(80) BASED(PARS);  
           ...   /* Произвольная обработка */  
           CALL  RCSPAR (PARM,PARS);  
           ...
```

Прикладная программа может идентифицировать инициатора работы с помощью оператора CALL RCSIDP. Он имеет один аргумент: указатель на 80-байтный буфер обмена с системой ОБЬ. В буфере первоначально находятся параметры терминала. После выполнения оператора в нем записаны следующие идентификаторы: процесса прикладной программы (8 байтов); номера запроса (1 байт); локального процесса инициатора работы (8 байтов); тип запроса (1 байт: Т — если вызов пришел с локального терминала, Э — вызов с другой ЭВМ); абонента терминала (4 байта) или процесса другой ЭВМ (8 байтов).

Вывод произвольной текстовой информации длиной до 80 байтов на терминал пользователя осуществляется оператором CALL RCSWTS. Он имеет аналогичный аргумент, что и оператор CALL RCSIDP. Например, если в предыдущем примере после оператора CALL RCSPAR поставить

```
CALL RCSIDP(PARS);  
CALL RCSWTS(PARS);
```

то на терминал абонента программы в случае запроса с терминала местной ЭВМ в первую свободную строку будет записано:

```
PCS000I ИДПРОГ: ИДПРОГ Н ИДТЕРМ Т ИДПЛ
```

в случае вызова с другой ЭВМ:

```
RCS000I ИДПРОГ: ИДПРОГ Н ИДЭВМ Э ИДТЕРМ
```

Определить имя модуля ввода-вывода, который обслуживает терминал пользователя, можно с помощью оператора CALL RCSMIO. Оно помещается в первые восемь байтов буфера обмена. Необходимость в таком обращении к системе из прикладной программы может быть вызвана, например, если программа ведет разный диалог с абонентом в зависимости от типа терминала. В системе ОБЬ тип терминала идентифицируется именем модуля ввода-вывода.

Последовательный метод доступа (ПМД). Он позволяет прикладной программе обрабатывать данные пользователя терминала, которые находятся в файле работ системы и оформлены в виде текстов.

ПМД обеспечивает следующее:

—чтение текста, который указан в команде обращения к программе, как последовательного набора 80-байтных записей:

—формирование новых текстов, которые называются ответами и аналогичны по структуре входным текстам;

—переадресацию ответов другим прикладным программам или процессам управления печатью.

Чтение информации с помощью ПМД соответствует чтению, например, набора данных с перфокарт во входном потоке задания ОС ЕС в пакетном режиме.

Запись информации с помощью ПМД аналогична выводу, например, набора данных на АЦПУ через выходную очередь ОС ЕС. При этом размер строки ограничен 80 байтами, так как ответ предназначен в первую очередь для просмотра с терминалов.

Переадресация ответов другим процессам системы ОБЬ отдаленно напоминает конвейер в ОС UNIX и ДОС ПЭВМ.

Преимущество в использовании ПМД системы ОБЬ по сравнению с ПМД ОС ЕС в прикладных программах заключается, главным образом, в том, что обеспечивается терминальный доступ к текстам и поддерживаются средства их модификации, которые, благодаря списочной структуре файла работ, достаточно эффективны.

Непосредственное применение ПМД системы ОБЬ основано на двух операторах CALL RCSGET и CALL RCSPUT. Они имеют аргумент, аналогичный предыдущим операторам CALL RCSIDP, CALL RCSWTS, CALL RCSMIO.

По CALL RCSGET происходит чтение входного текста. В результате выполнения оператора в буфер обмена с системой помещается следующий блок текста или признак его окончания. Признаком окончания текста является равенство первого байта буфера обмена значению «FF» (в шестнадцатеричном формате). Он помещается в буфер при попытке читать после получения последнего блока текста или при первом обращении к входным данным, в случае отсутствия атрибутов текста в команде вызова программы.

Приведем пример программы на ПЛ/1 с ПМД:

```
TESTSAM:   PROC OPTIONS(MAIN);
           DCL (BUF CHAR(80), EOT ВПЧ8) BASED(PARS);
           CALL RCSPAR(PARM,PARS);
           CALL RCSGET(PARS);
           /* Читать 1-й блок текста*/
           DO WHILE (EOT* = (8)'1'B);
           /* Делай пока не конец текста */
           ... /* Произвольная обработка */
           CALL RCSGET(PARS);
           /* Читать следующий блок запроса */
           END;
END TESTSAM;
```

Если в команде вызова программы пользователь указал неправильные атрибуты текста, то первое же применение в программе оператора CALL RCSGET приведет к ее аварийному завершению с пользовательским кодом, равным пяти. Попытка читать текст после его конца также даст аварийное завершение работы программы с тем же кодом. Так, если в предыдущем примере после группы «DO-WHILE» поставить оператор CALL RCSGET, то произойдет аварийное завершение работы программы:

```
DO WHILE(EOT' = (8)'1'B);
...
END;
CALL RCSGET (PAR);
/* Аварийное завершение работы программы */
```

Пользователь терминала в этом случае получит сообщение:

```
RCS101I ЗАПРОС/ОТВЕТ XXXX НОМЕР NNNNN
      ВЫПОЛНЕН XXXXXXXXX U0005
```

XXXX — идентификатор текста (запроса или ответа); NNNNN — номер текста; XXXXXXXXX — имя прикладной программы.

Формирование выходных текстов выполняется с помощью оператора CALL RCSPUT. Данные помещаются в буфер обмена и по CALL RCSPUT дописываются в ответ. После окончания формирования одного ответа можно создавать следующий. Признаком конца является запись блока специального вида: первый байт содержит «FF», а остальные 79 байтов

— пробелы. В этом случае информация о сформированном тексте будет выведена на терминал пользователя. Если после байта «FF» программа поместила данные следующего вида:

```
ИДПРОЦ[,T] [,PARM = ...]
```

где ИДПРОЦ — идентификатор процесса-адресата; T — признак временного текста; PARM — параметры; то ответ будет направлен не абоненту терминала, а процессу, идентификатор которого совпадает в ИДПРОЦ. Как правило, адресат — это либо процесс управления прикладной программой, либо процесс документирования данных. В первом случае, такая передача управления соответствует вызову прикладной программы со стороны другой программы. Во втором — выводу данных с помощью процессов печати, инициированному прикладной программой.

Программа, запросившая конвейерную обработку данных, должна проанализировать код возврата, помещаемый в первый байт буфера обмена. Он может принимать значения (шестнадцатиричный формат): 00 — синтаксическая ошибка; 04 — нет процесса-адресата; 08 — файл работ переполнен; FF — нормальное завершение операции.

Приведем пример программы с конвейерной передачей данных:

```
TESTCON: PROC OPTIONS (MAIN, REENTRANT);
          DCL BUFPRM CHAR (80) BASED (PARM),
            BUFRCS CHAR(80) BASED(PARS),
            EOT BIT(8) BASED(PARS);
          CALL RCSPAR(PARM,PARS);
          CALL RCSGET(PARS);
          DO WHILE(EOT' = (8)'1'B);
            CALL RCSPUT(PARS);
            CALL RCSGET(PARS);
          END;
          SUBSTR(BUFRCS,2) = BUFPRM;
          CALL RCSPUT(PARS);
END TESTCON;
```

Эта программа копирует входной текст в ответ, который переадресуется процессу, заданному в поле PARM команды вызова TESTCON. Например, ввод команды

```
TESTCON,2,PARM = '00F,T'
```

приведет к тому, что входной текст (запрос) данного абонента с номером два будет скопирован во временный ответ и распечатан на устройстве с адресом 00F. Подразумевается, что в системе есть процесс печати с идентификатором 00F. После вывода ответ удаляется из файла работ.

Система оповещает абонента терминала о сформированном ответе после окончания его формирования в прикладной программе или после завершения работы программы. В область листинга терминала записывается сообщение:

```
RCS105I ПРОГРАММА XXXXXXXX СФОРМИРОВАЛА ОТВЕТ
XXXX НОМЕР NNNNN
```

XXXXXXXX — имя прикладной программы; XXXX — идентификатор ответа; NNNNN — номер ответа.

Если программа была вызвана с помощью команды «ЗА», то система попытается сразу показать содержимое ответа. Если терминал находился к этому моменту в режиме листинга, то он будет переведен в режим фрейма и в области фрейма появится начало текста. Телемонитор выполнил команду просмотра текста на данном устройстве. Если на терминале в это время осуществлялась работа с текстом или диалоговой прикладной программой, то по ее окончании на устройство будет выведено одно из следующих сообщений:

```
RCS000I ПРИШЕЛ ОТВЕТ XXXX НОМЕР NNNNN ОТ XXXXXXXX
RCS000I ПРИШЛО ПИСЬМО NNNNN/0/XXXX ОТ
ИДЭВМ(XXXXXXXX)
```

XXXXXXXX — имя прикладной программы; ИДЭВМ — идентификатор ЭВМ, где работает программа; XXXX — идентификатор ответа; NNNNN — номер ответа.

Терминальный метод доступа (ТМД). Он позволяет прикладной программе выполнять интерактивное взаимодействие с пользователем без применения средств ОС ЕС для терминального ввода-вывода.

ТМД обеспечивает: инициализацию устройства; чтение и запись данных; отключение терминала от программы; установку интервала времени, в течение которого будут ожидать запросы на чтение; идентификатор внимания и адрес курсора; прием сигнала прерывания с терминала и его передачу программе, не выполнявшей в момент возникновения сигнала операций ввода-вывода на устройстве.

ТМД может работать в двух режимах: фрейм и листинг.

В режиме фрейма обмен осуществляется блоками, вообще говоря, произвольной длины. Имеются шесть способов выполнения операций ввода-вывода:

—чтение данных, начиная с левого верхнего угла экрана дисплея, после нажатия клавиши «ВВОД»;

—чтение данных с начала экрана, без ожидания нажатия клавиши «ВВОД»;

—чтение информации, расположенной в произвольном месте экрана (для разных терминалов технология ввода данных отличается, например: для дисплеев типа ЕС-7920 — это чтение модифицированных полей; для ЕС-7066, АП-62, АП-64 — это передача кодов между старт- и стоп-символами; для терминалов типа СМ — это ввод с позиции курсора);

—запись с начала экрана с предварительной очисткой;

—запись без стирания экрана, начиная с левого верхнего угла или с позиции курсора;

—ввод-вывод данных в формате ЕС-7920.

В режиме листинга диалог с программой основан на 80-байтном обмене данными с сохранением протокола работы и обеспечении определенного управления взаимодействием со стороны ТМД.

Ввод данных выполняется после появления подсказывающих сообщений, например:

RCS109A ВВОДИТЕ КОМАНДУ

из поля, начало которого отмечается курсором или положением каретки. Данные дополняются пробелами до 80 байтов.

Выводимая информация (до 80 байтов) помещается в область листинга (протокол обмена). Она отображается на экране дисплея.

Управление листингом и вывод сообщений, разрешающих ввод команд, обеспечивается ТМД. Управление листингом позволяет выполнять чистку области листинга (команда «К») и снятие прикладных программ (команда «СТОП»).

По заполнению области листинга вывод данных прекращается и появляется:

RCS110A СООБЩЕНИЕ ОЖИДАЕТ ВЫВОДА

Для очистки области листинга и продолжения работы программы необходимо нажать клавишу «ВВОД».

Преимущество в применении ТМД системы ОБЪ по сравнению с аналогичными средствами ОС ЕС заключается в том, что обеспечивается независимость прикладных программ от типов устройств. От программиста не требуется знания особенностей работы терминального оборудования.

Явное применение ТМД реализуется с помощью операторов CALL. Первый аргумент всегда должен указывать на буфер обмена с системой — как в предыдущих операторах.

Инициализация устройства или открытие ТМД в прикладной программе выполняется либо автоматически, при первой операции чтения или записи, либо принудительно, если применяются операторы CALL RCSTOPN или CALL RCSOPNH. Они имеют один аргумент. Отличие в использовании операторов состоит в следующем. Открытие ТМД по CALL RCSTOPN всегда приводит к работе с терминалом пользователя, вызвавшего программу. ТМД использует данные для инициализации области управления вводом-выводом из блока параметров терминального процесса абонента. По CALL RCSOPNH информация, требуемая для настройки этой области, берется из буфера обмена с системой, содержимое которого

подготавливается прикладной программой. Это позволяет средствами ТМД выполнять ввод-вывод на другие устройства, что имеет смысл для многокомпонентных терминалов, например, АП-4.

Если в программе открытие ТМД явно не указано, то перед первой операцией ввода-вывода выполняется инициализация устройства с помощью CALL RCSTOPN.

Приведем фрагмент программы, выполняющей открытие ТМД:

```

TESTAM CSECT
        USING    *,12
        STM      14,12,12(13)
        LR       12,15
        LR       15,13
        LA       13,SAVE
        ST       13,8(15)
        ST       15,SAVE + 4
        LM       2,3,0(1)           R1 → на параметры ОБЬ
        CR       2,3               Указан PARM в команде
        LA       3,2(3)           вызова программы?
        ST       3,PARS
        BE       TOPN             Нет
        LH       4,0(2)           R4 = кол. символов PARM
        LTR      4,4
        BNP      TOPN             Нет данных параметров
        BCTR     4,0               Перенести информацию из
        MV1     0(3),C' '         параметров в буфер
        MVC     1(79,3),0(3)     обмена ОБЬ
        EX      4,MOVE
        CALL    RCSOPNH,PARS     Открыть ТМД
        B      EXEC
TOPN   CALL    RCSTOPN,PARS     Открыть ТМД
EXEC   EXEC
MOVE   MVC     0(0,3),2(2)
SAVE   DS      18F
PARS   DS      A                Адрес буфера обмена с системой
        END

```

После выполнения открытия ТМД в первых двух позициях буфера обмена находится двоичное число, равное размеру экрана дисплея в байтах или длине буфера для терминалов телетайпного типа, например 120 байтов для АП-1.

Отключение терминала или закрытие ТМД в прикладной программе происходит автоматически при завершении работы или принудительно, если применяются операторы CALL RCSTCLS или CALL RCSTEND. Передаваемый аргумент такой же как и при открытии ТМД. В результате, выполнения оператора CALL RCSTEND управление устройством возвращается терминальному процессу абонента, а по CALL RCSTCLS — нет. В последнем случае возврат терминала выполняется после окончания работы программы. Применяя CALL RCSTEND, программист может так спланировать работу прикладной программы, что сначала она ведет диалог с абонентом, а затем отключается от устройства и продолжает работать как программа пакетного типа, то есть пользователь может продолжать взаимодействие с системой асинхронно с выполнением программы.

Чтение данных в режиме фрейма реализуется с помощью трех операторов CALL: RCSTGET, RCSGETH и RCSGETB. Все они имеют три аргумента: указатель на буфер обмена с системой; адрес области приема данных; двухбайтное поле, содержащее двоичное число считываемой информации. Например:

```
CALL RCSTGET,(PARS,ADAT,SDAT)
```

ADAT — адресная переменная, указывающая на область памяти, куда требуется поместить данные, принятые с терминала;

SDAT — двоичное полуслово, где находится длина запрашиваемых данных. После выполнения операции туда помещается фактическая длина считанной информации.

Отличия в использовании операторов чтения заключаются в следующем. По CALL RCSTGET и RCSGETH прикладная программа переходит в состояние ожидания до тех пор, пока абонент терминала не подготовит необходимую информацию и не нажмет клавишу «ВВОД» или не закончится интервал времени ожидания ввода данных. С помощью RCSGETB программа считывает информацию вне зависимости от действий пользователя за терминалом. Эта возможность обеспечивается ТМД только для дисплеев. Для печатающих терминалов операция ввода не запускается и управление возвращается прикладной программе с нулевым значением в поле длины принятых данных. После нажатия клавиши «ВВОД» по CALL RCSTGET информация будет считываться из левого верхнего угла экрана дисплея, а по CALL RCSGETH — из произвольного места экрана в зависимости от типа устройства: с начала первого модифицированного поля (ЕС-7920 и его аналоги); после старт-символа (АП-62, АП-64, ЕС-7066); с позиции курсора (СМ-1604, СМ-1611, СМ-7209 и др.). Для печатающих терминалов ввод данных всегда выполняется с первого набранного кода.

Приведем пример использования оператора CALL RCSTGET в программе на ПЛ/1:

```
TESTAM:   PROC OPTIONS(MAIN);
          DCL BUF CHAR(80) BASED(PARS),
          LEN BIN FIXED(15) INIT(80);
          CALL RCSPAR(PARM,PARS);
          CALL RCSTGET(PARS,PARS,LEN);
          /* Читать 80 байтов с терминала */
          /* в буфер обмена с системой ОБЪ */
```

END TESTAM;

Вывод информации в режиме фрейма осуществляется с помощью двух операторов CALL: RCSTPUT и RCSPUTH. Они используют те же аргументы, что и операторы чтения. Второй и третий аргументы, естественно, определяют блок записываемых данных. Они выводятся на терминал, начиная с левого верхнего угла экрана дисплея или с начала новой строки пишущей машинки. Экран предварительно стирается, если применяется оператор CALL RCSTPUT. Информация может содержать символы форматирования данных, например, коды указаний и символы-атрибуты для дисплея типа ЕС-7920.

Приведем пример использования оператора CAIX RCSTPUT в программе на ассемблере:

```
TESTAM   CSECT
         USING   *,12
         SAVE   (14,12)
         LR     12,15
         LR     15,13
         LA     13,SAVE
         ST     13,8(15)
         ST     15,SAVE + 4
         L     2,4(1)           Запомнить адрес буфера
         LA     2,2(2)         обмена с системой ОБЪ.
         ST     2,PARS
         CALL   RCSTPUT, (PARS,ADAT,SDAT)
         ...
SAVE     DS     18F
PARS     DS     A
ADAT     DC     A(DATA)       Адрес данных
SDAT     DC     Y(DATAEND-DATA)  Длина данных
DATA     DC     C' ПРОВЕРКА ВЫВОДА ИНФОРМАЦИИ'
```

```

DC      C' НА ТЕРМИНАЛ С ПОМОЩЬЮ'
DC      C' ТЕРМИНАЛЬНОГО МЕТОДА ДОСТУПА'
DC      C' СИСТЕМЫ ОБЬ'
DATAEND EQU  *
END

```

В режиме листинга чтение и запись данных реализуется с помощью операторов CALL RCSGETL и RCSPUTL, имеющих один аргумент. Команда считывается из поля, указанного курсором, дополняется пробелами до 80 байтов и помещается в буфер обмена. При записи данные из буфера обмена выводятся на терминал в область листинга.

Обеспечение ввода-вывода данных в формате ЕС-7920 является важным компонентом ТМД. С помощью этого средства можно адаптировать программы, предназначенные для работы с дисплеями ЕС-7920, на другие типы терминалов, имеющих размеры экранов 24×80. Прикладные программы, написанные с расчетом на обработку данных, представленных в формате ЕС-7920, с помощью системы ОБЬ могут взаимодействовать, например, с терминалами серии СМ. Имитация и интерпретация информационных потоков ЕС-7920 выполняется ТМД.

Реализация этой функции осуществляется с помощью оператора CALL RCS7920. Он, кроме описанных выше аргументов, имеет еще один — выровненное на границу полуслова двухбайтное поле, содержащее условный код операции. Всего имеется пять кодов, которые соответствуют основным операциям ввода-вывода терминала ЕС-7920:

0— считывание модифицированных полей, после нажатия одной из функциональных клавиш (ее аналогом в БТМД ОС ЕС является операция READ TI);

1— прием сообщений с дисплея независимо от действий пользователя. При помощи этой операции считываются все модифицированные поля (READ TM);

2— чтение содержимого всей буферной памяти устройства, включая модифицированные и немодифицированные поля, символы-атрибуты и нули (READ TB);

3— запись без стирания (WRITE TI);

4— запись со стиранием (WRITE TS);

Данные для оператора CALL RCS7920 могут содержать управляющие коды: «УСТАНОВИТЬ АДРЕС БУФЕРА», «НАЧАЛО ПОЛЯ», «УСТАНОВИТЬ КУРСОР», «ПОВТОРИТЬ ДО АДРЕСА» и должны начинаться с символа управления записью в формате ЕС-7920.

ТМД позволяет прикладной программе устанавливать интервал времени ожидания ввода данных. Он начинает отсчитываться в момент запуска каждой операции чтения. Если данные не были введены с терминала за указанный промежуток времени, то управление вернется программе. Отрицательное значение в поле длины (режим фрейма) или признак «FF» (в шестнадцатеричном формате в первой позиции буфера обмена для режима листинга) сигнализирует программе, что закончился интервал времени. Установка значений временного промежутка выполняется с помощью оператора CALL RCSSTM. У него один аргумент. Значение в секундах в двоичном формате заносится в первые два байта буфера обмена I с системой. Нулевое значение снимает режим отсчета времени. Каждый последующий запрос на ввод данных в этом случае будет заканчиваться только по завершению операции чтения. Оператор может указываться в любом месте прикладной программы. Его действие сводится к занесению значения интервала времени в управляющую область ТМД. Если в программе нет данного оператора, то отсчет времени не производится.

Получение идентификатора сигнала прерывания (нажатие клавиши «ВВОД» или ее аналога) и адреса курсора после приема данных реализуется в ТМД с помощью оператора CALL RCSSAD. Он имеет два аргумента-указателя на буфер обмена и область памяти, куда требуется записывать информацию после каждой операции чтения. Область памяти должна быть длиной не менее шести байтов и выровнена на границу полуслова. В первые три позиции записываются идентификатор сигнала прерывания и адрес курсора в формате ЕС-7920, четвертый байт не используется, а пятый и шестой байты содержат двоичное смещение курсора на экране. Например, нулевое смещение соответствует левому верхнему углу экрана. Оператор может указываться в любом месте программы. Его действие сводится к

запоминанию терминальным методом доступа адреса области, специфицированной вторым аргументом. Если определить, «нулевой» указатель, то пересылка идентификатора и адреса курсора выполняться не будет.

Режим асинхронного приема сигнала прерывания от пользователя терминала устанавливается с помощью оператора CALL RCSTAI. Его синтаксис аналогичен оператору CALL RCSSAD. Единственное отличие состоит в том, что минимальная длина области данных равна четырем байтам. Прикладная программа может передать терминальному методу доступа через первый байт области идентификатор функциональной клавиши в формате ЕС-7920, при нажатии которой будет выдаваться сигнал прерывания. ТМД в результате выполнения оператора помещает в область данных блок управления событием ECB ОС ЕС. Он будет отмечен в момент возникновения асинхронного прерывания. Программа может ожидать возникновения события, выдав макрокоманду WAIT, или периодически проверять значение бита завершения — 1-й бит блока ECB,— если он равен единице, то произошло ожидаемое событие. Сброс признака возникновения события осуществляется ТМД перед возвратом управления после выполнения запросов на ввод-вывод. При нулевом значении второго аргумента режим асинхронного приема сигнала прерывания отключается.

Приведем пример программы, иллюстрирующей работу ТМД в режиме листинга с использованием ожидания ввода данных определенное время и асинхронного приема сигнал прерывания. Программа предназначена для проверки работоспособности минитерминала СМ-1605.М1. Алгоритм работы заключается в следующем. Ожидается в течение 30 секунд ввод произвольного действительного числа. Если оно не было определено, то выводится сообщение:

----- T

Если число было введено синтаксически неправильно, то на терминал записываются ошибочные данные, за которыми следуют символы «пробел» и «Е». В случае правильного ввода информации программа начинает периодически проверять область памяти, где должно быть отмечено ТМД появление сигнала прерывания от клавиши «ВВОД». Поэтому ответ от программы можно получить только нажав эту клавишу. Он состоит из искомого числа — в формате F(13,5),— за которым следует символ «*».

Окончание работы программы выполняется при нажатии клавиши «ВВОД» без предварительного ввода данных. Записывается сообщение:

----- #K

```
SER1605: PROC OPTIONS (MAIN.REENTRANT);
          DCL BUF    CHAR(80) BASED(PARS),
          STR      CHAR(80),
          TAIB     BIT(32) INIT((32)'0'B),
          TAIP     PTR,
          TIME     BIN FIXED BASED(PARS),
          TIMEF    BIT(8) BASED(PARS),
          IFLAG    INIT(0),
          X        DEC FLOAT9(16);
ON CONVERSION
BEGIN;
  BUF = ' ';
  SUBSTR(BUF,1,13) = ONSOURCE;
  SUBSTR(BUF,14,2) = 'E';
  ONSOURCE = '0';
  IFLAG = 1;
END;
CALL RCSPAR (PARM, PARS);
TAIP=ADDR (TAIB);
CALL RCSTAI (PARS, TAIP);
TIME = 30;
```

```

CALL RCSSTM (PARS);
DO WHILE (IFLAG^ = 2);
  CALL RCSGETL (PARS);
  IF TIMEF = (8) '1' B THEN
    DO;
      BUF= (13)'-';
      SUBSTR (BUF, 14,1) = 'T';
    END;
  ELSE
    IF BUF = ' ' THEN
      DO;
        BUF = (13)'-';
        SUBSTR (BUF, 14,2) = '#K';
        IFLAG = 2;
      END;
    ELSE
      DO;
        STR = BUF;
        IFLAG=0;
        GET STRING (STR) LIST(X);
        IF IFLAG = 0 THEN
          DO;
            STR = ' ';
            PUT STRING (STR) EDIT (X, '*')
              (F(13,5), A);
            BUF = STR;
            DO WHILE
              (SUBSTR(TAIB, 1,8) = (8)'0'B);
              DELAY (3000);
            END;
          END;
        END;
      END;
    CALL RCSPUTL(PARS);
  END;
END SER1605;

```

Неявное использование ПМД и ТМД в прикладных программах. В прикладных программах, написанных на любом языке программирования, допускается использование операторов последовательного ввода-вывода ОС ЕС для чтения-записи текстов и интерактивного взаимодействия с использованием терминалов. Нет ограничений на применение спецификаторов форматов и других опций в этих утверждениях. Операторы DD файлов должны иметь следующий вид:

```
//XXXXXXXX DD DUMMY, LABEL = YYY
```

XXXXXXXX — имя оператора DD (не должно начинаться с префикса RCS); YYY — код метода доступа системы ОБЬ (100 — ПМД, 200 — ТМД-фрейм, 300 — ТМД-листинг).

Необходимо указывать специальные операторы DD:

```
//RCSLS DD DUMMY — определяет ПМД в системе;
```

```
//RCSLTF DD DUMMY — определяет ТМД-фрейм;
```

```
//RCSLTL DD DUMMY — определяет ТМД-листинг.
```

Отсутствие этих операторов приводит к стандартным действиям ОС ЕС, выполняемым для «макетных» (DUMMY) наборов данных. Допускается употребление специальных операторов для ввода-вывода в прикладных программах. Рекомендуется использование файлов с записями фиксированной длины 80 байтов.

При неявном использовании методов доступа системы в программах, написанных на ПЛ/1, следует учитывать, что запись заносится в файл при выполнении следующего оператора PUT (WRITE).

При необходимости немедленно вывести запись (последняя строка текста ответа или данные на терминал) следует использовать оператор PUT с опцией SKIP или дополнительный оператор WRITE. Сказанное относится и к программам на ассемблере, использующим метод доступа QSAM в режиме указания.

В случае неявного ТМД в режиме фрейма система отводит буфер памяти, равный размеру экрана дисплея. Весь терминальный ввод-вывод выполняется через него. 80-байтные строки записываются и считываются с буфера. Данные выводятся на терминал при заполнении буфера или при выполнении запроса на ввод — экран предварительно очищается. Этот запрос переводит прикладную программу в состояние ожидания до тех пор, пока абонент не подготовит данные и не нажмет клавишу «ВВОД». Информация считывается с терминала в буфер и из него передается в программу по мере поступления операторов чтения. По исчерпанию данных из буфера программа снова переводится в состояние ожидания ввода данных.

Ввод команды «/*» в режиме ТМД-листинг вызывает состояние «КОНЕЦ ФАЙЛА» в прикладной программе.

Программа может воспользоваться оператором CALL RCSLTB для получения размера буфера терминала. В нем указывается один аргумент (двоичное полуслово), куда помещается длина буфера, кратная 80 байтам. Например (ПЛ/1):

```
DCL N BIN FIXED(15); CALL RCSLTB(N);
```

Приведем пример фортран-программы с неявным использованием методов доступов. Она строит таблицы биоритмов. Входные данные состоят из последовательности 80-байтных записей следующего вида:

```
PP.ДД.ММ.ГГ
```

PP — расчетный год (последние две цифры);

ДД.ММ.ГГ — день, месяц и год рождения.

По каждой записи программа выводит таблицу биоритмов. Она может работать в любых комбинациях режимов: ПМД системы ОБЬ и ОС ЕС; ТМД-фрейм и ТМД-листинг. Все зависит от определения операторов DD с именами FT05F001 и FT06F001.

```
INTEGER NN(15), NDG(12)
1 /0,31, 59, 90, 120, 151, 181, 212, 243, 273, 304, 334/,
2 NDM(12)/30, 27, 30, 29, 30, 29, 30, 30, 29, 30, 29, 30/,
3 MFC (36)
4 / ' ЯН','ВАРЬ',' ' ' ФЕ','ВРАЛ',' Б ' ,
5 'МА','РТ ' ' ' АП','РЕЛЬ',' ' ,
6 'МА','Й ' ' ' ИЮ','НЬ ' ' ' .
7 ' ИЮ','ЛЬ ' ' АВ','ГУСТ',' ' ' ' СЕ',
8 НТЯБ','РЬ ' ' ОК','ТЯБР','Ь ' ' НО'
9 ЯБРЬ ' ' ' ДЕ','КАБР','Ь ' /,
A FRM(35)/('04X','1Н','3А4','1Н',
B 30*'2Н ' ') ' /,FR1','2X','/FR2','12,'/
1 FORMAT(2I,3(IX,2I1))
2 FORMAT(3(/),8X,'19',2I1,17X,
1 'КРИТИЧЕСКИЕ ДНИ ЦИКЛА:'//
2 6X,2I1,2(/),2I1),8X, 'ФИЗИЧЕСКОГО', 6X,
3 'ЭМОЦИОНАЛ. ИНТЕЛЛЕКТ. Ф-Э Ф-И Э-И !!!')
3 FORMAT (2(/))
16 READ(5,1,END = 4)(NN(I),I = 1,8)
IF(NN(1) + NN(2).EQ.0)GO TO 4
WRITE(6,2)(NN(I),I= 1,8)
```

```

DO 5 I = 1,4
5  NN(8 + I) = I0*NN(2*I-1) + NN(2*I)
   NG = NN(9)
   NR = (NN(12)-1)*365 + (NN(12)-1)/4 +
+ NDG(NN(11)) + NN(10)
   IF(4*(NN(12)/4).EQ.NN(12).AND.NN(11).GT2)
* NR = NR + 1
   DO 6 NM = 1,12
   NF = 1
   NE = 6
   NI = 9
   DO 7 I = 1,15
7  NN(I)=0
   ND1 = (NG-1) *365 + (NG- 1)/4 + NDG(NM) + 0
   IF(4*(NG/4).EQ.NG.AND.NM.GT.2)ND1 = ND1 + 1
   ND2 = ND1 + NDM(NM)
   IF(4*(NG/4).EQ.NG.AND.NM.EQ.2)ND2 = ND2+1
   DO 8 ND = ND1,ND2
   I = 10*(ND-NR)
   IF(115*(I/115).NE.I.AND.115*((I + 5)/115).NE.
* (I + 5).AND.115*((I-5)/115).NE.(I-5))GO TO 9
   NN(NF) = ND-ND1 + 1
   NF = NF + 1
9  IF(140*(I/140).NE.I)GO TO 10
   NN(NE) = ND-ND1 + 1
   NE = NE + 1
10 IF(165*(I/165).NE.I.AND.165*((I + 5)/165) .NE.
* (I + 5).AND.165*((I-5)/165).NE.(I-5))GO TO 8
   NN(NI) = ND-ND1 + 1
   NI = NI+1
8  CONTINUE
   DO 11 I= 1,5
   DO 12 J = 6,8
12 IF(NN(I).EQ.NN(J).AND.NN(I).NE.0)NN(12) = NN(I)
   DO 11 J = 9,11
11 IF(NN(I).EQ.NN(J).AND.NN(I).NE.0)NN(13) = NN(I)
   DO 13 I = 6,8
   DO 13 J = 9,11
13 IF(NN(I).EQ.NN(J).AND.NN(I).NE.0)NN(14) = NN(I)
   IF(NN(12).EQ.NN(13))NN(15)=NN(12)
   IF(NN(12).EQ.NN(14).OR.NN(13).EQ.NN(14))
* NN(15) =NN(14)
   ND1 = 15
   ND = 15
41 IF(NN(ND).EQ.0)GO TO 40
   FRM(ND*2 + 3) = FR2
42 ND = ND-1
   IF(ND)17,17,41
40 ND1=ND1-1
   FRM(ND*2 + 3) =FR1
   IF(ND.GT.ND1)GO TO 42
   DO 43 I = ND,ND1
43 NN(I)=NN(I + 1)
   GO TO 42

```

```

17  WRITE (6,FRM)
    * (MEC(3*(NM-1) +I),I= 1,3), (NN(I),I=1,ND1)
6   CONTINUE
    WRITE (6,3)
    GO TO 16
4   STOP
    END

```

В заключение данной главы в табл.3 приведен перечень операторов прикладного программирования системы ОБЬ.

Таблица 3

Оператор	Функция
Операторы общего назначения	
RCSPAR	передает адреса параметров системы
RCSIDP	записывает в буфер обмена идентификаторы прикладной программы, терминала, пользователя
RCSMIO	помещает в буфер обмена имя терминального модуля ввода-вывода
RCSWTS	выводит сообщение на терминал абонента
Операторы последовательного метода доступа	
RCSGET	читает записи входного текста
RCSPUT	записывает данные в ответ или инициирует конвейерную обработку
Операторы терминального метода доступа	
RCSTOPN	инициализирует устройство и область управления вводом-выводом, исходя из данных блока параметров терминального процесса пользователя
RCSOPNH	выполняет действия, аналогичные предыдущему оператору, но на входных данных, подготовленных прикладной программой
RCSTCLS	закрывает ТМД
RCSTEND	закрывает ТМД и отсоединяет устройство от программы
RCSTGET	вызывает чтение данных, начиная с левого верхнего угла экрана дисплея или с начала буфера терминала после приема сигнала окончания ввода
RCSGETH	читает информацию из произвольного места экрана дисплея в зависимости от типа терминала после нажатия клавиши «ВВОД»
RCSGETB	выполняет чтение буфера терминала, не дожидаясь сигнала окончания ввода
RCSGETL	вводит команды в режиме ТМД-листинг
RCSTPUT	записывает данные, начиная с левого верхнего угла экрана дисплея с предварительным стиранием
RCSPUTH	выводит информацию с начала экрана без его чистки
RCSPUTL	записывает сообщения в режиме ТМД-листинг
RCS7920	обеспечивает ввод-вывод в формате ЕС-7920
RCSSTM	устанавливает интервал времени ожидания приема данных с терминала
RCSSAD	задает режим приема идентификатора клавиши окончания ввода и адреса курсора
RCSTAI	выполняет запуск асинхронного приема сигнала прерывания от абонента
RCSLTB	возвращает размер буфера, выделяемого ТМД прикладной программе с неявным способом выполнения операторов ввода-вывода

ГЛАВА 9

ИНТЕРАКТИВНОЕ ВЗАИМОДЕЙСТВИЕ С СУБД

9.1. СУБД ADABAS и диалоговая система NATURAL

Ключевым вопросом при формировании базы данных является вопрос о выборе и адаптации соответствующей системы управления базами данных (СУБД) из числа существующих. От решения этого вопроса зависят как возможности пользователя при работе с базой данных, так и состав, и объем программного обеспечения более высокого уровня. Широко распространенная в настоящее время СУБД ADABAS является, по совокупности своих характеристик, оптимальной для построения баз данных. Основные критерии такого выбора:

- надежность СУБД ADABAS при работе с базами данных большого объема (сотни Мбайт);
- обеспечение быстрого многофакторного поиска данных в базе;
- наличие развитых средств формирования связей между базами данных для их интеграции в единый банк.

Некоторая громоздкость средств СУБД ADABAS при работе с оперативно меняющимися базами данных в данном случае несущественна, так как многочисленные базы данных являются, по преимуществу, архивными.

Следует заметить, что если заполнение базы данными ведется, в основном, в пакетном режиме, то для выбора данных из базы по запросу пользователя значительно более эффективен режим диалога. В диалоговом режиме пользователь может оперативно составлять и корректировать запросы, просматривать отобранные данные и обрабатывать их с помощью проблемно-ориентированных программ. СУБД ADABAS в этом плане выгодно отличается от других СУБД наличием взаимодействия с развитой диалоговой системой NATURAL.

Система NATURAL представляет собой комплекс программных средств, включающий в себя все необходимые компоненты организации диалогового режима:

- интерпретатор диалогового языка высокого уровня для доступа к данным СУБД ADABAS и их обработки;
- средства отображения информации и ввода данных в режиме диалога;
- развитый диалоговый редактор текстов;
- программное обеспечение для хранения программ пользователя и их выборки при исполнении и др.

Использование системы NATURAL позволяет вести все работы по эксплуатации базы данных, включая и ее формирование, в диалоговом режиме, что не исключает, конечно, возможности ввода данных в пакетном режиме при большом их объеме.

Система NATURAL, как и СУБД ADABAS, спроектирована для работы в среде операционной системы ШМ-370 и программно-совместимых отечественных операционных систем (ОС ЕС ЭВМ современных версий). Операционные системы этого класса являются, по существу, ориентированными на пакетную обработку, и организация в их среде диалогового режима представляет собой самостоятельную сложную проблему. Эта проблема решается с помощью разработки дополнительного программного обеспечения — управляющих программ среды телеобработки данных, программ-телемониторов.

В целях совместимости с любым телемонитором система NATURAL построена таким образом, что для ее эксплуатации требуется специальная программа-драйвер, которая обеспечивает интерфейс между телемонитором и программами NATURAL. Фирмой-разработчиком NATURAL поставляются драйверы для систем CICS (аналогом которой является система КАМА) и TSO, а также драйвер пакетного режима. Драйверы NATURAL для других телемониторов должны разрабатываться их пользователями самостоятельно согласно имеющимся описаниям. При этом большая часть функций NATURAL, именуемая далее специальными функциями, остается для пользователя такого телемонитора недоступной.

Еще одной специфической чертой программного обеспечения системы NATURAL является наличие двух режимов терминального обслуживания: телетайпного режима и режима полного экрана. Существенно более полный по набору функций режим полного экрана ориентирован на дисплеи IBM-3270 или программно-совместимые дисплеи локального комплекса ЕС-7920, что предполагает использование конкретных кодов управления форматом экрана, наличие функциональной клавиатуры и т. п.

Таким образом, задача применения средств телеобработки при формировании базы данных на основе СУБД ADABAS и системы NATURAL сводится к следующим основным моментам:

- выбор телемонитора среды ОС ЕС;
- программирование драйвера NATURAL, поддерживающего все функции системы, включая и специальные;
- расширение спектра терминалов, для которых допустимо использование режима полного экрана NATURAL.

Ниже описывается решение этой задачи на основе системы ОББ.

Использование системы ОББ в качестве программы управления NATURAL существенно

упрощается тем, что система ОБЬ является «чистым* телемонитором, не перегруженным функциями, не свойственными системам этого класса, такими, например, как диалоговая отладка программ или непосредственное управление конкретными СУБД. Программирование драйвера NATURAL для системы ОБЬ представляет собой достаточно простую, хотя и трудоемкую задачу.

С этой целью в состав системы ОБЬ были включены программные средства, обеспечивающие имитацию локального дисплея EC-7920 при использовании других типов терминальной техники. Такой подход требует сложного и большого по объему программирования, однако он позволил не только автоматически поддерживать режим полного экрана NATURAL на всех обслуживаемых дисплеях, но и дал возможность использовать эти терминалы при работе с другими системами, ориентированными на дисплеи EC-7920.

Ниже подробно описана реализация специальных функций NATURAL:

- получение твердых (печатных) копий и дополнительных отчетов;
- управление рабочими файлами, т. е. наборами данных, не обслуживаемыми СУБД ADABAS;
- управление страничным набором данных, позволяющим сохранять копии экранов для повторного просмотра;
- управление буферным пулом программ, позволяющим хранить NATURAL-программы в оперативной памяти для ускорения работы системы;
- ограничение времени исполнения NATURAL-программ для исключения заикливания;
- обработка асинхронных сигналов прерывания с терминала пользователя.

9.2. Реализация специальных функций NATURAL

9.2.1. Твердые копии и дополнительные отчеты. Терминальный драйвер NATURAL системы ОБЬ обеспечивает выдачу дополнительных отчетов NATURAL-программ, т. е. корректное исполнение операторов

DISPLAY(NN)... и WRITE(NN)...

а также получение твердых копий (команда %H, функции H в команде LIST и др.). Для обеспечения этой функции необходимо выполнить следующее:

включить в задание на запуск NATURAL оператор DD вида

```
// CMPRINT DD SYSOUT = КЛ [, SPACE = ...]
```

задать в модуле параметров RCSNATPA и/ или в поле PARM при вызове NATURAL ключевое слово

```
PRINTER = NN или PRINTER = (СПИСОК)
```

Здесь КЛ — класс выходной очереди ОС ЕС для наборов данных главного принтера (файлов твердых копий);

NN (1-31) — означает, что дополнительные отчеты (принтеры), имеющие номера с 1 по NN, разрешены в программах на языке NATURAL и будут помещаться в наборы данных SYSOUT с классом КЛ (тем же, что у главного принтера);

(СПИСОК) — список классов SYSOUT дополнительных отчетов. Каждый элемент списка позиционно связан с соответствующим принтером. Например:

```
PRINTER= (,A,B, ,C)
```

означает, что разрешены принтеры с номерами 2, 3 и 5, и классы SYSOUT наборов данных соответственно равны A, B и C. Помимо классов в списке могут присутствовать значения:

DUMMY — соответствующий принтер имеет класс КЛ (как у главного принтера);

OFF — соответствующий принтер запрещен.

Если ключевое слово PRINTER задано и в модуле RCSNATPA, и в поле PARM, то последнее дополняет первое с перекрытием. Принтеры с номерами, не описанными ни в RCSNATPA, ни в PARM, считаются запрещенными.

Наборы данных SYSOUT передаются в выходную очередь ОС ЕС в момент, когда не

исполняется никакая NATURAL-программа (что соответствует подсказке NEXT на терминале) или по завершению сеанса пользователя в NATURAL. Если необходимо передать файлы принтеров в выходную очередь без выхода из NATURAL-программы, то в ней следует в требуемый момент вызвать программу RCSNATCP. Единственный параметр этой программы — полное слово. Биты (равные 1) позиционно соответствуют номерам принтеров, наборы данных которых следует поместить в выходную очередь. Старший бит соответствует главному принтеру. Например:

```
RESET PARAM(B4)
PARAM = H'C0000000'
CALL 'RCSNATCP' PARAM
```

При исполнении оператора CALL файл главного принтера и файл первого дополнительного отчета будут переданы в выходную очередь.

Твердые копии экрана могут быть получены с помощью команды %H. Однако экран может быть отформатирован таким образом, что не будет содержать ни одного поля, в которое разрешен ввод команды. Твердые копии таких экранов (как и любых других) могут быть получены при нажатии клавиши «ВЫЗОВ ТЕСТА».

В исходной библиотеке системы ОБЬ в разделе NATPRNTR хранится задание на вызов NATURAL в пакетном режиме и помещение в библиотеку SYSTEM программы PRINTER и соответствующей разметки полного экрана (MPRINTER). После исполнения этого задания пользователю становится доступной команда PRINTER, которая позволяет получать на экране карту распределения принтеров для данного пользователя по классам и изменять назначенные классы SYSOUT и, при необходимости, размер набора данных принтеров.

Изменение производится вводом требуемых значений вместо отображенных на экране. При изменении класса SYSOUT главного принтера автоматически меняются классы всех DUMMY-принтеров (т. е. дополнительных отчетов, имеющих класс главного принтера).

Программа PRINTER завершает работу, если при нажатой клавише «ВВОД» ни одно поле на экране не модифицировано.

9.2.2. Рабочие файлы. Терминальный драйвер системы ОБЬ обеспечивает управление рабочими файлами NATURAL-программы, т.е. корректное исполнение операторов

```
READ WORK NN... и WRITE WORK NN...
```

Для обеспечения этой функции необходимо выполнить следующее: включить в задание на запуск системы NATURAL достаточное число операторов DD вида:

```
//CMWRF DD DUMMY
```

для динамического (во время сеанса) назначения рабочих файлов;

здать в модуле параметров RCSNATPA и/или в PARM при вызове NATURAL ключевое слово

```
WORK = NN или WORK = (СПИСОК)
```

Здесь NN (1-31) означает, что использование рабочих файлов, имеющих номера с 1 до NN, разрешено в программе на языке NATURAL, причем назначение файлам наборов данных ОС ЕС будет производиться динамически;

(СПИСОК) — список имен операторов DD статически распределенных рабочих файлов. Каждый элемент списка позиционно связан с соответствующим рабочим файлом. Например:

```
WORK= (,WA, WB, WC)
```

означает, что разрешены рабочие файлы с номерами 2, 3 и 5, и использование этих рабочих файлов в NATURAL-программе обеспечивает доступ к наборам данных, описанным в операторах DD с именами WA, WB и WC соответственно. Требуемые операторы DD должны быть включены в задание на запуск системы NATURAL, в противном случае рабочие файлы с указанными номерами будут запрещены. Помимо имен операторов DD в списке могут присутствовать значения:

DUMMY — соответствующий рабочий файл будет распределяться динамически;

OFF — соответствующий рабочий файл запрещен.

Если ключевое слово WORK задано и в модуле параметров RCSNATPA, и в поле FARM, то последнее дополняет первое с перекрытием. Рабочие файлы с номерами, не описанными ни в RCSNATPA, ни в PARM, считаются запрещенными.

Рабочие файлы обслуживаются системой NATURAL с использованием метода доступа QSAM, поэтому соответствующие наборы данных должны иметь последовательную организацию. Допускается обработка разделов библиотечных наборов данных. Распределение рабочих файлов производится для каждого пользователя независимо от других.

В исходной библиотеке системы ОБЬ в разделе NATWORK хранится задание на вызов пакетного NATURAL и помещение в библиотеку SYSTEM программы WORK и соответствующих разметок полного экрана (MWORK и MWORKALC). После исполнения этого задания пользователю становится доступной команда WORK, которая позволяет получать на экране карту распределения рабочих файлов, динамически назначать и освобождать доступные рабочие файлы.

Код требуемой функции (I — подробная информация о рабочем файле, A — назначение, F — освобождение рабочего файла) вводится в поле рядом с номером соответствующего файла. Допускается ввод одновременно нескольких запросов, в этом случае они обрабатываются последовательно.

При вводе функции A пользователю предлагается разметка экрана для ввода требуемой для распределения информации (имя набора данных, имя тома, диспозиция и др.) При этом поддерживается система умолчаний, принятая в языке управления заданиями ОС ЕС (например, не требуется вводить имя тома для каталогизированного набора данных).

Дисковые и ленточные тома, используемые при динамическом распределении рабочих файлов с помощью программы WORK, должны быть смонтированы командой MOUNT ОС ЕС или перечислены в разделе PRESRES библиотеки SYS1.PARMLIB. Если имя набора данных не задано, то программа WORK строит временный набор данных, размещая его на указанном пользователем дисковом томе или доступном томе со статусом STORAGE или PUBLIC. Использование в качестве имени набора данных значения NULLFILE позволяет распределить рабочий файл на набор данных DUMMY.

Если функция A запрошена для уже распределенного рабочего файла, то он автоматически освобождается, а поля разметки экрана для назначения файла будут заполнены информацией о предыдущем назначении.

Распределение каждого рабочего файла с помощью программы WORK требует наличия одного оператора

```
//CMWKF DD DUMMY
```

в задании на запуск системы NATURAL. Количество свободных операторов DD CMWKF отображается программой WORK.

При завершении сеанса пользователя сформированное им распределение рабочих файлов сохраняется и может быть использовано в следующих сеансах, не только в пределах одного запуска системы NATURAL.

Программа WORK завершает работу, если при нажатой клавише «ВВОД» ни одно поле на экране не модифицировано.

9.2.3. Страничный набор данных. Терминальный драйвер системы ОБЬ обеспечивает управление страничным набором данных, т. е. корректное исполнение команд %P, %O, %S, %I, %E и подкоманд команды %E для просмотра сформированного страничного набора. Для обеспечения этой функции необходимо выполнять следующее:

включить в задание на запуск системы NATURAL оператор DD вида:

```
//CMPAGE DD DISP = OLD, UNIT = SYSDA,  
//VOL = SER = XXXXXX
```

здать в модуле параметров RCSNATPA и/или в PARM при вызове NATURAL и/или в команде GLOBALS ключевое слово

```
PD = NN
```

Здесь XXXXXX — имя дискового тома, на котором будут размещаться страничные наборы данных; NN — число страниц в наборе. По умолчанию принимается значение NN, равное 50.

Страничный набор данных создается и форматируется при первом запросе на запись страницы, т. е. по команде %P, %S или %I и уничтожается при завершении сеанса пользователя. Уничтожение и пересоздание страничного набора происходит также при изменении значения параметра PD с помощью команды GLOBALS. Имена страничных наборов имеют принятый в ОС ЕС формат имени временного набора данных, но являются фиксированными для каждого пользователя, поэтому сохранившиеся при завершении работы системы NATURAL страничные наборы будут автоматически удалены при следующем запуске.

9.2.4.Буферный пул программ. Терминальный драйвер системы ОБЬ обеспечивает управление буферным пулом программ NATURAL, т. е. автоматическое занесение, хранение, выборку и замещение программ в буферном пуле и корректное функционирование программ системной библиотеки SYSTAT.

Для обеспечения этой функции необходимо в задании на запуск системы NATURAL включить оператор DD вида:

```
//CMBPOOL DD DUMMY, SPACE = (1024,(PS,,ND))
```

где PS — размер буферного пула в килобайтах; ND — число элементов оглавления пула, т. е. количество программ, которые могут в нем храниться.

При задании некорректных (например, нулевых) значений PS и/или ND буферный пул не поддерживается.

9.2.5.Ограничение времени исполнения программы. Терминальный драйвер системы ОБЬ обеспечивает защиту от возможного заикливания NATURAL-программы с помощью ограничения времени ее исполнения. Для обеспечения этой функции необходимо в модуле параметров RCSNATPA и/или PARM при вызове NATURAL и/или в команде GLOBALS задать ключевое слово:

```
MT = NN
```

где NN — число секунд процессорного времени исполнения NATURAL-программы. По умолчанию принимается значение NN, равное 60.

9.2.6.Обработка асинхронных сигналов внимания. Терминальный драйвер системы ОБЬ обеспечивает обработку асинхронных сигналов внимания с терминалов, т. е. прекращение выполнения NATURAL-программ при нажатии пользователем терминала клавиши прерывания. Для обеспечения этой функции необходимо в модуле параметров RCSNATPA и/или PARM при вызове NATURAL задать ключевое слово;

```
ATTN = N
```

где N может принимать значения:

0— обработка асинхронных сигналов внимания не требуется;

1— требуется обработка асинхронного внимания.

По умолчанию принимается значение ATTN равное 0.

ГЛАВА 10

ИНСТРУМЕНТАЛЬНЫЕ ДИАЛОГОВЫЕ ПРОГРАММНЫЕ СРЕДСТВА

10.1. Редактирование текстов и отладка программ

Программа КДОМ, являющаяся модифицированным вариантом известного компонента дисплейной отладки (КДО), служит для организации рабочего места программиста и призвана обеспечивать повышение эффективности труда программиста в процессе подготовки и отладки программ за счет замены трудоемкой работы на перфокартах на более оперативную и наглядную работу на основе алфавитно-цифровых дисплеев; использования более доступных и простых, чем аналогичные утилиты операционной системы, средств ведения библиотек; наличия команд-подсказок; использования принципа выбора команд из предлагаемого «меню»;

большого количества диагностических сообщений, позволяющих быстро освоить работу с комплексом.

КДОМ функционирует под управлением системы ОБЬ на любой модели ЕС ЭВМ, кроме ЕС-1010. Максимальное количество терминалов, одновременно обслуживаемых программой КДОМ, равно 16. Для вывода на печать, перфокарты или магнитные ленты требуются соответствующие устройства.

В целях наглядности простоты ввода и вывода информации экран дисплея разбивается при работе программы КДО на три части. Первая (верхняя) служит для вывода диагностических сообщений, вторая — для ввода команд программы КДО, остальные — для ввода команд, доступных в данный момент, с указанием формата команд и комментарием к ним, а также для вывода данных, нужных пользователю: страниц томов прямого доступа, оглавления и разделов библиотек. Эти строки служат также для ввода данных при создании, исправлении или обновлении разделов.

Входной информацией являются команды КДОМ, библиотечные и последовательные наборы данных; выходной — библиотечные и последовательные наборы данных, диагностические сообщения.

Пользователь, работающий с этой программой, имеет возможность просматривать на экране алфавитно-цифрового дисплея оглавление тома прямого доступа, исключать, переименовывать наборы данных с любой организацией. При этом ему доступны как характеристики тома (количество свободной памяти на томе и в оглавлении тома, количество дефектных дорожек и др.), так и характеристики набора данных (организация, формат и размер блоков, количество и распределение экстенгов, количество свободного места, признаки защищенности набора данных и др.)

При работе с оглавлением произвольных библиотек пользователь может его просматривать, исключать, переименовывать разделы, назначать разделу альтернативное имя, просматривать данные пользователя, включенные в элемент оглавления. При работе с разделами символических библиотек пользователь может просматривать раздел на экране, листая его вперед, назад, постранично и построчно, а также искать нужную страницу путем задания метки, номера и контекста. Избранные строки раздела можно корректировать, замещать, удалять, вставлять и перенумеровывать. При создании нового раздела в него можно включать данные, вводимые с экрана или копируемые из раздела других библиотек. Разделы можно выводить на печать или перфокарты. Задание, представленное разделом, может быть запущено для выполнения. Выходной листинг задания может быть выведен на экран дисплея.

Реализация возможностей программы КДОМ позволяет использовать ее для других целей, например, для организации информационно-справочной службы, ведения и размножения документации и т. д. Управление работой программы КДОМ производится с помощью вводимых с терминала команд. Команды программы КДОМ связаны между собой по иерархическому принципу, при котором в каждый момент времени для выполнения доступны только команды выбранного уровня иерархии. Команды КДОМ делятся на следующие группы:

- общего назначения, доступные на любом уровне работы;
- работы с оглавлением тома прямого доступа;
- чтения выходного листинга задания;
- работы с оглавлением библиотеки;
- работы с разделами библиотеки.

При вводе не обязательно набирать команду полностью. Достаточно набрать столько символов, сколько необходимо для однозначного распознавания команды. Ввод «пустой» команды равносителен повторению предыдущей.

В *первую группу* входят команды «?», T, WTO, DUMP, LOG, DD, «*», «.», GET, FREE, REP, UP, DOWN. Команда T используется для завершения работы на данном уровне и перехода на более высокий уровень. WTO применяется для передачи сообщений оператору ЭВМ. В результате выполнения передается текст из поля операнда этой команды. DUMP дает возможность просматривать и изменять содержимое ОП. В поле операндов задается адрес памяти в шестнадцатеричном виде. Формат выдачи аналогичен формату дампа, принятому в ОС ЕС. С помощью команды «.» задаются с терминала команды оператора ЭВМ. Текст

команды оператора задается в поле операндов. В ответ на вводимые команды, операционная система выдает сообщения не на терминал программы КДО, а оператору ЭВМ. LOG позволяет вести просмотр строк системного журнала. С помощью команды DD просматривается список DD-утверждений. В результате выполнения этой команды на терминал выводятся имена DD-утверждений, имена томов и наборы данных, описанных в этих утверждениях. Просмотр списка начинается с (N + 1)-го DD-утверждения (по умолчанию N = 0). Команда «*» служит для вызова на экран терминала сообщений операционной системы, помеченных символом «*», т. е. сообщений, по которым требуется действие оператора ЭВМ. GET формирует DD-утверждение для задания библиотечных или последовательных наборов данных. FREE служит для динамического освобождения набора данных. Команда REP повторяет выполнение команды через определенный интервал времени. В поле операнда указывается время в секундах. Если операнд опущен, то предыдущая команда повторяется всего один раз. UP (DOWN) перемещает содержимое экрана дисплея на несколько строк вверх или вниз, начиная с указанной строки. В поле операндов задается количество строк, на которое должен произойти сдвиг. При этом курсор необходимо подвести в строку, начиная с которой требуется сдвиг, а после операнда N набрать, по крайней мере, один пробел.

Ко второй группе относятся следующие команды. VOL определяет имя тома, с оглавлением которого будет производиться работа. GEN позволяет получить общие характеристики тома. В результате выполнения команды на терминал выдаются размер оглавления тома, количество свободных элементов оглавления, дефектных дорожек на томе и свободной памяти, емкости тома в цилиндрах, цилиндра в дорожках и дорожек в байтах. R служит для постраничного просмотра оглавления тома. Первая строка страницы резервируется под заголовок, а остальные распределены по одной на каждый набор данных. Для каждого набора данных выдается имя и серийный номер тома, на котором он был первоначально создан, дата создания набора данных, способ организации набора данных, количество экстенгов в наборе данных, формат блоков, размеры блока и записи. RS позволяет вернуться к начальной странице оглавления. FND используется для получения характеристики набора данных, имя которого указывается в поле операндов. В результате выполнения команды на экран выдаются: единица распределения памяти для набора данных, количество свободной памяти в наборе данных, физический адрес элемента оглавления тома для набора данных, признаки защиты набора данных по записи и чтению. По команде REN переименовывается набор данных. В поле операндов задаются старое и новое имена, отделенные запятой. По DEL набор данных с именем, указанным в поле операндов, уничтожается.

К третьей группе относятся следующие команды. Команда JOB используется для чтения выходного листинга задания. В поле операнда указываются имя задания и через запятую класс выходной очереди, где помещено задание. Команда R вызывает для чтения набор данных, включенный в выходную очередь SYSOUT. Эта команда дает доступ к командам следующего уровня, которые позволяют читать следующую страницу листинга; вернуться к начальной странице листинга; сдвинуть вводимую страницу на строку вверх; вывести на экран страницу листинга, начиная со строки с номером, заданным в операнде команды; вывести страницу, содержащую в первой строке контекст, заданный в операнде команды; удалить набор данных, включенный в выходную очередь SYSOUT. Команда RS применяется для чтения листинга системных сообщений и дает доступ к командам глубокого уровня. После чтения выходного листинга задания для перехода на верхний уровень применяется команда T. В поле операндов помещается класс системного вывода, в который надо поместить задание после чтения. При отсутствии операнда задание помещается в класс A.

К четвертой группе относятся следующие команды.

Команда верхнего уровня DIR определяет библиотеку, с оглавлением которой будет произведена работа, и дает доступ к командам подчиненного уровня. В поле операндов задается имя DD-утверждения, описывающего требуемую библиотеку. R позволяет постранично просматривать оглавление библиотеки. RS дает возможность вернуться к начальной странице оглавления. В поле операндов задается имя раздела. По REN переименовывается раздел. В поле операндов задаются старое и новое имена раздела. По DEL производится исключение раздела из оглавления библиотеки.

К пятой группе относятся следующие команды.

Команда LIB вызывает библиотеку, с разделами которой будет производиться работа. В поле операндов задается имя DD-утверждения требуемой библиотеки.

Команда R вызывает раздел библиотеки для просмотра и дает доступ к командам более низкого уровня. Ряд команд (RS, RL, RC, RCN) аналогичны командам третьей группы. По команде R на экран выводится очередная страница раздела. RB выводит на экран предыдущую страницу раздела. RBL сдвигает вводимую страницу на строку вниз. RN выводит на экран дисплея страницу раздела, начиная со строки с номером, заданным в операнде команды. RM выводит страницу раздела, начиная со строки с меткой в первой позиции, совпадающей с меткой, заданной в поле операнда.

Команда N вызывает раздел для перенумерации строк, внесения текста во все строки, для замены имен и дает доступ к командам более глубокого уровня (команды N, B, C, X), которые производят перенумерацию строк, занесение в заданные позиции всех строк раздела, замену имен во всех строках раздела, замену одного контекста на другой во всех строках раздела. Команда A выбирает раздел для исправления (при этом возможно только замещение строк) и открывает доступ к командам более глубокого уровня: W, R, RS, RB, RL, RBL, RN, RC, RCN. По команде W происходит запись исправленной страницы в раздел библиотеки. U вызывает раздел для обновления (строки раздела должны быть пронумерованы). D служит для исключения строк с номерами от N1 до N2. Команда W служит для записи необходимых строк. В операнде указывается количество записываемых с экрана строк.

Разметка экрана по формату бланка программирования на языке Ассемблер производится по команде S. Команда L применяется для создания нового раздела или замещения существующего разделом с таким же именем. Новый раздел записывается на свободное место в конце библиотеки. В результате выполнения команды L открывается доступ к командам просмотра указанного раздела. По команде K указанный раздел становится доступным для копирования без вызова копируемых строк на экран. При помощи C производится копирование строк по относительным номерам, а при помощи M производится копирование строк по номерам, заданным в 73 — 80-й позициях выбранного раздела. В операндах команд C и M указывается начало и конец диапазона номеров копируемых строк, Команда P выбирает раздел для распечатки или перфорации и дает доступ к командам PRINT и PUNCH. Команда PRINT производит распечатку раздела, команда PUNCH — перфорацию раздела.

Одним из актуальных направлений совершенствования возможностей компонента дисплейной отладки программ явилось расширение спектра обслуживаемых терминальных устройств. Актуальность этой задачи объясняется тем, что большинство других программных комплексов данного класса, как правило, обслуживают весьма ограниченный набор типов дисплейных устройств. После проведенных работ по усовершенствованию, программный комплекс КДОМ может обслуживать все типы диалогового периферийного оборудования, поддерживаемого системой ОБЬ, в том числе и печатающие (недисплейные) терминалы. Это дисплейный комплекс ЕС-7920 в локальном и удаленном варианте; стартстопные дисплейные станции АП-62, ИЗОТ-7925, АП-64; синхронные терминалы МЕРА и их аналоги в комплексе ЕС ТЕЛ 4.2 ЕС-S566; абонентские пункты АП-1, АП-1М, ТАП-2А во всех вариантах их подключения к аппаратуре передачи данных; консоль ОС ЕС при ее использовании в качестве терминала системы ОБЬ. Приведенный широкий спектр терминалов позволяет рекомендовать комплекс КДОМ для организации автоматизированного рабочего места программиста, организуемого под управлением системы ОБЬ.

Одной из проблем, решение которой необходимо для программ редактирования текстов, является реализация возможности документирования (распечатки) текстов через программу системного вывода. Чрезмерная жесткость операционной системы ОС ЕС, при распределении ресурсов проблемным программам, вызывает необходимость их закрепления за системой для получения распечатки через программу системного вывода. Проведенные доработки КДОМ позволили включить в состав данного комплекса программные средства, обеспечивающие по командам пользователей терминалов создание временного набора данных на одном из доступных томов прямого доступа; копирование информации из раздела обслуживаемой библиотеки в созданный набор; постановку созданного набора данных в системную выходную

очередь требуемого класса. Это значительно повысило эксплуатационные характеристики дисплейного отладочного комплекса.

Важной функцией программного обеспечения, используемого при организации рабочего места программиста, является предоставление пользователю возможности просмотра результатов исполнения задания на терминале. Без такой возможности, для обеспечения просмотра результатов задания требуются значительные организационные усилия (такие, как внесение печатающего устройства в дисплейный зал, организация дисциплины выходных классов и др.) и непроизводительный расход бумажных носителей. Модифицированный компонент дисплейной отладки КДОМ позволяет просматривать результаты исполненных заданий на пользовательских терминалах всех обслуживаемых типов. После просмотра листинг задания может быть переведен в любой класс системной выходной очереди, что дает возможность распечатать задание, сохранить его для последующих просмотров или уничтожить. Другой важной функцией по удаленному вводу заданий является организация работы программы системного ввода в режиме ожидания. Подобная функция необходима для уменьшения требуемого объема оперативной памяти и времени обработки задания. Стандартное решение этой проблемы — запуск отдельной программы системного ввода для каждого задания — не является, конечно, удовлетворительным. В процессе доработки КДОМ была полностью переработана и значительно усовершенствована программа управления потоком заданий как с терминалов пользователей, так и с консоли оператора ОС ЕС

При работе пользователя в режиме разделения времени ему часто требуется информация, выдаваемая системой на консоль оператора ОС ЕС. Это может быть информация о прохождении заданий и распределении ресурсов; о входных и выходных очередях системы; о наличии заданий в активном состоянии. Проведенные доработки рассматриваемой системы отладки позволили включить в командный перечень комплекса команды отображения указанной информации. При этом запросы на эту информацию, посылаемые с терминалов пользователей, не влияют на ее выдачу на консоль оператора ОС ЕС.

Помимо перечисленных, программы комплекса КДОМ получили свое развитие путем реализации возможностей создания наборов данных в процессе работы за терминалом КДОМ; чтения разделов библиотек исходных модулей, содержащих укороченные блоки; просмотра заданных операторов описания данных ОС ЕС с отметкой возможности их динамического распределения и назначения методам доступа системы ОБЬ.

Проведение усовершенствование комплекса дисплейной отладки осуществлено без увеличения требуемого для исполнения объема оперативной памяти и составляет 40 Кбайт.

Тщательный анализ возможностей различных программных комплексов организации рабочего места программиста позволил отобрать для адаптации под управление системы ОБЬ две перспективные программные системы: ДЖЕССИ — для организации рабочего места программиста-разработчика проблемных программ; и АРГУС — для организации рабочего места системного программиста. Выбор именно этих программных комплексов обусловлен их широкими возможностями, выделяющими их среди систем данного класса, и достаточной простотой адаптации для включения под управление системы ОБЬ.

Поскольку все современные средства дисплейной отладки представляют собой мультитерминальные системы, то их включение под управление мультитерминальной системы ОБЬ осложнено, конечно, возможными различиями в принципах обслуживания сети терминалов. Безотносительно к этим принципам, системы АРГУС и ДЖЕССИ имеют совпадающие друг с другом и с соответствующим механизмом системы ОБЬ способы динамической реконфигурации обслуживаемых терминалов. Это позволило наладить управление указанными подсистемами без всяких переработок. По команде пользователя терминал передается требуемой подсистеме и вновь захватывается системой ОБЬ по команде выхода из режима обслуживания подсистемой.

Программный комплекс ДЖЕССИ представляет собой развитое средство организации рабочего места программиста. Его эксплуатация показала хорошие качества ДЖЕССИ как программы редактирования текстов и отладки программ. В настоящее время в дополнение к обычным функциям организации дисплейной отладки, комплекс программ ДЖЕССИ позволяет:

—определять функциональную клавиатуру дисплея ЕС-7920 для каждого пользователя отдельно, закладывая в исполнение специальной функции произвольные команды;

—определять любое количество макросимволов и присваивать им произвольные значения. Полученные таким образом макросимволы могут затем использоваться в любых командах системы;

—определять пакеты команд, содержащие любое количество произвольных команд системы. При этом пакет команд исполняется системой как одна команда и может содержать программные структуры обобщенного цикла и альтернативы;

—определять шаблоны новых команд на базе стандартных команд системы, что позволяет перейти к другому командному языку.

В дополнение к вышеизложенному, заметим, что средства редактирования текстов в системе ДЖЕССИ отличаются простотой, наглядностью и удобством в работе. Единственным недостатком рассматриваемой системы является, в сущности, ограниченный спектр обслуживаемой терминальной техники (локальная дисплейная станция ЕС-7906 и локальный комплекс ЕС-7920).

Программный комплекс АРГУС содержит большое число функций и может быть, в принципе, использован для организации рабочих мест программистов любого профиля. Однако сложность некоторых функций для проблемного программиста и слабая организация системы разделения доступа делает его не всегда пригодным для использования программистами-прикладниками. Все же, было признано целесообразным провести работы по включению системы АРГУС под управление системы ОБЬ для организации на ее базе рабочего места системного программиста. По ряду вопросов, относящихся к компетенции системного программиста — установка и снятие томов прямого доступа, обслуживание системного каталога и др.— система АРГУС содержит средства, выгодно отличающие ее от других систем данного класса.

10.2. Программа диалоговой проверки и корректировки данных, хранящихся на магнитных лентах

Основное назначение программного средства ПЛТ (проверка ленточных томов) осуществлять анализ информации, записанной на магнитные ленты, путем визуального контроля в интерактивном режиме. Универсальность программы ПЛТ заключается в ее полной независимости от структур данных, расположенных на ленточном томе. Программа ПЛТ позволяет читать любые записи на магнитной ленте, длина которых не превышает 32 Кбайта. Прочитанная информация пофрагментно выводится на экран дисплея. Размер фрагмента и его начало задаются пользователем. На экране информация отображается в одном из трех форматов:

—символьный формат (соответствует коду ДКОИ);

—шестнадцатиричный формат;

—командный формат (соответствует формату и кодам команд ассемблера ЕС ЭВМ).

Переход от одного формата к другому осуществляется при помощи команд ФС, ФХ, ФК. Перечисленные команды могут задаваться при помощи клавиш функциональной клавиатуры дисплея ЕС-7927.

Технология тестирования информации, хранящейся на магнитной ленте, полностью определяется пользователем. Работа осуществляется по принципу «запрос — ответ» при помощи команд управления. Если при выполнении очередной команды возникает ошибка ввода/вывода, то программа ПЛТ переходит к анализу причины ее возникновения. Для анализа сбойных ситуаций используются стандартные средства диагностики, предусмотренные в операционной системе. Диагностические сообщения об ошибках выводятся на экран дисплея. По требованию пользователя программа ПЛТ представляет возможность визуально отобразить данные, составляющие сбойный блок. В процессе работы выдается следующая дополнительная информация: текущий номер файла, длина последнего прочитанного блока, положение отображаемого фрагмента внутри блока.

Проверку магнитной ленты можно производить даже после достижения логического конца тома. При наличии сбойных блоков, нарушении логической структуры тома, при

возникновении острой необходимости корректировки информации на магнитной ленте, без которой дальнейшая работа с ленточным томом при использовании стандартных средств операционной системы является невозможной, программа ПЛТ предоставляет специальные средства восстановления. В случае их использования, ответственность за сохранность информации на магнитной ленте ложится на пользователя. Это связано с тем, что команды средств восстановления производят операцию записи на магнитную ленту. Основными условиями 122 нормального восстановления и исправления информации на магнитных лентах являются полная техническая совместимость лентопротяжных устройств, на которых создается и исправляется информация, использование при формировании данных стандартных методов записи ОС ЕС и установка при создании и исправлении информации одинаковой плотности записи.

Программу ПЛТ можно использовать в качестве дополнительного средства проверки работоспособности лентопротяжных устройств при проведении профилактических и ремонтных работ. Для этого следует заранее подготовить тестовую магнитную ленту. Эксплуатация программы ПЛТ не требует особой предварительной подготовки пользователей. В ПЛТ предусмотрена возможность подключения дополнительных модулей для обслуживания новых типов терминальных устройств.

10.3. Программа проверки и корректировки информации, хранящейся на дисковых томах

Для анализа информации, записанной на магнитные диски, разработана универсальная программа проверки дискового тома (ПДТ). Универсальность программы ПДТ заключается в ее полной независимости от структур данных, расположенных на дисковом томе.

Программа ПДТ функционирует как под управлением системы ОБЬ, так и автономно на любой модели ЕС ЭВМ, кроме ЕС-1010. Требуемая для работы оперативная память не превышает 40Кбайт. Отображение информации осуществляется на экран дисплея одного из следующих устройств: дисплейного комплекса ЕС-7920 (локального или удаленного варианта); абонентских пунктов ЕС-8562 и ЕС-8566; дисплейного комплекса ЕС-7970, работающего в режиме ЕС-7920; дисплейного комплекса ЕС-7910.

Входной информацией программы ПДТ являются ее команды и любые записи на диске, выходной — записи на магнитном диске и диагностические сообщения. Программа позволяет читать любые записи на диске, подготовленном для работы в ОС ЕС. Допускается чтение произвольной записи (включая записи RO) по ее дисковому адресу, отображение и обновление полей ключа и данных. Программа ПДТ обеспечивает сканирование тома прямого доступа, последовательный просмотр записей по возрастанию или убыванию дисковых адресов, поиск метки в оглавлении тома по имени набора данных, непосредственный выход на начало первого экстента набора данных по его имени, непосредственный выход на начало раздела библиотечного набора данных по имени раздела, перезапись данных и назначение альтернативных дорожек для дефектных. Просмотр записей на экране дисплея производится по трем форматам: символьному, шестнадцатеричному и командному. Можно вызывать на экран дисплея произвольные фрагменты записи или анализировать информацию кадрами. В целях наглядности и простоты ввода и вывода информации экран Дисплея разбивается программой ПДТ на три области:

—все поле экрана минус две строки служит для вывода справочной таблицы команд, а также для вывода информации, записанной на диск, размер области определяется в соответствии с типом дисплея;

—предпоследняя строка экрана или верхняя строка для ЕС-8562, ЕС-8564 служит для вывода диагностических сообщений;

—последняя строка экрана или вторая строка для ЕС-8562, ЕС-8564 предназначена для ввода команд.

Анализируемый дисковый том должен находиться в состоянии ONLINE и содержать хотя бы один блок DSCB нулевого формата в оглавлении тома. Все необычные ситуации ввода-вывода на диске анализируются с помощью макрокоманды SYNADAF ОС ЕС, которая формирует текстовое диагностическое сообщение. Для назначения альтернативных дорожек применяется

макрокоманда ATLAS ОС ЕС. Управление работой программой ПДТ производится с помощью команд, вводимых с терминалов. Команды собраны в справочную таблицу, которая высвечивается на экране дисплея в начале работы и может быть вызвана на экран в любой момент времени с помощью команды С или клавиши ВЫЗОВ ТЕСТА (для терминалов, имеющих функциональную клавиатуру). Введенная команда запоминается в буфере команд, поэтому пустой ввод (нажатие клавиши ВВОД без операндов) воспринимается как повторение предыдущей команды. Для терминалов с функциональной клавиатурой ЕС-7920, ЕС-7910, ЕС-8566 ряд функциональных клавиш задействован под команды программы ПДТ, их названия указаны в скобках.

Команды программы ПДТ делятся на три группы.

1. Команды управления.

Вызов справочной таблицы. Формат команды:

С (ВЫЗОВ ТЕСТА)

вывод справочной таблицы во время работы не изменяет содержимого буфера дисковой записи.

Завершение работы. Формат команды:

К (СТРН ЭКР)

Переход на другой том. Формат команды:

D VOLSER

команда D позволяет перевести обслуживание на другой том, находящийся в состоянии ONLINE, без пропуска программы.

2. Дисковые команды.

Чтение следующей записи. Формат команды:

В (ПФ1)

Чтение предыдущей записи. Формат команды:

Н (ПФ2)

Чтение записи по адресу. Формат команды:

A CССNHRR (СССННRR ПФ3)

дисковые адреса записей задаются в шестнадцатеричном формате, адрес цилиндра накопителей, имеющих менее 256 цилиндров, можно задавать в форме СС.

Обновление текущей записи. Формат команды:

О (ПФ4)

не следует выдавать команду О после вывода справочной таблицы, так как в этом случае справочная таблица переписывается на диск. *Назначение альтернативного трека.* Формат команды:

T CССNH (СССНН ПФ5)

Чтение метки набора данных. Формат команды:

M XXX...XX (XXX...XX ПФ11)

в поле операнда указывается имя набора данных.

Чтение первой записи набора данных. Формат команды:

Ф XXX...X (XXX...X ПФ12)

Чтение первой записи раздела библиотечного набора данных. Формат команды:

Ф XXX...X (МММ...М) (XXX...X(МММ...М) ПФ12)

XXX...X — имя набора данных, МММ...М — имя раздела.

3. Команды анализа записей. *Вывод первого кадра записи.* Формат команды:

ЛС (ПФ6)

Вывод следующего кадра записи. Формат команды:

ЛВ (ПФ7)

Вывод предыдущего кадра записи. Формат команды:

ЛН (ПФ8)

Вывод указанных байтов. Формат команды:

ЛБ ХХХХХ, ХХХХХ (ХХХХХ, ХХХХХ ПФ9)

номера байтов, указанные в поле операнда, начинаются с единицы. *Обновление записи в памяти.* Формат команды:

ОБ (ПФ10)

по этой команде информация на диск не переписывается, но буфер в ОП обновляется.

Вывод на экран записи в командном формате. Формат команды:

ФК (ПД1)

Вывод на экран записи в шестнадцатиричном формате. Формат команды:

ФХ (ПД2)

Вывод на экран записи в символьном виде. Формат команды:

ФС (ПД3)

На экране дисплея постоянно поддерживается справочная информация об анализируемой записи: адрес на диске, размеры полей ключа и данных, номера отображаемых в настоящий момент байт. Например:

ГОТОВО * АДРЕС СССННRR, ДЛИНА КЛЮЧА = КKK,

ДЛИНА ДАННЫХ = DDDDD, БАЙТЫ = LLLLL - ННННН

Формат отображения информации на экране дисплея аналогичен формату программы IMASPZAP.

10.4. Система автоматизированной подготовки текстовой информации

Для совершенствования хранения, копирования и передачи пользователям технической документации требуется автоматизация процесса ее подготовки на магнитных носителях, обеспечивающая удобный способ редактирования и распечатки в соответствии с требованиями к оформлению документов по стандартам ЕСПД. Для этого целесообразна разработка специализированных программных средств, которые должны обеспечивать простоту и удобство подготовки и корректировки исходной информации, а также возможность работы с программным комплексом пользователя-непрограммиста и применения системных утилит ОС ЕС для копирования распечатки информации и ведения архивов.

Решение поставленной задачи основано на подготовке документации на МН в образах перфокарт и на введении в исходный текст специальных управляющих операторов, с помощью которых производятся различные процедуры редактирования и форматирования документации. Управляющие операторы набираются на отдельных перфокартах и располагаются непосредственно перед формируемой информацией.

Программный комплекс ФОРМАТОР может функционировать на любой модели ЕС ЭВМ, кроме ЕС-1010, под управлением ОС ЕС. Для работы программного комплекса требуется ОП не менее 10 Кбайт, внешняя память на диске не менее 1 Мбайт, Входная и выходная информация представлена в образе перфокарт. Для пользователя такая форма представления информации наиболее привычна, легко обозрима, дает возможность коррекции массивов с использованием средств дисплейной отладки, а также обеспечивает простоту обслуживания.

Система ФОРМАТОР состоит из базового программного комплекса и вспомогательных программ FRMNUMBR, FRMLIST2, FRMPRINT. Базовый программный комплекс системы ФОРМАТОР представляет собой программу динамической структуры, состоящую из управляющего модуля FRMCTL04, модулей форматирования текста FRMPAGE и оглавления FRMDIR.

Для исполнения программного комплекса ФОРМАТОР в отдельном шаге задания рекомендуется следующий набор управляющих операторов языка управления заданиями ОС ЕС:

```
// EXEC PGM = FRMCTL04, PARM = 'ТЕКСТ'  
// STEPLIB DD DSN = FRMLOAD, DISP = SHR  
// FRMIN DD описание основного входного массива  
// FRMFIG DD описание массива рисунков  
// FRMLIST DD описание выходного массива  
// FRMDIR DD UNIT = SYSDA, SPACE = (TRK, (5,5))
```

Параметр ТЕКСТ определяет собой колонтитул, помещаемый системой ФОРМАТОР на каждую страницу выходного документа, кроме первой. Поле PARM может быть опущено.

Наборы данных, описываемых операторами DD с именами FRMIN, FRMFIG, FRMLIST, представляют собой массивы в образах перфокарт на произвольных носителях. Допускается использование разделов библиотек (одной или разных) исходных модулей с произвольным коэффициентом блокирования. При выводе информации на АЦПУ и МЛ описание выходного массива должно содержать параметр DCB = BLKSIZE.

Программа FRMNUMBR обрабатывает текст, получаемый в результате работы системы ФОРМАТОР с целью переноса содержания на третью страницу (после титульного листа и аннотации).

Для исполнения программы FRMNUMBR рекомендуется следующий набор управляющих операторов:

```
// EXEC PGM = FRMNUMBR  
// STEPLIB DD DSN = FRMLOAD, DISP = SHR  
// FRMUT1 DD описание массива  
// FRMUT2 DD UNIT = SYSDA, SPACE = (TRK, (500, 100))  
// FRMDIR DD UNIT = SYSDA, SPACE = (TRK, (5,5))
```

Набор данных, описываемый оператором DD с именем FRMUT1, содержит текст, полученный после окончания работы системы ФОРМАТОР. Этот набор данных должен размещаться на томе прямого доступа (допускается использование раздела библиотеки). Выходной массив располагается на том же месте. Если на титульном листе документа содержится текст

ЛИСТОВ 0000

то в поле, заполненное нулями, помещается число страниц документа.

Программа FRMLIST2 формирует текст, содержащий по 128 символов в строке, таким образом, чтобы обеспечить печать двух смежных страниц отформатированного текста на одном листе АЦПУ. Для исполнения программы FRMLIST2 следует задать следующий набор управляющих операторов:

```
// EXEC PGM = FRMLIST2  
// STEPLIB DD DSN = FRMLOAD, DISP = SHR  
// FRMUT1 DD описание входного массива  
// FRMUT2 DD описание выходного массива
```

Входной массив содержит текст, полученный в результате работы системы ФОРМАТОР или программы FRMNUMBR. При выводе информации на АЦПУ и МЛ описание выходного массива должно содержать параметр DSN = BLKSIZE.

Программа FRMPRINT обеспечивает добавление управляющих символов для ускорения выдачи текста на АЦПУ. Набор управляющих операторов для исполнения программы FRMPRINT составляется следующим образом:

```
// EXEC PGM = FRMPRINT, PARM = 'ЧИСЛО КОПИЙ'  
// STEPLIB DD DSN = FRMLOAD, DISP = SHR  
// FRMUT1 DD описание входного массива  
// FRMUT2 DD описание выходного массива
```

Параметр ЧИСЛО КОПИЙ задает требуемое количество копий выходного документа (по умолчанию число равно 1). В операторе DD FRMUT2 должен быть указан параметр DSN = BLKSIZE, определяющий размер блока выходного набора данных, кратный длине записи выходного массива плюс единица.

Входная информация в системе ФОРМАТОР задается в виде текстов и рисунков. Каждый вид информации составляет отдельный массив исходных данных. Исходная текстовая информация представляет последовательность набиваемых на перфокарты предложений, которые записываются на стандартных бланках для программирования. Записи исходного текста в строке занимают позиции 1 — 72, позиции 73 — 80 игнорируются.

Если входная информация задана в виде рисунков, то записи, составляющие рисунок, занимают в каждой строке 1 — 60-ю позиции, остальные позиции строки игнорируются.

Выходной информацией системы ФОРМАТОР являются отформатированные тексты и рисунки, которые выводятся на магнитные носители в образах перфокарт. Указанная информация занимает 17 — 76-ю позиции в строке, причем после каждой перфокарты с информацией помещается по одной пустой перфокарте для вывода текста на печать через строку. На второй строке каждой страницы (кроме титульного листа) печатается номер страницы, в четвертой — колонтитул. Текст размещается на странице, начиная с седьмой строки, в конце каждой страницы остается пять чистых строк. В текст титульного листа может быть введено количество страниц, содержащихся в документе.

Содержание документа формируется автоматически системой ФОРМАТОР и помещается на последних страницах. При необходимости содержание может быть перенесено на третью страницу документа.

Управляющие операторы набираются на отдельных перфокартах и располагаются непосредственно перед информацией, которую необходимо отформатировать. Последовательность расположения операторов определяется требуемой структурой текста или рисунка. Признаком управляющего оператора является наличие символа в первой позиции. Формат управляющего оператора:

#КОД [X,Y,Z]

Код операции занимает 2 — 4-ю позиции, 5-я позиция — пробел. Операнды X, Y, Z необязательны и могут быть опущены (в этом случае принимаются умалчиваемые значения). Если символ # встречается не в 1-й позиции, то это означает запрет переноса данного слова.

Если символ # обнаружен в 1-й позиции, а код операции не распознается, то обрабатывается оператор # IND с умалчиваемыми значениями параметров, и последующие строки включаются в форматирование.

Существуют следующие управляющие операторы системы ФОРМАТОР.

1. Оператор выбора режима. Формат оператора:

#OPT [N]

В режиме копирования текста (N = 0) данные переносятся из входного массива в выходной без изменений. В режиме форматирования (N = 1) из входного массива выбираются разделенные пробелами слова, из некоторых набираются строки выходного массива; по умолчанию N=1.

2. Оператор начала абзаца. Формат оператора:

#IND [K,L,M]

где K — номер позиции начала абзаца; L — число строк, которые надо пропустить перед абзацем; M — остаток строк; по умолчанию K = 6, L = 0, M = 2. Если абзац расположен в начале страницы, строки перед абзацем не пропускаются, а если на странице осталось меньше строк, абзац переносится на другую страницу.

3. Оператор сужения столбца. Формат оператора:

#FLD[K,L]

где K — начало поля текста; L — начало первого пустого поля; по умолчанию K= 1, L = 255. Форматируемый текст размещается между указанными позициями до появления следующего оператора FLD.

4. Оператор пропуска строк. Формат оператора:

#SKP [N] где N — число строк; по умолчанию N = 1.

5. Оператор начала страницы. Формат оператора:

#PAG [N]

где N — число чистых страниц; по умолчанию N = 0. Осуществляется переход на новую страницу, и выводится N чистых страниц, В начале страницы перехода на новую страницу не произойдет.

6. Оператор проверки остатка. Формат оператора'

#RST [N]

где N — число пустых строк; по умолчанию N = 6. Если на странице осталось меньше N строк, то произойдет переход на новую страницу.

7. Оператор вывода на середину страницы. Формат оператора:

#MDL[N]

где N — число строк над серединой страницы; по умолчанию N = 0. Форматирование текста будет продолжено со строки, расположенной на N строк выше середины страницы.

8. Оператор вывода в конец страницы. Форма оператора:

#LST[N]

где N — число строк над концом страницы; по умолчанию N = 0. Форматирование текста будет продолжено со строки, расположенной на N строк выше конца страницы.

9. Оператор центровки текста. Формат оператора:

#CNT[K,L,M]

где K — число центрируемых строк; L — число строк, которые надо пропустить перед отцентрированным текстом; M — остаток строк; по умолчанию K=1, L = 0, M = 0. Из выходного массива выбирается K строк, и каждая размещается симметрично относительно границ страницы. Если текст располагается в начале страницы, то чистые строки перед ним не пропускаются, а если до конца страницы осталось меньше M строк, то происходит переход на новую страницу.

10. Оператор копирования рисунков. Формат оператора:

#FIG[N]

где N — число страниц рисунков; по умолчанию N=1. По завершении форматирования текущей страницы из массива рисунков в выходной массив будет скопировано N страниц.

11. Оператор копирования заголовка. Формат оператора:

#HDR[K,L,M]

где K — число строк в заголовке; L — число строк, пропущенных перед заголовком; M — остаток строк; по умолчанию K= 1, L = 2, M = 6. Из выходного массива выбираются K строк и переносятся в формируемый текст без изменений. Эти строки, кроме того, участвуют в формировании содержания документа. Если текст располагается в начале страницы, то чистые строки перед заголовком не выводятся, а если на странице осталось меньше M строк, то заголовок переносится на другую страницу.

ГЛАВА 11

СИСТЕМА ОТОБРАЖЕНИЯ ГОТОВЫХ ВИДЕОГРАММ

Система ЛИСТЕР обеспечивает оперативное информирование административного персонала путем выдачи на дисплеи в режиме «запрос — ответ» данных из хранимого на томе прямого доступа массива заранее сформированных дисплейных форм видеogramм. Настоящий комплекс программ предназначен для накопления, коррекции, отображения и защиты от

несанкционированного доступа текстовой информации, сформированной в виде дисплейных видеограмм.

Система ЛИСТЕР функционирует под управлением ОС ЕС версии 6.1 в режиме MVT или SVS, либо в среде системы ОБЬ, либо как автономная проблемная задача. В настоящее время осуществляется управление терминалами ЕС-7906, ЕС-7920 (локальный и удаленный варианты) и ЕС-8562 (вариант VTS-56100 или ИЗОТ-7925). Спектр поддерживаемых терминальных устройств постоянно расширяется. При помощи данной системы обеспечивается обслуживание терминальных сетей, содержащих порядка ста дисплеев. Требуемая оперативная память составляет 30 Кбайт плюс 2 Кбайта на каждый входящий в терминальную сеть видеотерминал. Чтение и корректировка документа одновременно возможны с нескольких терминалов, а создание, обновление и удаление только с одного. Синхронизация работы терминальных подзадач, а также обращений к массиву документов осуществляется путем использования макрокоманд ENQ и DEQ.

Документ является основной логической информационной единицей в системе ЛИСТЕР. Он представляет собой упорядоченную последовательность экранных страниц. База данных системы, называемая массивом дисплейных форм, состоит из оглавления и набора хранимых документов. Оглавление предназначено для хранения элементов описания документов. В каждом из этих элементов содержится имя документа; пароль; указатель первой страницы документа; указатель последней страницы документа; количество составляющих документ экранных страниц; счетчик числа обращений со стороны пользователей к данному документу.

Требуемый документ определяется в массиве при помощи указателя первой страницы. В связи с тем, что все страницы документа связаны между собой в двухсвязный список, имеется возможность просмотра документа в прямом и обратном направлениях. Наличие в оглавлении указателя списка свободных страниц позволяет создавать и удалять как целые документы, так и отдельные их страницы.

Для упрощения работы с системой ЛИСТЕР для дисплейных комплексов ЕС-7920 пользовательские команды реализуются при помощи клавиш СТРН ЭКР, ВЫЗОВ ТЕСТА, ВВОД, а также клавиш функциональной клавиатуры (ПД1 — ПД3, ПФ1 — ПФ12).

Диагностика в системе ЛИСТЕР выдается на терминал в первых позициях первой строки экрана, затем располагается восьмисимвольное поле, предназначенное для ввода запроса. Далее выдаются индексы функциональных клавиш, отсутствующих на терминалах, снабженных клавиатурой подготовки данных. С помощью этих индексов пользователь осуществляет имитацию нажатия клавиш: выбирает нужный индекс фотоселектором или подводит курсор под любую цифру индекса, нажимая при этом клавишу ВВОД.

Команды системы ЛИСТЕР могут потребовать ввода дополнительной информации в 8-байтное поле, в начало которого устанавливается курсор после выдачи каждой диагностики. В этом случае следует сначала набрать требуемые данные, а затем нажать соответствующую функциональную клавишу. Если клавиша нажата, а поле ввода содержит пробелы, то выдается диагностическое сообщение ВВЕДИТЕ КОМАНДУ. Большинство команд требует многократного обмена с системой — ввода имени документа, пароля и др. Этот обмен осуществляется нажатием одной и той же клавиши, соответствующей выбранной функции. При вводе паролей поле ввода маскируется и вводимая информация на экране не отображается. Если на экран выдана диагностика ЖДИТЕ, следует подождать завершения обработки предыдущей команды. Новые команды в это время не воспринимаются.

Функциональные клавиши реализуют следующие команды системы ЛИСТЕР: ВЫЗОВ ТЕСТА — вызов таблицы команд системы и содержимого оглавления; ВВОД — отображение первой страницы документа, СТРН ЭКР — восстановление страницы документа, читавшегося в тот момент, когда буфер терминала был занят сообщением оператору или от него; переход к первой странице документа, если в данный момент читается не первая страница; завершение работы (исполняется с повторным нажатием этой клавиши); ПД1 — запрос на выполнение команд создания, обновления (расширения и сокращения), удаления, переименования защищенных паролями документов; ПД2 — форматизация экрана для посылки сообщения и передачи его при повторном вводе команды; ПФ1 — чтение предыдущей страницы документа; ПФ2 — чтение следующей страницы документа; Г1Ф3 — запись страницы документа, ПФ4 —

печать страницы документа; ПФ5 — печать всего документа; ПФ6 — переименование документа; ПФ7 — добавление страницы документа после текущей страницы документа; ПФ8 — копирование страницы; ПФ9 — копирование документа под новым именем; ПФ10 — удаление текущей страницы; ПФ11 — удаление всего документа; ПФ12 — создание документа с указанным именем размером в одну страницу.

Команда, требующая многократного обмена, игнорируется, если ее обменный цикл выполнен не полностью, но пользователем уже начата работа с другой командой. Снятие защиты документа достигается заданием пароля, состоящего из восьми пробелов. После двухкратного ввода неверного пароля оператор ЭВМ информируется о попытке несанкционированного доступа. В этом случае документ становится доступным для чтения без предъявления пароля. При записи данных в поле ввода предшествующие вводимому тексту пробелы игнорируются.

В системе ЛИСТЕР обеспечивается распечатка высвечиваемых ранее дисплеев документов на АЦПУ, т. е. предусмотрена возможность получения «твердых копий». Диагностические сообщения системы выдаются со второй по двадцать четвертую позицию первой строки экрана. В этом случае используется режим отображения символов с повышенной яркостью. Время реакции системы при эксплуатации на локальных дисплейных комплексах ЕС-7920 не превышает двух секунд.

Запись принятой из базы данных информации в архив системы обеспечивается наличием автономного архивного блока загрузки информации. При этом архивизируемая информация оформляется в виде видеограмм (документов системы) для дальнейшего использования. Система ЛИСТЕР не накладывает ограничений на использование в программах пользователей средств ОС ЕС. Программы ввода информации должны быть составлены на языке ассемблера. Управление терминалами осуществляется при помощи базисного телекоммуникационного метода доступа ОС ЕС.

В режиме диалога пользователь имеет возможность вызывать на экран дисплея любой документ из архива системы; листать его в прямом и обратном направлениях через одну или через несколько страниц, корректировать информацию в документе; создавать новый документ или уничтожать старый; защищать документ паролем; распечатывать на устройстве документирования весь документ или отдельную его страницу; передавать и принимать сообщения оператора ЭВМ. ЛИСТЕР позволяет каждому пользователю иметь свою программу выдачи информации на персональное устройство документирования ЕС-7934, которое может находиться на его рабочем месте.

В состав системы входят программы копирования архива, предназначенного для ввода и хранения документов, а также для накопления статистических данных о работе пользователей с документами.

Вызов нужной программы ввода информации осуществляется путем передачи имени оператора DD, описывающего исходный набор данных на МД или МЛ в шаге задания на загрузку или обновление, в основную программу автономного блока. После анализа имени оператора DD основная программа автономного блока загрузки информации вызывает требуемую программу ввода информации из библиотеки загрузочных модулей и передает ей управление. При этом нельзя использовать имена операторов DD, описывающих наборы данных, отличные от принятых в системе, и нельзя включать в задание на загрузку операторы DD, описывающие те наборы данных, для которых нет соответствующих программ ввода в библиотеке загрузочных модулей системы.

Программы ввода поставляются вместе с настроенным вариантом системы или составляются персоналом предприятия, на который возложены функции по ее эксплуатации. В составе автономного блока поставляется программа формирования базы данных. Формируемые наборы данных должны иметь длину логической записи равную 80 байтам и содержать подготовленную для отображения информацию, т. е. информацию, оформленную в виде видеограмм. В системе ЛИСТЕР обрабатываются данные следующих типов: сообщения диалога с пользователем, диагностика, команды пользователя, информация, возникающая в результате работы автономного блока загрузки.

Диагностические сообщения выводятся в верхнюю левую часть экрана дисплея или на

консоль оператора.

Команды системы ЛИСТЕР вводятся пользователями в диалоговом режиме. Атрибуты команд вводятся в специальное поле экрана, начало которого помечено курсором, а сама команда задается путем нажатия одной из соответствующих функциональных клавиш. В случае, если используются терминалы, не имеющие функциональной клавиатуры, в системе ЛИСТЕР предусмотрен режим имитации отсутствующих функциональных клавиш, предполагающий идентификацию требуемых команд либо при помощи курсора и клавиши ВВОД, либо путем использования фотоселектора.

В модернизированном варианте системы ЛИСТЕР, разработанном В. Н. Низкоусом, работа пользователя может вестись в двух режимах отображения видеogramм: основном и полиэкранном. В основном режиме (задается по умолчанию) видеogramмы отображаются на полном объеме экрана для дисплеев типа ЕС-7927, причем пользователь использует весь набор системных команд.

При работе в полиэкранном режиме отображения информации пользователи работают со страницами документов размером 23 строки на 40 столбцов, т. е. имеющих вдвое уменьшенный в ширину формат, что позволяет вызывать на экран дисплея разные страницы одного и того же документа или страницы разных документов. Для перехода в полиэкранный режим пользователю необходимо ввести специальную команду переключения режима отображения. Набор команд, используемых в режиме полиэкранного отображения, отличается от основного набора. Поэтому для облегчения работы пользователей в системе предусмотрена специальная команда ПОДСКАЗКА, информирующая о командах, которые могут быть исполнены системой в текущий момент работы. Для возврата из полиэкранного режима в основной достаточно задать команду переключения режимов.

В процессе своей работы система ЛИСТЕР выдает сообщения пользователям терминалов и оператору ЭВМ. Пользователь системы ЛИСТЕР получает диагностические сообщения в первой строке экрана дисплея (на дисплеях типа ЕС-7927 с повышенной яркостью отображения).

ГОТОВО — выполнение заданной команды успешно завершилось. Система готова к приему новых команд.

ЖДИТЕ — принятая команда требует определенное время на обработку, все введенные в это время команды игнорируются.

ВВЕДИТЕ КОМАНДУ — нажата клавиша ВВОД, но параметры команды не заданы. В этом случае введенная команда игнорируется и пользователь должен повторить ее ввод.

ПОВТОРИТЕ КОМАНДУ — произошел сбой в работе дисплея, вводимая в этот момент команда игнорируется. Ее ввод необходимо повторить.

ТЕРМИНАЛ СВОБОДЕН — дисплей логически отключился от системы ЛИСТЕР. Для того чтобы подключиться к системе ЛИСТЕР вновь, пользователю необходимо ввести произвольную команду. Например, нажатием клавиши ВВОД можно ввести «пустую» команду.

РАБОТА ЗАКОНЧЕНА — система ЛИСТЕР закончила сеанс связи с данным дисплеем. На АЦПУ выдаются копии предназначенных для печати страниц документов.

КОМАНДА ЗАПРЕЩЕНА — введена команда, защищенная системным паролем. В этом случае команда игнорируется, система ожидает ввода новых команд. Пользователю необходимо ввести системный пароль и повторить ввод нужной команды.

ПЕЧАТЬ ЗАПРЕЩЕНА — при запуске системы данный терминал запущен без программы печати или при вызове системы не указан номер запроса с индексом программы печати (это происходит в случае работы системы ЛИСТЕР под управлением системы ОБЬ). Команды печати документов, выдаваемые пользователями, игнорируются. Необходимо повторить вызов пакета ЛИСТЕР с указанием номера необходимого запроса.

ВВЕДИТЕ ПАРОЛЬ — введена команда типа команды ЧТЕНИЕ ДОКУМЕНТА, но требуемый документ защищен паролем. Пользователь должен ввести с позиции курсора пароль.

ОШИБКА. ПОВТОРИТЕ — введен неверно или пароль, или номер страницы документа при копировании. Система ЛИСТЕР введенный пароль или номер страницы документа игнорирует и ожидает ввода правильного значения.

ПАРОЛЬ НЕВЕРЕН — пользователем повторно введен неправильный пароль. Система игнорирует введенный пароль и сообщает оператору ЭВМ о возможной попытке несанкционированного доступа к документу. Пользователю надо повторить ввод текущей команды, и при этом введя правильно пароль. Если он забыл пароль своего документа, то следует обратиться к системному программисту, отвечающему за сопровождение системы ЛИСТЕР.

ВВЕДИТЕ НОВЫЙ ПАРОЛЬ — требуется ввести с позиции курсора новый пароль при создании или переименовании документа, а также в случае смены системного пароля. Если пароль не нужен для создаваемого документа, пользователь должен ввести состоящий из пробелов «пустой» пароль.

Ряд сообщений для пользователей терминалов выдается системой ЛИСТЕР для информирования о ситуациях, возникающих при работе непосредственно с документами.

НАЧАЛО ДОКУМЕНТА — команда **ЧТЕНИЕ ПРЕДЫДУЩЕЙ СТРАНИЦЫ ДОКУМЕНТА** отвергнута в связи с тем, что предыдущая страница отсутствует.

ВВЕДИТЕ НОВОЕ ИМЯ — требуется ввод нового имени при переименовании документа.

ДОКУМЕНТ ОТСУТСТВУЕТ — требуемый документ не найден в архиве документов. Пользователю необходимо проверить правильность задания имени требуемого документа при помощи системного оглавления архива документов.

ДОКУМЕНТ ЗАНЯТ — совершена попытка обновить или удалить документ, используемый в настоящее время другим пользователем.

НЕТ МЕСТА В АРХИВЕ — отсутствуют свободные страницы в архиве документов. Система отвергает все команды, требующие увеличения размера документа. Пользователь должен удалить из архива ненужные документы или обратиться к системному программисту, ответственному за сопровождение системы ЛИСТЕР, с заявкой на увеличение объема памяти, отведенной под архив.

ВВЕДИТЕ НОМЕР СТРАНИЦЫ — для выполнения команды копирования необходимо, чтобы пользователь ввел порядковый номер страницы документа или отказался от выполнения данной операции путем использования команды **СБРОС**.

КОНЕЦ ОГЛАВЛЕНИЯ — достигнут конец оглавления при его просмотре путем задания команды **ПОДСКАЗКА**.

ИМЯ ПОВТОРЯЕТСЯ — при создании или переименовании документа введено имя, повторяющее уже имеющиеся в оглавлении. Необходимо повторить ввод команды с указанием нового имени, отличного от имеющихся в оглавлении.

НЕТ МЕСТА В ОГЛАВЛЕНИИ — новый документ не может быть создан, так как переполнилось оглавление архива документов. Система отвергает все команды, требующие создания новых документов. Необходимо удалить ненужные документы из оглавления архива или обратиться к системному программисту, ответственному за сопровождение системы с требованием увеличить размер оглавления архива.

УДАЛИТЬ ДОКУМЕНТ? — введен запрос на удаление документа и требуется подтверждение этой операции. Система переспрашивает пользователя и ожидает ввода подтверждения, либо отказа от введенной ранее команды. Для подтверждения необходимо повторно ввести команду **УДАЛЕНИЕ ДОКУМЕНТА**, а для отказа любую другую команду.

ЗАКОНЧИТЬ РАБОТУ? — введен запрос на завершение работы с системой. Если пользователь желает закончить работу, то необходимо повторно ввести команду завершения работы. Если нет, то надо задать любую другую команду.

Система ЛИСТЕР выдает сообщения на консоль оператора в следующем формате:

RR LSTXXXU ТЕКСТ СООБЩЕНИЯ

где RR — идентификатор ответа сообщения; LST — шифр сообщений системы ЛИСТЕР, позволяющий отличать ее сообщения от сообщений других систем, работающих одновременно с ней; XXX — порядковый номер сообщения в системе ЛИСТЕР; U — тип сообщения, принимающий следующие значения: I — информационное сообщение, не требующее ответных действий со стороны пользователя или оператора ЭВМ; D — сообщение, описывающее ситуацию и требующее принятия решения оператора ЭВМ; A — сообщение, описывающее

ситуацию и приостанавливающее работу системы.

В сообщениях типа D или A обычно указываются варианты ответа или действия, которое необходимо выполнить. Система не будет продолжать свою работу до получения правильного ответа на сообщение.

Сообщения, выдаваемые на консоль оператора ЭВМ, содержат информацию, идентифицирующую сбои в работе терминальных устройств; несоответствия, возникающие при работе с архивом документов; сбои, возникающие при выполнении операций ввода-вывода на накопителях на магнитных дисках; ошибки в командах оператора ЭВМ; статус указанных терминалов и т. д.

Персонал, которому поручена эксплуатация системы ЛИСТЕР, в процессе подготовки ее к работе должен переписать библиотеку загрузочных модулей с поставляемой МЛ на МД, отвести память на МД под архивную базу данных, подготовить задания для инициализации архива и запуска системы, провести инициализацию системы.

Выгрузка библиотеки загрузочных модулей производится утилитами ОС ЕС IENMOVE или IEBCOPY. Память под архивный файл системы отводится в соответствии с соглашениями, принятыми в ОС ЕС для выделения памяти для последовательного набора данных на МД. Параметры DCB в метке, описывающей этот набор, должны быть следующими: DSORG = PS, RECFM = F, LRECL = 1614, BLKSIZE = 1614.

Инициализация архива системы осуществляется в пакетном режиме программой LSTFINIT. Требуемое для работы программы LSTFINIT количество элементов оглавления (каждый элемент оглавления описывает один документ) передается через поле PARM. При этом следует помнить, что чем больше задано элементов оглавления, тем больше различных документов можно хранить в архиве. Однако увеличение размера оглавления увеличивает требуемый объем оперативной памяти, так как во время работы системы оглавление архива полностью располагается в ней.

Первый элемент оглавления формируется во время инициализации архива и содержит количество элементов в оглавлении, число страниц в архиве, количество страниц оглавления, указатель на первую страницу справочного или «главную» документа, описывающего содержание архива документов, указатель на первую свободную страницу в списке свободных страниц, количество страниц главного документа, количество свободных страниц, системный пароль. Последующие элементы оглавления служат для описания хранимых в архиве документов. Они имеют следующий формат: имя документа, пароль, указатель на первую страницу документа, указатель на его последнюю страницу, количество страниц в документе, количество обращений к документу.

При формировании задания на запуск системы ЛИСТЕР, состоящего из одного шага, в поле PARM оператора EXEC указывается число подключаемых терминалов к системе, определяется программа печати для каждого из них. По одному оператору DD требуется для описания библиотеки загрузочных модулей, файла архива системы и каждого подключаемого терминала. Имена операторов DD, описывающих архив и терминалы, имеют фиксированную структуру.

При разработке программного обеспечения системы ЛИСТЕР использовался принцип модульности. Система функционирует под управлением ОС ЕС как проблемная задача. Управляющие секции LSTHNDLR, LSTDIALG, LSTIOTRM, LSTPRESS, LSTDPLOY отредактированы в виде реентерабельного загрузочного модуля LSTHNDLR. Контрольные секции LSTSTRTR и LSTIOFLE объединены в загрузочный модуль LSTSTRTR.

Управление от операционной системы получает модуль LSTSTRTR. Вызов модулей LSTSEANC, LSTDINIT и LSTFINSH осуществляется по макрокоманде LINK. Модуль LSTHNDLR получает управление по макрокоманде ATTACH, при этом через область параметров ему передаются адреса входов в программу LSTIOFLE и модуль выхода COPYXX. Программа COPYXX загружается модулем LSTSRTR по макрокоманде LOAD. Запустив подзадачи обслуживания терминалов, модуль LSTSTRTR не теряет с ними связь. Модули могут обмениваться сообщениями, отмечая с помощью макрокоманды POST блоки ECB, адреса которых известны обоим взаимодействующим программам. Модуль LSTSTRTR может передать терминальной подзадаче LSTHNDLR приказ принять сообщение от оператора ЭВМ или завершить работу. При этом модуль LSTHNDLR ведет общий для всей системы учет

активных терминалов и, в случае логического отключения всех терминалов сообщит об этом модулю LSTSTRTR. Это вызовет исполнение команды завершения работы всей системы.

Программа LSTSTRTR является главной управляющей программой системы, основными функциями которой являются: вызов программ LSTSEANC и ISTDINIT; загрузка требуемых программ выхода; запуск задач терминального обслуживания LSTHNDLR; обслуживание связи с оператором ЭВМ; связь с программами терминального обслуживания в случае передачи сообщений от оператора ЭВМ и приказов о завершении работы; восстановление терминальных подзадач после их аварийного завершения; вызов программы LSTFINSH. Для выхода на связь с оператором ЭВМ программа LSTSTRTR выдает макрокоманду WTOR со следующим сообщением:

ВВОДИТЕ КОМАНДЫ СВЯЗИ (C/D LISTER/TERM XX)

Программа LSTSEANC, управляемая программой LSTSTRTR по макрокоманде LINK, анализирует переданное ей поле PARM для определения параметров сеанса связи и возвращает их в программу LSTSTRTR в отформатированном виде. Заданные параметры сеанса программа LSTSEANC передает оператору ЭВМ. Обнаружив недопустимые параметры, программа LSTSEANC выдает специальное сообщение с требованием ввода новых параметров.

Программа LSTDINIT инициализирует управляющие блоки, считывает оглавление с помощью программы LSTIOFLE и проверяет корректность системы указателей архива документов, подсчитывая число страниц, учтенных в оглавлении, и сравнивая это число с размером архива. При этом возможна задача сообщений о нарушении системы указателей и архиве документов.

В целях устранения интерференции между терминальными подзадачами при создании, обновлении или уничтожении документов задачи LSTHNDLR синхронизированы с помощью макрокоманд ENQ/DEQ. Чтение и коррекция документа допускается одновременно с нескольких дисплеев, а создание, обновление или удаление — только с одного. С помощью макрокоманд ENQ/DEQ синхронизируются также обращения к модулю работы с архивом документов LSTIOFLE и одноименному модулю COPYXX.

Программа LSTFINSH, управляемая программой LSTSTRTR по макрокоманде LINK, завершает работу программ COPYXX и удаляет их из основной памяти, а также освобождает память, занимаемую управляющими блоками системы и оглавлением архива документов.

Программа LSTIOFLE осуществляет ввод-вывод записей архива документов, описанного оператором DD с именем DATA и инициализированного программой LSTFINIT. Набор данных открывается на обновление с использованием метода доступа EXCP. Каждая запись набора данных соответствует одной странице документа. Выход на нужную страницу осуществляется при помощи использования ее номера, который служит указателем. При этом каждая запись имеет фиксированный размер. Неудачное завершение операции ввода-вывода в архив идентифицируется выдачей соответствующего служебного сообщения и задача, выдавшая запрос на ввод-вывод, заканчивается аварийно с кодом U001.

Программы LSTHNDLR и LSTDIALG, реализующие диалоговый режим работы системы, обслуживают команды, поступающие от пользователей, а также директивы программы LSTSTRTR. При этом программа LSTHNDLR представляет основной обрабатывающий модуль, а программа LSTDIALG — модуль предварительной обработки. Поступившая от пользователя терминала команда попадает на обработку программе LSTDIALG, которая пытается обработать ее своими средствами. Если этого оказывается недостаточно, то на дисплей пользователя выдается диагностика ЖДИТЕ и управление переходит к программе LSTHNDLR. Обмен информацией между программами осуществляется через общие области, описываемые тремя фиктивными секциями. При этом секция LSTPARAM записывается в память, предоставленную программой LSTSTRTR, а две другие секции накладываются на загрузочные (нереентерабельные) модули, помещаемые в основную память программой LSTHNDLR по макрокоманде LOAD.

Программы LSTPRESS и LSTDPLOY предназначены для дополнительного разделения доступа к архиву документов и уплотнения информации па томе прямого доступа. В системе ЛИСТЕР применен следующий формат хранения страниц документов: текстовая информация

перекодируется в специальный 7-битный код, после чего информация уплотняется с исключением 8-го бита. Процедуры сжатия информации и ее приведения к отображаемому виду постранично осуществляют программы LSTPRESS и LSTDPLY.

Программа LSTIOTRM осуществляет операции ввода-вывода информации на терминалы пользователей, используя при этом базисный телекоммуникационный метод доступа. В случае возникновения неисправимой ошибки ввода-вывода на дисплей пользователя выдается соответствующее информационное сообщение и подзадача терминального обслуживания завершается аварийно с кодом UXXX, где XXX — код возврата макрокоманды READ или WRITE.

В описываемой системе предусмотрена возможность постраничной пересылки документов программам выхода, разработанных пользователями. Указанная передача происходит в случае использования команд ПЕЧАТЬ ДОКУМЕНТА и ПЕЧАТЬ СТРАНИЦЫ ДОКУМЕНТА. Программа пользователя должна иметь имя COPYXX, где XX — порядковый номер программы выхода, соответствующий номеру терминала пользователя, и записываться в пригодном для исполнения виде в библиотеку загрузочных модулей. Программа COPY01 поставляется вместе с системой и обеспечивает вывод страниц на печать. В этой программе использован метод доступа QSAM, что обеспечивает независимость задачи от устройства вывода.

Программа инициализации архива документов LSTFINIT запускается на исполнение в режиме пакетной обработки. Информация о количестве элементов оглавления передается ей через поле PARM. Одна страница архива (1614 байт) содержит 67 элементов оглавления. При недостаточности объема памяти, зарезервированного под архив, или в случае задания ошибочного значения через поле PARM строится одна страница оглавления, содержащая 66 элементов. Оставшаяся часть набора данных форматизируется под свободные страницы, объединяемые при помощи списка свободного пространства. Строится также справочный документ, содержащий одну чистую страницу. При этом на печать выдается следующая информация: заголовок оглавления, содержащий число элементов оглавления; число страниц архива, оглавления, справочного документа и свободного пространства; системный пароль; указатели начала справочного документа и начала свободного пространства.

Программа печати оглавления архива документов LSTDLIST работает в пакетном режиме. Для доступа к архиву документов используется программа LSTIOFLE, которая выдает на печать заголовок оглавления архива, содержащий число элементов оглавления, число страниц архива и оглавления, пароль, указатель на начало справочного документа и т. д. Затем печатается таблица элементов оглавления.

Пакетная программа загрузки документов в архив системы ЛИСТЕР (автономный блок загрузки) предназначена для ввода новых документов в систему и периодического обновления информации в документах. Функционально автономный блок состоит из вызывающей программы SELDDN, программы ввода и обработки информации из последовательных массивов PRXXX и шаблонов видеogramм FORMAXXX. Обычно номер документа в системе соответствует номеру XXX обрабатывающей программы и шаблону FORMAXXX.

Программа SELDDN осуществляет поиск в задании на загрузку документов, имеющих имена DD, начинающиеся с префикса FORMA или UFURM. Имена DD наборов данных, начинающихся с префикса FORMA, обрабатываются соответствующей программой PRXXX. Имена DD наборов данных, начинающихся с префикса UFURM, обрабатываются одной из программ ввода данных из библиотечных или последовательных наборов данных, имеющих следующие параметры DCB: LRECL=80, BLKSIZE = длина, кратная 80, DSORG = F или FB. Кроме того, программа SELDDN производит вызов программ ввода и обработки информации PRXXX, необходимых для введенного на обработку задания. Если в именах DD неверно заданы номера XXX, то автономный блок информирует об этом оператора ЭВМ при помощи сообщения LST0151. Программы ввода и обработки информации получают от программы SELDDN имена наборов данных, с которыми им предстоит работать; формируют имена документов; ведут обработку исходных для системы данных; записывают при помощи программы LSTIOFLE документы в архив.

Программный комплекс ЛИСТЕР может функционировать как автономно, так и под централизованным управлением мультитерминальной системы распределенной обработки

данных ОБЬ. В последнем случае при поступлении запроса к системе ЛИСТЕР управление получает драйвер LSTDRIIVE, определяющий дальнейший порядок обработки.

Подключение под управление системы ОБЬ пакета ЛИСТЕР существенно расширяет по сравнению с его базовым вариантом комплекс обслуживаемых технических средств.

Для получения твердых копий документов необходимо сформировать средствами системы ОБЬ запрос, содержащий номер программы выхода COPYXX. Он будет храниться в файле работ системы ОБЬ. Выдача твердых копий документов обеспечивается как на АЦПУ, так и на удаленные устройства документирования типа ЕС-7934.

В качестве примера рассмотрим систему оперативного информирования руководства (СОИР), эксплуатируемую на одном из машиностроительных предприятий и созданную на основе комплекса программ ЛИСТЕР, СОИР предназначена для накопления, коррекции, защиты и отображения на дисплее текстовой информации. Одновременно она может обслужить до 100 дисплеев типа ЕС-7920 и такое же число печатающих устройств типа ЕС-7934. Единицей информации в системе является документ, состоящий из последовательности страниц, каждая из которых соответствует 23 строкам экрана дисплея. Страницы в документе связаны между собой в двухсвязный список, что позволяет читать документ постранично в прямом и обратном направлениях. Данная организация архивного массива системы позволяет сравнительно легко добавлять и удалять страницы из хранимых документов.

СОИР позволяет выполнять следующие функции: создавать новые документы; корректировать документы с экрана дисплея; копировать документы как целиком, так и отдельными страницами; выводить на печатающие устройства (АЦПУ или печатающие устройства типа ЕС-7934) как целые документы, так и отдельные их страницы; удалять документ из архивного массива системы; листать каждый документ в любом из двух направлений через указанное количество экранных страниц; выдавать на экран дисплея страницы разных документов (так называемый полиэкранный режим, при котором отображаемой единицей информации является 1/2 страницы документа).

Все команды СОИР пользователь набирает, используя функциональную клавиатуру дисплея. Каждой команде соответствует определенная клавиша. В наборе команд, исполняемых системой, имеется команда-подсказка, которая выдает на экран список команд и обозначения соответствующих им функциональных клавиш.

В состав программного обеспечения СОИР входит набор сервисных программ, в том числе программа загрузки/обновления массива документов и программа архивизации массива отображаемых документов.

К достоинствам СОИР следует отнести в первую очередь малое время реакции системы на запросы пользователей (менее 1 с); высокую живучесть системы за счет использования иерархии подзадач; наличие специального блока диагностики, повышающего устойчивость работы системы; возможность сбора статистических данных по работе с отображаемыми документами; использование режима «полиэкрана», позволяющего производить анализ и сопоставление информации из различных документов на экране дисплея.

На одном из предприятий средствами СОИР работники административно-хозяйственных служб оперативно информируются путем отображения на терминалах дисплейных видеogramм следующих типов:

	Количество форм
Ход, ритмичность и результаты производства	11
Технико-экономические показатели по подразделениям	18
Исполнительская дисциплина	9
Состояние технологической дисциплины и культуры производства	8
Анализ и восстановление рекламационных изделий	2
Выполнение плана оргтехмероприятий по снижению трудоемкости	2
Работа подсобною хозяйства	2

Директор завода получил возможность при помощи СОИР проводить более эффективно производственные совещания по вопросам организации и совершенствования планово-экономической работы (еженедельно), состоянию выполнения производственной программы и договоров (2 раза в неделю), состоянию исполнительской дисциплины (еженедельно), работе подсобного хозяйства, еженедельные производственные совещания. Главный инженер, заместитель директора по экономике, начальник производства при проведении перечисленных ниже совещаний обязательно пользуются услугами СОИР. Главный инженер — по состоянию исполнительской дисциплины (еженедельно), техническим вопросам подготовки производства (еженедельно), вопросам планирования подготовки производства новых изделий (2, 3-я недели месяца), техническим вопросам производства изделий (1, 3, 4-я недели месяца), культуре производства (еженедельно). Заместитель директора по производству и экономике и начальник производства — оперативные совещания по итогам работы за прошедшие сутки (ежедневно), еженедельные совещания начальников планово-диспетчерских бюро цехов.

Двухлетний анализ использования СОИР позволяет сделать вывод о том, что система способствует обеспечению ритмичности работы основных и вспомогательных цехов, соблюдению плановых сроков освоения новых изделий, повышению фактических коэффициентов оценки состояния технологической дисциплины и культуры производства.

В результате внедрения системы СОИР повысилась производительность управленческого труда. В первую очередь за счет улучшения организации процессов учета, контроля, принятия управленческих решений, планирования и координации работ; своевременного и полного получения управленческой информации. Помимо этого повысилась достоверность информации, поступающей на верхний уровень управления предприятием. По сравнению с технологией, существовавшей до внедрения СОИР, произошло частичное высвобождение персонала, занятого рутинными операциями по переработке данных.

ГЛАВА 12

ДИАЛОГОВАЯ СИСТЕМА ПРОСМОТРА И КОРРЕКТИРОВКИ ДАННЫХ, ОРГАНИЗОВАННЫХ В ВИДЕ ПОСЛЕДОВАТЕЛЬНЫХ ФАЙЛОВ

К классу мультитерминальных систем безбумажного информирования относится пакет ТЕРМЕС, являющийся диалоговой системой просмотра и корректировки данных, хранящихся в последовательных файлах. Последовательные (линейные) массивы информации ввиду простоты их сопровождения нашли широкое использование в различных приложениях и, в частности, при построении баз данных АСУП. На первых стадиях разработки автоматизированных систем традиционным режимом использования массивов подобной организации был пакетный. Он включал этапы подготовки данных на перфокартах, ввод данных, сортировку информационных массивов с целью ускорения поиска требуемых данных, ввод запросов и получение результирующих справочных данных в виде распечаток на АЦПУ.

Длительность поиска информации является основным недостатком использования линейных массивов. Частичное устранение этого недостатка может быть достигнуто за счет сортировки массива по возрастанию (убыванию) ключевых полей. Это позволяет уменьшить среднее время поиска, главным образом за счет сокращения времени, затрачиваемого на установление фактов отсутствия в массиве записей с указанными в запросах значениями ключей.

Надежность хранения информации, простота сопровождения и возможность осуществления сложного поиска — основные достоинства массивов, имеющих последовательную организацию, благодаря которым вопрос их использования при построении систем автоматизированной переработки данных по-прежнему актуален. Рассмотрим подробнее положительные свойства, которые определяют ряд существенных преимуществ линейных массивов по сравнению с базами данных, организованными в виде некоторых структур записей с указателями или индексно-последовательных массивов различной организации.

Надежность хранения информации. Линейный массив содержит записи, состоящие только из

информационных или ключевых полей, причем некоторые поля могут служить одновременно и полем ключа, и информационным. На возможность доступа к определенной записи не влияет содержание других записей массива. Следовательно, даже при разрушении или искажении одной или нескольких записей массива, в целом он остается пригодным к использованию. Напротив, разрушение структур указателей или индексных полей сложных баз данных может привести к логической деформации базы и даже сделать ее полностью непригодной для эксплуатации.

Другим положительным свойством линейных массивов является простота их сопровождения. Для поддержания в рабочем состоянии и обслуживания линейных массивов информации не требуется сложных специальных программ (утилит). В отличие от развитых баз данных процедуры копирования (восстановления), заполнения, коррекции и сортировки линейных массивов могут быть выполнены при помощи стандартных средств операционной системы или в отдельных особых случаях простыми программами пользователей.

Последовательная организация данных представляет возможность осуществления сложного поиска информации. Индексно-последовательные массивы и структуры с указателями эффективны при поиске данных по одному — двум ключевым полям, а именно, по тем ключам, по которым и составлена упорядоченная структура базы данных. Осуществить сложный многоключевой поиск данных или поиск по контексту, т. е. по информационным, а не ключевым полям, в таких структурах затруднительно. Поскольку поиск в линейном массиве сводится к последовательному просмотру записей, то количество ключей поиска не имеет существенного значения. Причем ключевые и информационные поля могут перекрываться и накладываться друг на друга. Таким образом, в случае сложных и непредсказуемых запросов на выборку информации из базы данных линейные наборы данных могут оказаться предпочтительнее других файловых организаций. Развитый библиотечный метод доступа ВРАМ ОС ЕС позволяет свести несколько последовательных массивов в один набор данных — библиотеку, к любому разделу которой операционная система обеспечивает доступ по его имени, служащим одновременно и индексом поиска. Доступ к разделу библиотеки осуществляется пользователями элементарными средствами ОС ЕС. При этом объем организованной таким образом базы данных ограничивается емкостью используемого тома прямого доступа.

Преимущества, описанные выше, обеспечили в свое время линейным массивам информации широкое распространение при создании баз данных различных подсистем АСУ. Переход на качественно новую технологию безбумажного информирования вызвал насущную необходимость создания специальных программных средств, обеспечивающих получение справок в диалоговом режиме из множества ранее сформированных в пакетном режиме линейных массивов информации. Одним из таких созданных средств является пакет программ ТЕРМЕС. Прежде чем перейти к описанию диалоговой информационно-поисковой системы ТЕРМЕС, сформулируем ряд требований, выдвинутых перед ее разработкой.

1. Система рассматриваемого класса должна быть мультитерминальной и обеспечивать одновременный доступ к базе данных нескольких пользователей с исключением недопустимой интерференции.

2. Должна быть обеспечена независимость программ от типа устройства, на котором хранится база данных системы.

3. Объем массива должен быть ограничен несколькими тысячами записей. При получении информации в диалоговом режиме время ответа на запрос не должно превышать одной минуты.

4. Настройка системы на конкретную базу данных должна быть обеспечена несложными средствами, доступными программисту средней квалификации.

Система ТЕРМЕС построена с учетом принципов адаптивности программного обеспечения мультитерминальных диалоговых систем. Реализуется выборочный просмотр информации из линейного массива с поиском данных по задаваемым пользователем ключам, дополнительная обработка выбранной для просмотра информации типа формирования итоговых значений, получение на печатающем устройстве «твердых» копий видеogramм.

Система ТЕРМЕС может функционировать под управлением мультитерминальной системы ОБЬ, а также непосредственно в среде ОС ЕС версии не ниже 6.1 в режимах MVT или SVS.

Количество обслуживаемых терминалов не должно превышать заложенного в ОС ЕС максимального значения счетчика использований реентерабельного модуля — 225. Допускается одновременный доступ с разных терминалов к различным линейным массивам, а также одновременный линейный просмотр одного массива с нескольких терминалов системы. Общий объем требуемой ОП зависит от количества обслуживаемых терминалов и линейных массивов. Размер памяти, занимаемый собственно программным обеспечением ТЕРМЕС, равен 10 Кбайт.

Отображение информации осуществляется на видеотерминалах локального дисплейного комплекса ЕС-7920, а также абонентского пункта ЕС-8562 в вариантах VTS-56100 и ИЗОТ-7925. Используемый линейный массив может храниться на любых применяемых в ЕС ЭВМ устройствах ввода-вывода на магнитных носителях информации. Он может быть разделом библиотечного набора данных на томе прямого доступа. Копирование (распечатка) видеограмм производится на печатающие устройства и компоненты печати, входящие в состав комплексов ЕС-8562 и ЕС-7920. Ввиду мобильности программного обеспечения системы ТЕРМЕС нетрудно расширить состав обслуживаемых терминалов и устройств копирования,

Обеспечиваются поиск, преобразование и отображение на экранах дисплеев информации, хранящейся в линейном массиве, состоящем из записей фиксированной длины с произвольным блокированием. Каждая запись содержит произвольное количество полей фиксированной длины любого применяемого в ОС ЕС формата. Любое поле может быть отображено в видеограмме и служить ключом поиска. Массив упорядочен по возрастанию или убыванию ключей, по которым осуществляется поиск. При этом допускается произвольная глубина вложенности сортируемых полей. Возможен просмотр массива в прямом и обратном направлениях, а также сплошной просмотр массива без отсева информации. Сравнение данных массива с ключами, задаваемыми пользователями, производится на равенство, неравенство, больше, меньше и т. п. При необходимости в систему могут включаться модули дополнительной обработки. Они могут как корректировать и дополнять информацию видеограммы, так и накапливать данные для последующего отображения.

Иерархическая структура диалога системы ТЕРМЕС состоит из четырех уровней. На первом уровне диалога происходит отображение входной видеограммы системы, предлагающей пользователю ввести имя раздела библиотеки загрузочных модулей для настройки на требуемый массив. На втором уровне осуществляется отображение видеограммы, предлагающей пользователю выбрать с помощью клавиш функциональной клавиатуры один из форматов ввода ключей и вывода информации. Третий уровень идентифицируется отображением видеограммы, предлагающей пользователю ввести ключи и тип поиска. Тип поиска подразумевает реализацию операций «равенство», «неравенство» и т. д. На последнем уровне диалога происходит управляемый пользователем просмотр информации массива. Допускается чтение массива в прямом и обратном направлениях, с начала массива, а также копирование информации на печатающем устройстве. На каждом уровне диалога пользователь может не только перейти на следующий уровень, но и вернуться на предыдущий. Попытка возврата с первого уровня интерпретируется как запрос на завершение работы с ТЕРМЕС

Организуемый между ТЕРМЕС и пользователями терминалов диалог позволяет осуществить:

— выбор определенной задачи, решение которой базируется на получении ряда логически связанных между собой сводок, формируемых системой в диалоговом режиме;

— выбор интересующей пользователя сводки по указанной задаче (на этом этапе происходит настройка системы ТЕРМЕС на соответствующий этой сводке информационный массив);

— задание реквизитов поиска данных в информационном массиве с помощью видеограмм ввода ключей поиска;

— просмотр сводки, сформированной системой ТЕРМЕС на основе информации, заданной пользователем на предыдущем этапе.

Каждый уровень работы системы ТЕРМЕС характеризуется соответствующими ему типами видеограмм. Количество видеограмм третьего и четвертого уровней определяется количеством справок, получаемых по запросам пользователей для каждой конкретной задачи. Формирование видеограмм и их управляющих таблиц осуществляется на этапе подключения конкретной

задачи к системе ТЕРМЕС.

Выполнение задач первого уровня обусловлено возможностью системы одновременно поддерживать несколько различных задач, работающих с разными базами данных линейной структуры. Видеограмма этого уровня в общем случае «одержит» перечень задач, поддерживаемых системой, а также шифр и назначения функциональных клавиш для каждой задачи. Настройка системы на конкретную задачу с последующим переходом на следующий уровень диалога обеспечивается вводом с экрана терминала шифра требуемой задачи или нажатием соответствующей ей функциональной клавиши.

На втором уровне диалога пользователю предоставляется возможность выбора интересующей его справки по данной задаче. Видеограмма этого уровня содержит перечень справок и соответствующих им команд и названий функциональных клавиш. Настройка системы на нужную справку и переход на третий уровень диалога обеспечивается вводом команды (шифра справки) или нажатием соответствующей функциональной клавиши.

Выборка информации из линейного массива при формировании справки производится в соответствии с задаваемыми пользователями ключами. На третьем уровне предполагается работа с видеограммой ввода ключей. Под ключом подразумевается один из реквизитов записи информационного набора данных, представляющий совокупность символов, вводимых пользователем в систему с экрана терминала. Для получения более конкретной информации по справке может быть задано несколько ключей, так называемых уточняющих реквизитов. В этом случае поиск информации осуществляется по совокупности ключей. Количество задаваемых пользователем ключей определяется при формировании видеограмм ввода ключей для каждой справки на этапе подключения конкретной задачи к системе ТЕРМЕС. При задании ключей в первой позиции поля ключа указывается условие поиска: > (больше), < (меньше), ≥ (больше или равно), ≤ (меньше или равно), = (равно), ^ (не равно). Отсутствие условия поиска или пробел в первой позиции поля ключа интерпретируется как равенство. Если первым символом ключевого поля является *, то отбор записей по данному ключу не производится.

Переход на четвертый уровень, обеспечивающий работу с видеограммой справки, осуществляется после формирования ее первого кадра. На данном уровне пользователю предоставляется возможность выборочного просмотра с экрана терминала записей информационного набора данных в прямом и обратном направлениях, корректировки данных с экрана и записи их в информационный массив, получения распечатки кадра видеограммы справки, а также получения распечатки всей видеограммы справки, сформированной в соответствии с заданными ключами. После распечатки всей видеограммы система переходит на третий уровень, предоставляющий пользователю возможность задания новых ключей для получения интересующей его информации по справке.

Все программные модули ТЕРМЕС реентерабельны и в совокупности обеспечивают обслуживание одного терминала системы. При обслуживании нескольких терминалов пользователей строится соответствующее число подзадач.

Модуль обслуживания терминала TRST загружается в соответствии с типом терминала основным модулем системы ТЕРМЕС. Таким образом, в оперативной памяти находится столько модулей TRST, сколько типов (а не единиц терминалов) обслуживается системой в данный момент.

Диалог с пользователем ориентирован на использование функциональных клавиш дисплея ЕС-7927, что обеспечивает доступ к данным при минимальном оперировании клавиатурой. При работе с дисплеями абонентского пункта ЕС-8562, где функциональная клавиатура слабо развита, в качестве команд вводятся условные коды функциональных клавиш дисплея ЕС-7927.

Для настройки системы ТЕРМЕС на требуемые пользователями массивы и видеограммы применяются управляющие таблицы, хранящиеся в библиотеке загрузочных модулей. Использование таких таблиц позволяет обойтись без специальных блоков ввода и преобразования. По запросу пользователя нужные таблицы загружаются в ОП с помощью макрокоманды LOAD. Управляющие таблицы системы ТЕРМЕС можно разбить на две группы: видеограммы и описания (шаблоны). Таблицы-видеограммы — это трафареты, готовые к отображению на экране или требующие заполнения информацией. Таблицы-описания содержат указания о типах полей записей массива, их длинах, расположении в записи и т. д., а также

аналогичную информацию о видеограммах для отображения выбранной информации. Для составления таблиц описаний и видеограмм пользователю системы ТЕРМЕС предоставлены специальные макрокоманды. Процесс составления управляющих таблиц занимает у квалифицированного пользователя не более одного часа.

Классификация команд ТЕРМЕС в основном соответствует уровням диалога системы с пользователями терминалов,

В состав команд первого уровня входят команда ЗАКОНЧИТЬ РАБОТУ и команда настройки на задачу. По команде ЗАКОНЧИТЬ РАБОТУ завершается работа системы. Для выполнения этой команды необходимо с экрана терминала ввести в систему шифр СНЯТЬ или нажать клавишу СТРН ЭКР. По команде настройки на задачу система настраивается на базу данных, относящуюся к интересующей пользователя задаче, и переходит на второй уровень системы. Для выполнения этой команды необходимо с экрана терминала ввести шифр задачи пользователя или нажать соответствующую ей функциональную клавишу, которые определяются на этапе подключения конкретной задачи к системе.

В состав команд второго уровня входят команды ПЕРЕЙТИ НА ПЕРВЫЙ УРОВЕНЬ СИСТЕМЫ и команда настройки на справку. По команде ПЕРЕЙТИ НА ПЕРВЫЙ УРОВЕНЬ СИСТЕМЫ происходит переход системы со второго уровня на первый. Для выполнения этой команды необходимо с экрана терминала ввести шифр команды СНЯТЬ или нажать клавишу СТРН ЭКР. По команде настройки на справку система настраивается на нужную справку и переходит на третий уровень. Для выполнения этой команды необходимо с экрана терминала ввести шифр справки или нажать соответствующую ей функциональную клавишу, которые определяются на этапе подключения конкретной задачи к системе: Группу команд третьего уровня составляют три команды. По команде ПЕРЕЙТИ НА ВТОРОЙ УРОВЕНЬ СИСТЕМЫ происходит переход с третьего уровня на второй. Для выполнения этой команды необходимо с экрана терминала ввести в систему шифр команды СНЯТЬ или нажать клавишу СТРН ЭКР. По команде ввода ключей осуществляется поиск информации, удовлетворяющий значениям введенных ключей, формирование и вывод на экран терминала интересующей пользователя справки. Для выполнения этой команды необходимо с экрана терминала ввести значения ключей, по которым предполагается производить отбор записей информационного набора данных, и нажать клавишу ВВОД.

Команда РАСПЕЧАТАТЬ КАДР ВИДЕОГРАММЫ позволяет получить распечатку кадра видеограммы ввода ключей. Для выполнения этой команды необходимо ввести шифр ПЕЧАТЬ или нажать клавишу ПД2,

Наибольшую группу составляют команды четвертого уровня.

По команде ЗАКОНЧИТЬ ПРОСМОТР система завершает работу в режиме просмотра информации и переходит с четвертого уровня на третий. Для выполнения этой команды необходимо с экрана терминала ввести шифр команды СНЯТЬ или нажать клавишу СТРН ЭКР.

По команде ЧИТАТЬ ВПЕРЕД НА СТРАНИЦУ на экран терминала выводится следующая страница видеограммы справки. Для выполнения этой команды необходимо ввести шифр команды ВПЕРЕД или нажать клавишу ПФ1.

Внедрение ТЕРМЕС позволяет использовать уже имеющуюся на предприятиях информационную базу в режиме безбумажного оперативного информирования. Мобильность программного обеспечения системы ТЕРМЕС, широкий и допускающий дальнейшее расширение спектр обслуживаемых ею устройств, возможность формирования произвольных программ диалога, незначительный объем занимаемой ОП позволяют рекомендовать ТЕРМЕС для реализации на ЕС ЭВМ диалоговых систем.

ГЛАВА 13

ВАРИАНТ ДИАЛОГОВОЙ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ КОНТРОЛЯ ИСПОЛНИТЕЛЬСКОЙ ДЕЯТЕЛЬНОСТИ

Одной из существенных задач руководителя любого промышленного предприятия, НИИ или КБ является обеспечение четкой исполнительской дисциплины. От успешного решения этой

задачи часто зависит своевременность выполнения плановых заданий и обязательств перед заказчиками.

Постоянный контроль исполнения поручений руководства является одной из важнейших функций автоматизированной системы управления деятельностью предприятия. При этом экономия трудовых затрат достигается путем внедрения средств ВТ, на которые перекладываются трудоемкие операции хранения больших объемов исполнительской информации; наблюдение за процессом исполнения поручений; периодическая выдача извещений работникам предприятия о тех поручениях, по которым они должны предоставлять отчеты руководству; информирование руководства предприятия и сотрудников аппарата контроля о тех директивах, которые следует проконтролировать; анализ выполнения директивных документов за различные периоды времени (месяц, квартал, год, пятилетка и т.д.); осуществление периодического сопоставления процесса выполнения с программой, заданной в директиве, для выявления «узких» мест.

Внедрение автоматизированной обработки исполнительской информации способствует унификации форм документов и технологических этапов их обработки. Это позволяет, в определенной степени, решить задачу взаимозаменяемости административных работников, которая была затруднена тем, что каждый сотрудник имеет свой, присущий ему, стиль работы с директивными документами. Известны случаи, когда замена одного сотрудника административно-управленческого персонала (АУП) другим приводила к выпадению из-под контроля отдельных поручений.

Известны два основных подхода к решению задачи, какие поручения следует записывать в информационную базу автоматизированной системы контроля исполнительской деятельности. В первом случае осуществляется запись на магнитный носитель данных обо всех директивах, поступающих в организацию. Во втором случае на автоматизированный контроль берутся только те поручения, которые не выполнены в срок и по которым руководством было принято решение о переносе срока.

Одной из практических форм координации взаимодействий между различными научными, производственными и управленческими организациями является работа с организационно-распорядительными документами (ОРД), директивными документами вышестоящих организаций, приказами и указаниями руководства НИИ и КБ, письмами и телеграммами, протоколами совещаний, планами мероприятий и др. В процессе деятельности научного учреждения в течение года ведется работа с десятками тысяч ОРД, а в наиболее крупных НИИ и КБ их численность составляет порядка 100 тысяч. В современной организованной системе управления научным учреждением по каждому ОРД определяется порядок и срок его исполнения, назначается ответственный исполнитель. Осуществляется это с помощью специальных резолюций руководства или на основе стандартных требований, установленных соответствующими программно-техническими документами. Обязательным элементом системы управления деятельностью НИИ является контроль своевременности и правильности исполнения ОРД, т. е. контроль исполнительской дисциплины.

Определенный интерес вызывает опыт, полученный в процессе создания и внедрения системы КОНТРОЛЬ — автоматизированной системы контроля исполнительской дисциплины. Система КОНТРОЛЬ построена на основе пакета ТЕРМЕС с учетом принципов безбумажного информирования. Основной задачей данной системы является обеспечение оперативного контроля своевременного исполнения поручений руководства предприятия на основе использования ЕС ЭВМ и дисплейных станций типа ЕС-7920. В число важнейших функций системы входят постановка поручений на контроль по срокам исполнения и составу исполнителей; выдача справок о состоянии исполнения директивных документов; формирование сводных ведомостей руководству предприятия по состоянию исполнения поручений; выдача уведомлений исполнителям о сроках исполнения поручений; внесение изменений в состав исполнителей и ранее установленные сроки исполнения; выдача уведомлений о нарушении сроков исполнения поручений.

Система КОНТРОЛЬ обеспечивает оперативный контроль своевременного исполнения директивных документов. Директивным документом называется документ, содержащий, в общем случае, несколько поручений исполнителям и поступивший в соответствующую службу

предприятия для принятия на контроль. Этот документ регистрируется в журнале и обрабатывается группой контроля. На этом этапе для обеспечения надлежащего контроля поступивший ДД может быть преобразован в несколько ДД по поручениям. Последние передаются в группу эксплуатации для ввода в информационную базу системы КОНТРОЛЬ.

Система КОНТРОЛЬ представляет собой человеко-машинную информационную управляющую систему, состоящую из следующих четырех основных частей: технического обеспечения, информационного обеспечения, организационного обеспечения и программного обеспечения. Рассмотрим каждую из этих компонент системы.

Система КОНТРОЛЬ эксплуатируется на ЭВМ серии ЕС. Одной из технических основ безбумажной информатики является оснащение рабочих мест руководителей и специалистов средствами регистрации и отображения информации, непосредственно соединенными с ЭВМ. На рабочем месте директора института и в службе контроля устанавливаются алфавитно-цифровые дисплейные комплексы типа ЕС-7920. Использование видеотерминалов позволяет в режиме диалога «Человек — ЭВМ» формировать запросы и получать на экране дисплея справки по информации, хранящейся в памяти машины. Использование руководством и службой контроля видеотерминалов практически не мешает, в то же самое время, осуществлять на ЭВМ решение других задач. Кроме видеотерминалов, в комплект внешних устройств входят: МД типа ЕС-5061, необходимые для работы операционной системы и банка контролируемых документов; МЛ типа ЕС-5010 для размещения копий банка контролируемых документов; АЦПУ типа ЕС-7032, необходимое для печати уведомлений и сводок.

Информационное обеспечение системы составляет массив директивных документов (ДД), находящихся на контроле руководства предприятия. Каждая запись этого массива имеет фиксированную длину и содержит один директивный документ. В целях обеспечения сохранности информационной базы осуществляется ее периодическое копирование на МЛ в соответствии с графиком эксплуатации системы. Предусмотрен режим дисплейной корректировки массива документов при помощи дисплея ЕС-7927 в случае изменения сроков исполнения и состава исполнителей, угрозе срыва исполнения поручения и т. д. При этом обеспечивается фиксация документов, регламентирующих указанные изменения. Для хранения переменной информации о документах также организован специальный массив дополнений, служащий для хранения новых сроков исполнения и нового состава исполнителей. Записи массива дополнений, принадлежащие одному документу, связаны между собой при помощи специальных указателей. Массивы документов и дополнений организованы в виде последовательных файлов. Одна логическая запись основного набора данных системы КОНТРОЛЬ содержит сведения об одном ДД. Записи могут быть объединены в блоки, которые должны быть одинаковой длины. Последний блок может быть укороченным. Эти требования соответствуют формату FB или FBS, принятому в ОС ЕС. Для одного из случаев практического использования системы КОНТРОЛЬ в научно-исследовательском учреждении структура логической записи массива ДД состояла из следующих тридцати полей.

Поле 1 — пятизначный номер ДД. Он присваивается сотрудником, ответственным за работу системы КОНТРОЛЬ. Первые три цифры определяют порядковый номер по журналу учета и контроля ДД, две последние — порядковый номер поручения. Нулевое значение этого поля указывает на то, что запись не содержит никаких сведений. Поле 2 — код тематики, к которой относится поручение. Заполняется в случае необходимости контроля тематического плана НИР. Он устанавливается в соответствии с принятой на предприятии системой индексации. Поле 3 — код куратора, ответственного за контроль. Поле 4 — код вида ДД (приказы министра, информационные и директивные письма, приказы руководящих организаций, приказы руководителя предприятия, планы-графики работ предприятия, письма сторонних организаций, прочие). Поле 5 — код подразделения; присваивается группой контроля. Поля 6 и 7 — количество дней, прошедших с момента последнего уведомления исполнителя о поручении и дата последнего уведомления. Поля 8, 9, 10, 11, 12, 13 соответственно даты издания ДД, поступления ДД на предприятие, выдачи резолюции руководителям, исполнения поручения по ДД, фактического исполнения поручения, исполнения поручения (назначенная внутри предприятия). Поле 14 — символьное обозначение (типа "aa-бб-вв") пункта ДД, к которому относится поручение, aa — номер пункта; бб — номер подпункта; вв — номер пункта в

подпункте. Поле 15 — символьное обозначение пункта приложения ДД (соответствует обозначениям поля 14). Поля 16 и 17 соответствуют номеру приложения ДД и номеру ДД. Поле 18 содержит символы, характеризующие ход выполнения ДД. Присваивается группой контроля. Поле 19 — краткое описание причин срыва выполнения поручения. Поле 20 — название подразделения-исполнителя. Поля 21 и 22 содержат фамилии куратора и исполнителей. Поле 23 — краткое описание поручения. Поле 24 — название выходного документа. Поле 25 является расшифровкой поля 14. Поле 26 — номер ДД, присвоенный канцелярией. Поле 27 — исходящий номер документа, содержащего результат выполнения поручения. Поле 28 — расшифровка поля 2. Поле 29 — расшифровка поля 4. Поле 30 — гриф секретности ДД.

В режиме диалога, пользователям системы КОНТРОЛЬ на экраны дисплеев выдаются справки, включающие: перечень работ, неисполненных на данный момент по каждому из подразделений и по всему предприятию в целом; состояние исполнения ДД, находящихся на контроле; перечень ДД, находящихся под угрозой срыва по всему предприятию в целом и по каждому конкретному подразделению предприятия; текущее состояние работ по исполнению каждого ДД и т. д. Основы выдаваемых пользователем видеogramм составляют колонки «Наименование документа», «Отклонения», «Исполнители», «Учетный номер».

Программное обеспечение системы КОНТРОЛЬ функционирует под управлением ОС ЕС версии не ниже 6.1, включающей методы доступа VTAM и BSAM в режимах MVT или SVS. Система КОНТРОЛЬ может выполняться как один из программных комплексов, функционирующих под управлением системы ОБЬ. Базовым пакетом, на основе которого построено программное обеспечение системы КОНТРОЛЬ, является комплекс ТЕРМЕС.

Для обеспечения функционирования системы КОНТРОЛЬ из персонала подразделения, осуществляющего контроль исполнения поручений, назначается сотрудник для координации связанной с системой КОНТРОЛЬ деятельности руководства и исполнителей. Сотрудник, ответственный за контроль, получив подлежащие контролю документы, вводит в ЭВМ сведения о них непосредственно с дисплея в ЭВМ. При этом на документах делается пометка о взятии работ на машинный контроль с датой и подписью ответственного. Для ввода одного поручения в среднем требуется 2—2,5 мин рабочего времени сотрудника, ответственного за сопровождение системы. Периодическая проверка состава документов, находящихся на машинном контроле, проводится на основе соответствующих регулярно выдаваемых распечаток.

В системе КОНТРОЛЬ предусмотрен очень простой и удобный язык ведения диалога. Информационной единицей, выдаваемой на экран видеотерминала, является кадр, содержащий 24 строки (длина строки 80 знаков). В нижней части каждого кадра помещаются указания о том, какие клавиши на клавиатуре видеотерминала необходимо нажать для получения требуемой информации. Большинство операций осуществляется путем нажатия одной из зарезервированных функциональных клавиш. Сеанс диалога с ЭВМ начинается выдачей на экран перечня видов документов, находящихся на контроле. Руководитель просматривает состояние выполнения по какому-то определенному виду или по всем контролируемым документам. После чего, однократным нажатием клавиши осуществляется переход к следующему кадру, где руководителю предлагается указать подразделение, работы которого будут просматриваться, и временной интервал просмотра (при ретроспективных или перспективных просмотрах). Возможна также выдача на экран дисплея информации по запросу о состоянии выполнения конкретного указанного документа. В соответствии с полученными указаниями на ЭВМ осуществляется автоматическая выборка необходимой информации из банка контролируемых документов и формирование рабочего массива для выдачи сведений на экран.

Системой предусмотрен еженедельный автоматизированный просмотр на ЭВМ созданного в памяти машины банка контролируемых документов с последующей выдачей уведомлений подразделениями сводной ведомости. В каждое из подразделений предприятия направляются уведомления о поручениях, подлежащих исполнению в ближайшие 10 дней (срок можно при желании менять), а также невыполненных вовремя. Сводная ведомость, суммирующая содержание всех уведомлений, формируется, в первую очередь, для сотрудников службы контроля исполнения. Подразделения, получив уведомления, в течение суток возвращают их

ответственному за контроль с оформленными извещениями о состоянии выполнения поручений. Предусмотрены следующие типовые формулировки извещений о ходе выполнения работ: выполнена (с указанием даты фактического исполнения, исходящего номера ответственного документа, причин выполнения задания с опозданием, если оно имело место); выполняется в срок; под угрозой срыва; не выполнена в срок (с указанием причин и применяемых мер); прекращена (с указанием документа, которым отменено поручение); срок исполнения перенесен (с указанием нового срока исполнения и документа, на основании которого перенесен срок); передана другому исполнителю (с указанием подразделения, нового исполнителя и документа, санкционировавшего передачу). На основании этих сообщений в ЭВМ вводится информация о состоянии выполнения контролируемых документов. Таким образом, в недельном цикле обеспечивается ведение на ЭВМ банка контролируемых документов, на основе которого в любое время руководитель предприятия, используя режим диалога «Человек — ЭВМ» может оперативно получить требуемые сведения о состоянии выполнения работ.

Для формирования рабочего массива и выдачи на экран первого служебного дисплейного кадра обычно требуется на более 10—15 с, а для ожидания при вызове на экран требуемых информационных кадров — 1 — 2 с. При просмотре массива документов осуществляется переход к следующему информационному кадру, возврат к предыдущему или обращение к началу массива. Так как каждый сотрудник в организационной структуре предприятия занимается определенным кругом вопросов, целесообразно обеспечить такой режим работы с системой, чтобы доступ исполнителю был возможен только к информации, соответствующей роду выполняемых им функций. При этом руководителю предприятия и определенному кругу лиц должен быть обеспечен доступ к любым документам, хранящимся в информационной базе.

Эффективной формой использования системы КОНТРОЛЬ руководителями предприятий показали себя оперативные производственные совещания. На таких совещаниях, при участии начальников отдельных подразделений, анализируется ход выполнения контролируемых работ в процессе просмотра на экране дисплея массива документов, срок выполнения которых нарушен. Руководитель заслушивает объяснения исполнителей, сообщения об осуществленных мерах, принимает соответствующие управленческие решения. В процессе рассмотрения фактически отрабатываются типовые алгоритмы действий, которые должны предприниматься для устранения отставаний и обеспечения своевременности исполнения работ. Наличие автоматизированной системы контроля исполнительской деятельности сокращает трудозатраты сотрудников группы контроля, связанные с постоянным напоминанием о выполнении поручений подразделениям предприятия. Следствием этого является увеличение времени на аналитическую работу — анализ состояния дел по подразделениям и тематикам и проработку вопросов на перспективу. Оперирова с текущим и архивным массивами директивных документов, автоматизированная система контроля исполнительской деятельности может осуществлять анализ выполнения поручений подразделениями, отдельными исполнителями, а также предприятием в целом практически за любой период времени.

Наряду с оперативным информационным обслуживанием руководства и службы контроля система КОНТРОЛЬ позволяет реализовать ряд других функций. Так, перед началом планового периода (месяца, квартала, года) с ЭВМ могут быть выданы для всех подразделений перечни контролируемых работ на плановый период. Сведения о результатах исполнения документов, хранящихся в памяти ЭВМ, используются при подведении итогов деятельности подразделений за рассматриваемый период. В системе предусмотрена выдача «твердых» копий путем распечатки дисплейных форм на АЦПУ или устройстве типа ЕС-7934. В одном из вариантов использования, в системе КОНТРОЛЬ обслуживается три дисплея комплекса ЕС-7920. Один из них информирует руководство предприятия при проведении диспетчерских совещаний (запросам, поступающим с этого терминала, присваивается наивысший приоритет), другие два — сотрудников группы эксплуатации системы и исполнителей поручений руководства предприятия.

Внедрение системы КОНТРОЛЬ на нескольких десятках промышленных предприятий и научных учреждений позволило автоматизировать на них ряд трудоемких этапов проверки

исполнения директивных документов, подтвердило высокую степень надежности ее программного обеспечения. В целом эта система показала себя как удобный и эффективный инструмент управления. В перспективе рабочее место каждого руководителя должно быть оснащено дисплейным комплексом или персональным компьютером. Поэтому система КОНТРОЛЬ представляет интерес не только как средство решения задач сегодняшнего дня, но и как опыт отработки методов безбумажной информатики для последующих приложений.

ЗАКЛЮЧЕНИЕ

В заключение хотелось бы сделать некоторые предположения о перспективах развития программного обеспечения систем телеобработки данных.

Можно уже считать свершившимся фактом то, что ЕС ЭВМ потеряли ведущую роль в отечественном парке вычислительной техники. Им на смену приходят персонально-профессиональные ЭВМ, выполненные на микропроцессорах, архитектура которых мало напоминает ЕС.

Разработка программного обеспечения систем телеобработки данных для этих ЭВМ намного проще, чем для ЕС ЭВМ. Многие вопросы решены на уровне аппаратного и системного программного обеспечения.

Казалось, что время систем телеобработки данных ЕС ЭВМ уходит, и они теперь имеют только познавательный интерес. Но это не так. В конце 80-х годов фирма IBM выпустила новое семейство компьютеров IBM Enterprise System/9370. Его архитектура базируется на архитектуре System/370, и обеспечивается работа программного обеспечения IBM/370. Машины выполнены по новой технологии с учетом последних достижений микроэлектроники. Назначение ЭВМ этой серии аналогично IBM/370.

Вероятно, можно утверждать, что цель, которую преследовала фирма этой разработкой, состояла в том, чтобы объединить архитектуру System/370 и особенно богатство ее программного обеспечения с микроэлектронной технологией.

Зная авторитет и влияние фирмы IBM на мировом компьютерном рынке, можно в недалеком будущем ожидать появление этих машин или их аналогов у нас. Тогда вновь появится интерес к программному обеспечению ЕС ЭВМ.

В области программного обеспечения систем телеобработки данных возникнут трудности, так как этот компонент вычислительного комплекса зависит от периферийного окружения и аппаратуры передачи данных, на которую автоматический перенос зарубежного программного обеспечения, как правило, не приводит к успеху. В связи с этим материал данной книги может оказаться полезным не только пользователям ЭВМ Единой Системы, но и разработчикам программного обеспечения.

СПИСОК ЛИТЕРАТУРЫ

- 1.Блэкман М. Проектирование систем реального времени.— М.: Мир, 1977.— С. 29—314.
- 2.Дьяконов В. Ю., Калинин И. А., Китов В. А. Принцип мобильности программного обеспечения мультитерминальных систем распределенной обработки данных // Программирование.— 1984.— № 2.— С. 46-53.
- 3.Дьяконов В. Ю., Калинин И. А., Китов В. А. Мультитерминальная система распределенной обработки данных ОБЪ // ВТ соц. стран.— М.: Финансы и статистика, 1987.— Вып. 22.— С. 131 — 136.
- 4.Мартин Дж. Программирование для вычислительных систем реального времени.— М.: — Наука, 1975.
- 5.Мартин Дж. Системный анализ передачи данных.— М.: Мир, 1975.
- 6.Позин И. Л., Щербо В. К. Телеобработка данных в автоматизированных системах.— М.: Статистика, 1974.
- 7.Сипсер З. Архитектура связи в распределенных системах.— М.: Мир, 1981.— Кн. Г.— С. 16—429, кн. 2.— С. 446—500, 557—661.
- 8.Уилко М. Системы с разделением времени.— М.: Мир, 1972.— С. 9—121.
- 9.Хромов В. И., Ульянов С. А. Введение в программирование для систем телеобработки данных.— М.: финансы и статистика, 1982.— С. 65-164.

Научное издание

ДЬЯКОНОВ Владимир Юрьевич
КАЛИНЧЕВ Игорь Алексеевич
КИТОВ Владимир Анатольевич

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ СИСТЕМ ТЕЛЕОБРАБОТКИ ДАННЫХ

Серия «Библиотечка программиста», выпуск 65

Заведующий редакцией *Е. Ю. Ходам*
Редактор *Т. И. Пташник*
Художник *А. М. Пономарева*
Художественный редактор *Г. И. Коровина*
Технический редактор *И. Ш. Аксельрод*
Корректоры *Л. И. Назарова, Н. Д. Дорохова*

ИБ№ 41064

Сдано в набор 26.04.91. Подписано к печати 17.01.92. Формат 84×108/32. Бумага тип. № 2 Гарнитура таймс. Печать офсетная. Усл. печ. л. 8,4. Усл. кр.-отг. 8,61. Уч.-изд. л. 10,99. Тираж 6530 экз. Заказ № 797.

Ордена Трудового Красного Знамени издательство «Наука»
Главная редакция физико-математической литературы
117071 Москва В-71, Ленинский проспект, 15

Четвертая типография издательства «Наука»
630077 г. Новосибирск, 77, Станиславского, 25