ЭЛЕКТРОННЫЕ

6 Средства общения с ЭВМ

ВЫЧИСЛИТЕЛЬНЫЕ

ДИАЛОГ

13 МАШИНЫ

ЭВМ

ЭЛЕКТРОННЫЕ ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ

В **восьми** книгах

Под редакцией
д-ра техн. наук проф. А.Я. Савельева

А.В. Савченко, Ю.В. Сальников
А.Н. Филиппов

Средства общения с ЭВМ

КНИГА **6**

Издание второе, переработанное и дополненное

Москва «Высшая школа» 1991

ББК 32.973
Э45
УДК 681.3

Рецензент—кафедра вычислительной техники Ленинградского электротехнического института им. В. И. Ульянова-Ленина

Э 45 Электронные вычислительные машины: В 8-ми кн. Кн.6. Средства общения с ЭВМ: Практ. пособие для вузов / А.В.Савченко, Ю.В.Сальников, А.Н.Филиппов; Под ред. А.Я.Савельева. — 2-е изд., перераб. и доп. — М.: Высш. шк., 1991. — 127 с.: ил.

ISBN 5-06-001771-0

Шестая книга серии "ЭВМ" посвящена описанию состава и структуры средств общения с ЭВМ, а также этапов жизненного цикла диалоговой системы. Рассмотрены основные понятия и правила работы с диалоговыми системами, особенности диалогового окружения, специфика диалога с ЭВМ в основных сферах автоматизированной деятельности человека.

Второе издание (1-е —1987г.) дополнено новыми материалами.

$Э\frac{2405000000-419}{001(01)-91}191-91$

ББК 32.973
6Ф7

ISBN 5-06-001771-0

ВВЕДЕНИЕ

Научно-технический прогресс на современном этапе характеризуется широким внедрением вычислительной техники во все сферы деятельности человека. ЭВМ, являясь катализатором научно-технического прогресса, сами подвержены этому прогрессу. Успехи микроэлектронной техники, наряду с разработкой сверхмощных вычислительных машин и комплексов, позволили построить компактные и сравнительно дешевые микроЭВМ, габаритные размеры которых позволяют свободно размещать их на рабочем месте. Таким образом, ЭВМ постепенно становится рабочим инструментом и советчиком инженера и ученого, врача и журналиста, геолога и архитектора, учителя и студента. И сейчас, как никогда, важно быть готовым к грамотному использованию ЭВМ, к общению с ЭВМ. Недалеко то время, когда мы будем пользоваться ЭВМ так же легко и свободно, как телефоном или телевизором.

Под средствами общения с ЭВМ следует понимать совокупность аппаратных средств и соответствующего программного обеспечения для ввода, обработки и вывода информации, ориентированной на ведение диалога в системе "человек—ЭВМ".

Глава 1 посвящена аппаратным средствам общения с ЭВМ. Здесь предполагается краткое знакомство с дисплеем большой ЭВМ и персональным компьютером, а также с другими периферийными устройствами, обеспечивающими обмен информацией с ЭВМ.

В главе 2 представлено начало диалога с ЭВМ. Следует иметь в виду, что термин "диалог" здесь понимается условно, так как ЭВМ не участвует в диалоге, а лишь поддерживает запрограммированные сценарии возможного течения диалога, какими бы сложными они ни были. В главе приведены типы и общие правила диалога.

Глава 3 посвящена наиболее распространенным применениям ЭВМ в диалоговом режиме. Наряду с описанием таких применений в микроЭВМ, показаны соответствующие режимы в системе коллективного пользования "Фокус", разработанной для ЕС ЭВМ. В конце главы приведены примеры практической работы на ЭВМ с использованием рассмотренных режимов.

Освоив стандартные применения ЭВМ, многие специалисты приходят к мысли применить ЭВМ для решения профессиональных задач. И здесь наиболее привлекательным, с точки зрения пользователя, является диалоговый режим. Описание некоторых программных средств диалога и интенсивно развивающейся в настоящее время машинной графики можно найти в главе 4. Здесь же приводится краткий обзор сфер применения диалога с ЭВМ.

Глава 5 посвящена проблемам, связанным с разработкой и сопровождением программ на всех этапах жизненного цикла диалоговых средств.

В главе 6 рассмотрены перспективные направления развития средств общения с ЭВМ.

В данной книге не ставилась задача энциклопедического охвата всех аппаратных и программных средств общения с ЭВМ. Задача состояла в том, чтобы дать обучаемому начальный объем знаний в данной области. Полностью понять и освоить все тонкости общения с машиной можно только в непосредственном и регулярном контакте с ЭВМ.

Глава 1
ТЕХНИЧЕСКИЕ СРЕДСТВА ДИАЛОГА

1.1. Дисплей

Вычислительная техника проникает во многие нетрадиционные сферы применения. Несмотря на это, известно только два основных способа ее использования: а в т о м а т и ч е с к и й, когда вычислительная машина непосредственно управляет работой машин и механизмов; а в т о м а т и з и р о в а н н ы й, когда ЭВМ взаимодействует с предметами труда через человека. Эффективность использования ЭВМ во втором способе зависит как от уровня подготовки пользователя, так и от качества программных и аппаратных средств ведения диалога. Основным аппаратным средством общения с ЭВМ являются различного рода дисплеи.

Использование дисплея в различных сферах деятельности человека становится обычным явлением. Уже сегодня можно видеть дисплей в вузовской аудитории и кабинете врача, на столе руководителя и рабочем месте мастера цеха, в редакции центральной газеты и кабинете информатики сельской школы. Завтра он появится в квартире, домоуправлении, детском саду. Поэтому знание дисплея и умение обращаться с ним необходимы каждому — от научного работника до квалифицированного рабочего, от руководителя предприятия до воспитателя детского сада.

В технической литературе дисплеи иногда называют *экранными пультами*, а также *диалоговыми терминалами* или просто *терминалами*, т.е. оконными устройствами вычислительной техники.

Основным назначением дисплеев является ввод информации (данных, программ, директив) в ЭВМ и ее отображение из памяти вычислительной машины. Двойное

использование дисплеев — в качестве устройств ввода и вывода информации — позволяет применять их при общении с ЭВМ.

Разработанные в настоящее время дисплеи отличаются большим разнообразием форм и размеров, составом конструктивных элементов. Обычно они выполняются в виде отдельных функционально законченных устройств, однако иногда встраиваются в аппаратуру. Отличие конструкций обусловлено различным назначением дисплеев и разными эргономическими требованиями, предъявляемыми к ним в конкретных применениях.

Функционально в дисплее можно выделить следующие основные компоненты: *экран, пульт управления, фотоселектор* и др.

Экран дисплея внешне похож на обычный телевизионный приемник. (Иногда в качестве дисплея используется и телевизор.) Как правило, дисплей конструируется на базе электронно-лучевой трубки, аналогичной кинескопу телевизора. Электронно-лучевые трубки различаются размерами, видом отображаемой информации, методами и скоростью генерации знаков и другими особенностями. В частности, дрожание изображения на экране, характерное для телевизора, в дисплеях устраняется за счет более высокой частоты обновления кадров (обычно 50 кадр/с по сравнению с 24 кадр/с у телевизора).

Различают *алфавитно-цифровые* и *графические* дисплеи.

На экране алфавитно-цифрового дисплея воспроизводятся буквы, цифры и некоторые дополнительные специальные символы. Чтобы уменьшить объем буферной памяти, в которой хранится информация для экрана дисплея, в ней запоминаются не сами символы, а только их коды. Изображение символа генерируется по его коду специальным электронным устройством — генератором символов.

Графический дисплей устроен аналогично алфавитно-цифровому, но в отличие от него может высвечивать на экране любые графики и рисунки, иногда в полутоновом и цветном изображении. Поэтому графические дисплеи сложнее и в несколько раз дороже алфавитно-цифровых. Чтобы получить высокое качество рисунков на графических дисплеях, они проектируются с разрешающей

способностью, как правило, выше, чем у алфавитно-цифровых дисплеев. Кроме генераторов символов в графический дисплей встраиваются генераторы векторов и других линий (дуг окружностей, кривых, заданных уравнениями, и т.п.) разной длины и направлений. Используя графические функции, заложенные в программное обеспечение вычислительной машины, можно проводить разнообразные операции над графическими фигурами: повороты, перемещения, изменения масштабов и др.

Введение цвета повышает изобразительные возможности как графического, так и алфавитно-цифрового дисплея. Так, использование цвета в алфавитно-цифровом дисплее повышает информационную емкость экрана на 45%.

При одновременной работе с несколькими программами или автоматизированном контроле с дисплея за несколькими объектами может появиться необходимость в нескольких информационных "окнах" на одном экране дисплея. В случае аппаратной реализации такой "многооконный" дисплей называется полиэкранным дисплеем. Однако и обычный дисплей может использоваться как полиэкранный при соответствующей реализации программного обеспечения.

Кроме электронно-лучевых бывают и так называемые плазменные дисплеи. Они строятся по принципу тлеющего разряда, возникающего в разреженном газе между двумя перекрещенными проводниками, к которым приложено электрическое напряжение. Преимущество плазменного дисплея — возможность хранения регенерируемой на экране информации не в буферной памяти, а непосредственно на экране (в местах пересечения проводников), а также возможность построения плоского экрана дисплея, имеющего меньшие габаритные размеры по сравнению с обычным дисплеем.

Пульт управления дисплеем включает в себя индикаторы, ручки регулировки и клавиатуру. С помощью пульта управления пользователь может управлять ходом вычислительного процесса на ЭВМ, вводить и корректировать (редактировать) информацию на экране дисплея, записывать ее в память машины и считывать оттуда. Пульт управления обладает функционально избыточным набором органов управления работой ЭВМ.

Пользователь имеет возможность выбора наиболее удобных органов и приемов общения с ЭВМ.

Индикаторы информируют пользователя о готовности дисплея к работе и о режиме, в котором предстоит работать.

Ручки регулировки сходны с телевизионными и предназначены для установления требуемой контрастности, яркости, подстройки фокусировки изображения текстов и другой информации на экране дисплея.

В состав пульта управления дисплеем дополнительно могут входить малогабаритные светочувствительные передатчики и приемники: световое перо и фотоселектор. Световое перо используется для ввода на экран произвольной графической информации, а фотоселектор — для указания или выбора с экрана одной из предлагаемых на дисплее альтернатив — программно-реализованных функций управления дисплеем и ЭВМ.

Дисплей ЕС-7927 является алфавитно-цифровым устройством отображения электронно-вычислительных машин единой системы (ЕС ЭВМ) — комплекса программно и аппаратно совместимых машин. Этот тип дисплеев наиболее распространен. Вместе с присоединенной к нему клавиатурой терминал позволяет обмениваться информацией с вычислительной машиной, изменять или стирать данные на экране, а также возвращать откорректированные данные обратно в ЭВМ для хранения или последующей обработки. Дополнительно дисплей может комплектоваться фотоселектором. Расстояние от дисплея до вычислительной машины не должно превышать 1200 м. Данный тип дисплея может использоваться в качестве пульта вычислительной машины, абонентского пульта (удаленного терминала) или устройства подготовки данных.

Алфавитно-цифровые символы на экране электронно-лучевой трубки дисплея изображаются в матрице размера 9X7 точек. Всего на экран может быть занесено 24 строки по 80 символов в каждой.

На экране можно задавать поля (позиции групп подряд стоящих символов) различного типа, отличающиеся правилами работы с ними: защищенные от вмешательства оператора и незащищенные, алфавитно-цифровые и цифровые, выбираемые фотоселектором и невыбираемые, отображаемые с повышенной или нормальной яркостью и неотображаемые. В зависимости от особенностей работы системы на дисплее можно использовать один из перечисленных типов полей или любой их набор. Тип полей экрана задается и изменяется программно.

Специальный символ, называемый курсором, указывает позицию на экране дисплея, в которой появится очередной символ. При включении устройства курсор автоматически помещается на экране в крайней верхней позиции. Положение курсора можно изменить с помощью клавиатуры или программно. Курсор на экране отображается всегда, даже в том случае, если он помещен в неотображаемом поле.

Дисплей имеет звуковую сигнализацию, которая включается при перемещении курсора из последней позиции экрана в первую или по сигналу из ЭВМ.

Дисплей с помощью клавиатуры выполняет следующие основные действия:

— запись и изменение символов;
— стирание группы символов;
— стирание всех символов (кроме курсора);
— сдвиг текста строки влево с вычеркиванием одного символа;
— сдвиг текста строки вправо с вставкой одного символа;
— пошаговое, непрерывное или скачкообразное перемещение курсора;
— запрос со стороны ЭВМ на обслуживание.

На рис. 1.1 схематично изображен пульт управления дисплеем, состоящий из панели с индикацией и ручками регулировки и клавиатуры. При включении питания на панели дисплея загорается индикатор СЕТЬ. Чтобы включить дисплей в работу системы, надо поставить специальный ключ в положение ОТКЛ., повернуть его в положение ВКЛ. и подождать, пока загорятся индикаторы ВКЛ. и ВВОД ЗАПРЕЩЕН. После этого ключ поворачивается в положение РАБОТА, индикатор ВВОД ЗАПРЕЩЕН должен погаснуть, а индикатор СИСТ. ДОСТУПНА загореться. В верхнем левом углу экрана появляется курсор, и система становится доступной для работы оператора.

После загрузки оператором ЭВМ операционной системы в вычислительную машину пользователь может приступить к работе. Для регулирования контрастности и яркости изображения символов на экране служат ручки КОНТРАСТ. и ЯРКОСТЬ. В случае выхода дисплея из строя загорается индикатор АВАР. При установке курсора в защищенное поле загорается индикатор ВВОД ЗАПРЕЩЕН и передачи текста в ЭВМ не происходит.

Клавиатура служит для управления работой дисплея и функционально состоит из 8 клавиш управления курсором (все клавиши со стрелкой), 5 регистровых (ФВР, ВР, НР, СБ, ФСБ), 49 алфавитно-цифровых и специальных знаков (4 ряда по 12 белых клавиш и длинная нижняя пробельная клавиша в центральной части клавиатуры), 14 клавиш программного функционирования (ПФ1-ПФ12, ВВОД, ВЫЗОВ ТЕСТА), 8 функциональных (СТРН ЭКР — стирание экрана, СТРН ВВ — стирание ввода, СТРН КП — стирание до конца поля, ДУБЛ — дублирование, МТКП — метка конца поля, ВСТВ — вставка, ВЧРК — вычеркивание, СБРОС). Из 84 клавиш при фиксации нажатия 9 дублируют данный символ слева направо и последовательно от верхней строки к нижней. Эти клавиши на рис. 1.1. заштрихованы.

Клавиши управления курсором устанавливают курсор в ту позицию экрана, в которую в дальнейшем можно заносить алфавитно-цифровой или специальный знак с клавиатуры. Четыре клавиши управления курсором расположены в правой части клавиатуры.

Клавиши управления курсором делятся на две группы — перемещения курсора по позициям и по полям экрана.

При нажатии клавиши управления курсором содержимое экрана не меняется. Однократное нажатие клавиши (ВВЕРХ) перемещает курсор в той же колонке вверх на одну строку. Если курсор был в первой строке экрана, то он переместится в последнюю строку. Действие клавиши (ВНИЗ) аналогично действию предыдущей клавиши, только курсор перемещается сверху вниз и из последней строки в первую строку экрана. Однократное нажатие клавиши (ВПРАВО) перемещает курсор вправо на одну позицию той же строки. Если курсор располагается в конце строки, то он переместится в начало следующей строки, если располагается в конце последней строки, то переместится в начало первой строки. Действие клавиши (ВЛЕВО) аналогично действию предыдущей клавиши, при этом курсор перемещается в противоположном направлении. В правом верхнем углу центральной части клавиатуры расположена клавиша (ВОЗВРАТ), маркировка и функции которой совпадают с клавишей ВЛЕВО. Наличие в дисплее второй клавиши с одинаковыми

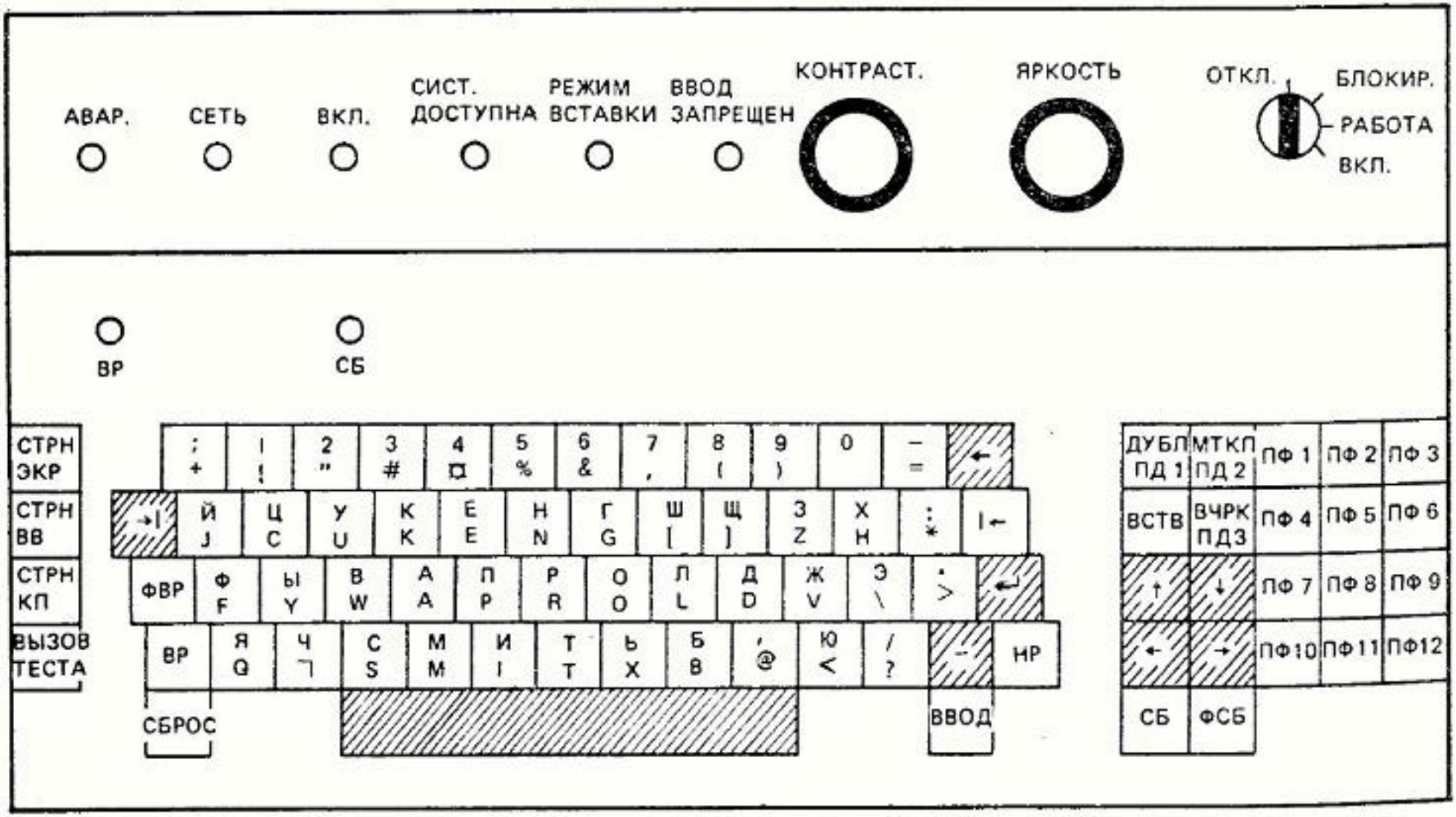
АВАР.
СЕТЬ
ВКЛ.
СИСТ. ДОСТУПНА
РЕЖИМ ВСТАВКИ
ВВОД ЗАПРЕЩЕН
КОНТРАСТ.
ЯРКОСТЬ
ОТКЛ.
БЛОКИР.
РАБОТА
ВКЛ.
ВР
СБ
СТРН ЭКР
СТРН ВВ
СТРН КП
ВЫЗОВ ТЕСТА
СБРОС
ВВОД
НР
ДУБЛ ПД 1
МТКП ПД 2
ПФ 1
ПФ 2
ПФ 3
ВСТВ
ВЧРК ПД3
ПФ 4
ПФ 5
ПФ 6
ПФ 7
ПФ 8
ПФ 9
ПФ10
ПФ11
ПФ12
СБ
ФСБ

Рис. 1.1

функциями связано с почти полным совпадением центральной части клавиатуры пишущей машинки и дисплея.

Клавиши перемещения курсора по полям предоставляют пользователю дополнительные сервисные возможности при работе с форматированным экраном. Клавиша ТАБУЛЯЦИЯ (→|) устанавливает курсор в первую позицию следующего незащищенного поля. Клавиша ОБРАТНАЯ ТАБУЛЯЦИЯ (|←) устанавливает курсор либо в первую позицию незащищенного поля (в случае, если курсор находится в любой другой позиции этого поля), либо в первую позицию предыдущего незащищенного поля. Если экран не содержит незащищенных полей, то курсор при нажатии любой из двух последних клавиш устанавливается в первую позицию экрана. Эту клавишу удобно использовать при вводе и корректировке информации, заданной в табличной форме. Клавиша НОВАЯ СТРОКА (↵) помещает курсор в первую незащищенную позицию следующей строки. Если экран не форматирован, то курсор устанавливается в первую позицию следующей строки.

Регистровые клавиши — это клавиши НР и ВР (НИЖНИЙ И ВЕРХНИЙ РЕГИСТРЫ), которые используются для выбора регистра отображения символов клавиатуры дисплея. При одновременном нажатии регистровой и символьной клавиш на экран занесется один из двух знаков, изображенных на нажатой символьной клавише, в соответствии с выбранным регистром. Для фиксации требуемого положения регистра используется клавиша ФВР (ФИКСАЦИЯ ВЕРХНЕГО РЕГИСТРА). При этом если на пульте горит (или не горит) лампочка ВР, то фиксируется состояние верхнего (или нижнего) регистра. Клавиши СБ (СТРОЧНАЯ БУКВА) и ФСБ (ФИКСАЦИЯ СТРОЧНОЙ БУКВЫ) предназначены для переключения клавиатуры в режим ввода строчных или прописных букв.

Клавиши с алфавитно-цифровыми и специальными символами нажимаются после установки курсора в требуемую позицию и при выборе регистра, соответствующего данному символу.

Клавиши программного функционирования блокируют клавиатуру дисплея при нажатии любой из них, что подтверждает зажженная лампочка ВВОД ЗАПРЕЩЕН. За клавишами ПФ1-ПФ12 программно закрепляются машинные операции, чаще других применяемые пользователем. В разных программных системах за этими клавишами закреплены различные операции. Использование клавиши программного функционирования заметно повышает эффективность работы за дисплеем.

После набора символов на экране они хранятся в памяти дисплея. Для ввода в ЭВМ набранной информации используется клавиша ВВОД.

Клавиша ВЫЗОВ ТЕСТА используется техническим персоналом при диагностике работы дисплея.

Функциональные клавиши используются для полного или частичного стирания текстов, вставки пропущенных символов и их удаления с экрана дисплея и т.д.

Рассмотрим основные действия при исправлении ошибок на экране дисплея.

Для замены ошибочного символа с помощью клавиш управления под него подводится курсор, нажимается клавиша с требуемым символом и ошибка устраняется. После этого курсор можно вернуть в исходное положение.

Для стирания текста с того места, где установлен курсор, до конца экрана надо или один раз нажать функциональную клавишу СТРН КП или многократно нажимать пробельную клавишу. При удалении информации из поля ввода достаточно нажать клавишу СТРН ВВ, а при стирании всего экрана — клавишу СТРН ЭКР. В этих случаях курсор установится вначале экрана.

Чтобы удалить из текста лишний символ, под него подводится курсор и нажимается клавиша ВЧРК. После этого лишний символ исчезнет, а текст, находящийся справа от курсора, сдвинется на одну позицию влево.

При вставке в текст недостающего символа курсор устанавливается в требуемую позицию, нажимается функциональная клавиша ВСТВ, после чего загорается лампочка РЕЖИМ ВСТАВКИ. Далее нажимается клавиша с нужным знаком, текст (вместе с курсором), начиная с позиции, в которой был установлен курсор, сдвигается вправо, а на том месте, где был ранее установлен курсор, появится вставляемый символ.

Клавиша СБРОС нажимается для того, чтобы отменить режим ВВОД ЗАПРЕЩЕН и РЕЖИМ ВСТАВКИ. Если в системе возникла ошибка, то команда сброса не выполняется.

Фотоселектор — малогабаритный фоточувствительный приемник, подсоединенный к пульту управления через электрический кабель. Фотоселектор вставляется в специальный держатель, закрепленный на корпусе дисплея. При выборе пользователем одного из сообщений, отображаемых на экране, фотоселектор подносится к нему и нажимается на поверхность экрана. На конце фотоселектора конструктивно располагается микровыключатель, после сбрасывания которого имитируется работа клавиши ВВОД, и ЭВМ отрабатывает выбранную программно реализованную операцию по управлению дисплеем или ЭВМ.

Фотоселектором можно пользоваться в то время, когда дисплей свободен от других работ с клавиатурой и передачи данных.

Персональный компьютер строится на базе микроЭВМ, содержащей те же блоки, на которых конструируются средние и большие вычислительные машины. Отличительная особенность этих ЭВМ — возможность объединения в большие интегральные схемы (БИС) многих узлов и блоков, наличие минимального числа и незначительной протяженности внешних и внутренних связей, а также наличие "открытой" архитектуры, т.е. простота объединения различных устройств в вычислительную систему. Процессор составляет как бы единое целое с дисплеем персонального компьютера.

Появление микроЭВМ стало возможным благодаря успехам микроэлектроники в разработке новых материалов, компонентов и узлов вычислительной техники. Собственно центральный процессор и оперативная память микроЭВМ занимают объем в несколько кубических сантиметров. Остальной объем аппаратуры приходится на внешние устройства, схемы сопряжения с ними, источники питания, различные конструктивные элементы. По вычислительным возможностям микроЭВМ близки или превзошли ряд ЭВМ второго и третьего поколений. Основные технические данные двух отечественных микроЭВМ и одной ЕС ЭВМ приведены в табл.1.1.

В настоящее время микроЭВМ — наиболее интенсивно развивающийся вид вычислительных машин. Растет разрядность, объем оперативной и внешней памяти, развивается номенклатура внешних устройств микроЭВМ. Дальнейшее развитие получает их программное обеспечение. Расширяются сферы применения микроЭВМ в народном хозяйстве. В зависимости от назначения персональные компьютеры делятся на бытовые, учебные, профессиональные (для автоматизации различных видов деятельности пользователя).

Персональные микроЭВМ обладают многими привлекательными для пользователей характеристиками и возможностями: относительной дешевизной и доступностью, возможностью работы в реальном масштабе времени, монопольным использованием ресурсов ЭВМ, высокой надежностью, компактностью.

Таблица 1.1

Технические данные	ДВК-2М	Искра-226	ЕС-1022
Объем оперативной памяти, Кбайт	52	128	128–512
Быстродействие процессора, (тыс. оп./с.)	300–500	800, наличие аппаратно–реализованных функций	80
Языки програмирования	Макроассемблер, Бейсик, Фортран, Паскаль, графический макроассемблер, графический Паскаль	Машинный код, Бейсик	Ассемблер, Фортран, Паскаль, ПЛ/1, Кобол, РПГ, графический ассемблер, графический Паскаль, Бейсик и другие языки программирования. Совместимость програмного обеспечения от младших моделей к старшим (снизу–вверх)

Технические данные	ДВК-2М	Искра-226	ЕС-1022
Дисплей	Алфавитно-цифровой дисплей (АЦД): 24 строки по 80 символов. Возможность подключения графической приставки к АЦД: массив графической информации 286×400 точек	АЦД: 24 строки по 80 символов, максимальный массив графической информации 256×560 точек	АЦД: а) 24 строки по 80 символов; б) 12 строк по 80 символов. Наличие графической станции
Другие периферийные устройства	НГМД, термопечатающее устройство (ТПУ) печатающие устройства (ПУ) типа ДЗМ-180 и "Роботрон", матричное печатающее устройство	НГМД, НМД, НМЛ, кассетный НМЛ, ПУ типа ДЗМ-180 и "Роботрон", планшетный графопостроитель, набор адаптеров	Различные устройства перфокарточного и перфоленточного ввода-вывода, пишущая машинка, АЦПУ параллельного действия, НМД и НМЛ широкой номенклатуры, графопостроители, другие устройства

Клавиатура персональных микроЭВМ частично совпадает с рассмотренной выше. Центральная часть клавиатуры большинства типов микроЭВМ неспециального назначения совпадает с клавиатурой пишущей машинки. У разных типов машин обычно совпадает назначение клавиш управления курсором и регистровых клавиш. Другие части клавиатуры, как правило, функционально значительно отличаются для разных ЭВМ.

Дополнительные элементы дисплея микроЭВМ, управляющие перемещением курсора по экрану — рычаг, шар, "мышь". Рычаг представляет собой подвижную консоль, закрепленную на клавиатуре дисплея и обладающую двумя степенями свободы, т.е. способностью наклоняться вперед — назад и вправо — влево. С помощью соответствующих наклонов рычага можно перемещать курсор на экране дисплея в любом направлении. Аналогичные действия можно выполнять с помощью вращения шара. Конструктивно шар размещается в небольшой коробке так, что значительная часть его выступает над поверхностью. Перевернув коробку с шаром вниз, можно получить другое устройство —

"мышь", которую можно катать по плоской поверхности, перемещая на экране курсор в требуемом направлении.

Дополнительные элементы управления курсором позволяют получить две входные величины, характеризующие координаты курсора. Часто описанные устройства снабжаются одной или двумя кнопками, дают возможность ввести дополнительное управление, например зафиксировать текущие координаты. При наличии визирных приспособлений с помощью дополнительных элементов управления курсором можно строить графические изображения.

1.2. Устройства ввода-вывода

Устройства ввода-вывода ЭВМ отличаются широкой номенклатурой. Так, только для ЕС ЭВМ промышленность серийно выпускает свыше 130 типов внешних устройств.

Устройства ввода используются для занесения в память ЭВМ требуемой информации: программ, управляющих операторов или директив, исходных данных. Наиболее известными являются устройства перфокарточного и перфоленточного ввода, использующие в качестве носителей информации перфокарты и перфоленты.

Кроме перфокарточного и перфоленточного существуют и другие устройства ввода: устройство ввода с голоса, читающие автоматы, различные кодировщики и др. Внедрены и широко известны, например, автоматы для распознавания почтовых индексов на конвертах.

Устройства вывода еще более разнообразны, чем устройства ввода. Наиболее широка номенклатура алфавитно-цифровых печатающих устройств. Эти устройства в комплекте с дисплеем компенсируют отсутствие "твердой" копии документированного диалога. В качестве устройств вывода могут использоваться различные электронные табло и индикаторы (цифровые и аналоговые). Графическая информация выводится на самописцы, графопостроители, рисующие изображение с помощью одного или нескольких (цветных) чернильных перьев. Некоторые типы мозаичных посимвольно печатающих устройств могут отображать графическую информацию с разрешающей способностью, пригодной для большинства практических применений.

Существуют и другие типы устройств вывода информации, например синтезаторы речи, устройства управления различными проекторами и т.п.

Устройства ввода-вывода могут выполнять функции как ввода, так и вывода информации. Одно такое устройство — дисплей — уже рассмотрено.

Наиболее развитый класс таких устройств — запоминающие устройства на магнитных носителях: лентах, дисках, барабанах и др. Для ряда этих устройств разработаны также автономные устройства подготовки данных.

Вопросы для самопроверки

1.1. Назовите основные функциональные части дисплея.

1.2. Для генерации изображения на экране дисплея ЕС-7927 используется буферная память с произвольным доступом. Какой требуется минимальный объем буферной памяти для дисплея ЕС-7927?

1.3. Найдите алгоритм определения оптимального расположения клавиш на клавиатуре с точки зрения минимизации перемещения пальцев оператора при работе с дисплеем.

1.4. Сколько комбинаций символов можно отобразить в матрице, изображающей символ на экране дисплея ЕС-7927? На сколько это превышает возможности кодировки символа байтом?

1.5. Какова максимально достижимая информационная емкость стандартной 80-колонной перфокарты?

1.6. Какова максимально достижимая информационная емкость 5-дорожечной перфоленты длиной 1м? То же, для 8-дорожечной перфоленты?

Глава 2
НАЧАЛО ДИАЛОГА

2.1. Режимы работы с ЭВМ

Взаимодействие человека с ЭВМ не сразу приняло диалоговые формы, наиболее распространенные в настоящее время. Исторически диалоговому режиму общения с вычислительной машиной предшествовали системы пакетного или закрытого типа.

Пакетный режим характеризуется тем, что задачи готовятся автономно от ЭВМ и собираются в пакет — массив входных заданий. Задания вводятся в машину оператором ЭВМ по одному в порядке очереди в пакете. Последовательность выполнения программ и их отдельных шагов задается системой таким образом, чтобы эффективность использования центральных, наиболее дорогих, устройств ЭВМ была максимальной. При работе в пакетном режиме программист связан с вычислительным процессом косвенно, а промежуток времени между подачей задачи на машину и получением результата составляет обычно часы или даже дни.

В качестве носителей информации при работе в пакетном режиме применяются магнитные ленты или гибкие магнитные диски. При пакетной обработке достигается максимальная эффективность использования ЭВМ. Такой режим обслуживания минимизирует нерациональную загрузку средств вычислительной техники.

При пакетной обработке достигается максимальная эффективность использования ЭВМ. Такой режим обслуживания минимизирует нерациональную загрузку средств вычислительной техники.

Широкое распространение вычислительных машин снизило актуальность максимальной загрузки ЭВМ и

выдвинуло на первый план повышение эффективности работы пользователя, главным образом непрофессионального. С решением этой задачи связано появление и развитие диалогового режима работы на ЭВМ.

Диалоговый режим общения с вычислительной машиной предполагает наличие диалоговой системы, с помощью которой пользователь получает непосредственный доступ к ЭВМ. Как правило, он делит машинные ресурсы (центральный процессор, память, внешние устройства, программное обеспечение ЭВМ) с другими пользователями, численность и состав которых могут меняться более или менее случайным образом. В процессе обработки задачи пользователь имеет возможность непосредственно и оперативно контролировать работу программы, менять решения и разнообразить действия при общении с ЭВМ. Возможности диалоговой системы и удобство работы с ней во многом зависят от гибкости пользования, определяемой числом различных процессоров, которые могут выполняться с помощью системы.

Организация работы с машиной в диалоговом режиме предполагает общение с ней на языках, приближенных к естественному, со скоростью, близкой к темпу диалога с человеком, причем в некоторых случаях на уровне операционной сложности, адаптируемой в процессе общения к опыту и квалификации пользователя. *Операционная сложность* означает сложность действий (операций), которые должен выполнить пользователь для решения поставленной задачи.

Для работы начинающих пользователей достаточно иметь системы с незначительной гибкостью пользования и обязательно с минимально допустимой операционной сложностью.

Сравнение диалогового и пакетного режимов не дает явных преимуществ ни одному из них безотносительно к типу решаемых на ЭВМ задач и квалификации пользователей, однако в ряде случаев диалоговый режим имеет неоспоримые преимущества.

Исторически более ранний пакетный режим ориентирован, в первую очередь, на повышение загрузки центральных устройств ЭВМ. Действительно, при использовании этого режима процессор загружен выполнением прикладных ("полезных") программ примерно на 40—50% больше, чем в диалоговом режиме. Общее время отладки

прикладных программ меньше (иногда в несколько раз) в диалоговом режиме, однако суммарные затраты машинного времени в этом режиме в 4—6 раз больше, чем в пакетном.

Задача практического освоения пакетного режима плохо разбивается на мелкие составляющие. Поэтому при его использовании многие начинающие бывают настолько обескуражены требуемым объемом начальных знаний и первыми результатами, что прекращают работу на ЭВМ.

Диалоговый режим лучше приспособлен к работе начинающих и непрофессиональных пользователей, чем пакетный. Время освоения ЭВМ в диалоговом режиме, как правило, меньше, чем в пакетном. Диалог повышает энтузиазм и активность обучающихся со средними способностями, что особенно важно при широком внедрении вычислительной техники. Хорошие результаты достигаются при его использовании и среди профессиональных пользователей.

Диалоговый режим нивелирует индивидуальные различия в ошибках и времени решения задач в системе "человек—ЭВМ" по сравнению с пакетным режимом.

Присущие рассмотренным режимам положительные качества объединяются в современных вычислительных системах, когда вместе с пользователями, работающими в диалоговом режиме, на вычислительной машине выполняется одна или несколько "фоновых" пакетных задач.

Экономический аспект затрат ресурсов следует оценивать с точки зрения как экономии времени пользователя, так и эффективности работы ЭВМ. Увеличение машинных ресурсов может оказаться необходимым не только для повышения эффективности деятельности пользователя, но и для улучшения технико-экономических показателей системы "человек—ЭВМ".

Рассмотрим, как меняются затраты на эту систему с развитием функциональных характеристик машинной составляющей (рис. 2.1). Кривая *1* соответствует суммарным затратам на разработку программной и аппаратной частей системы и ее эксплуатацию; кривая *2* — непосредственным затратам пользователя на общение с ЭВМ для получения требуемого результата; кривая *3* соответствует полным затратам на разработку и

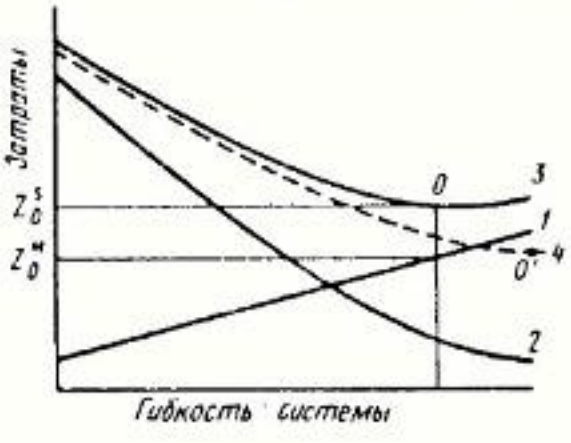

Рис. 2.1

эксплуатацию человеко-машинной системы. Минимум этих затрат приходится на точку *О*. Поэтому при затратах на машинную часть системы, меньших Z_O^M, общие затраты на систему будут превышать минимальное значение Z_O^S. Таким образом, дополнительные затраты на создание удобных аппаратных и программных средств, ориентированных на ведение активного диалога, могут снизить общие затраты в системе "человек—ЭВМ".

Средства вычислительной техники активно развиваются. Появляются новые технологии производства компонентов ЭВМ и разработки программного обеспечения, внедряются перспективные архитектурные и схемотехнические решения при проектировании и производстве вычислительной техники. Это ведет к существенному росту эффективности производства машинного компонента системы "человек—ЭВМ". В то же время стоимость часа человеческого труда в обществе со временем возрастает. Поэтому результирующая кривая затрат (рис. 2.1, кривая *3*) сместится таким образом, что новый минимум затрат окажется на графике правее и несколько ниже старого (точка *О*, кривая *4*). Таким образом, доля машинных затрат и гибкость вновь разрабатываемых систем "человек—ЭВМ" также объективно должны возрастать.

В диалоговых системах, реализованных на больших машинах, значительная часть затрат на разработку программного обеспечения приходится на поддержание режима разделения времени, создающего у пользователей иллюзию монопольной работы с ЭВМ. В то же время доля программного обеспечения в стоимости универсальных ЭВМ достигает 90% при незначительном росте производительности разработки программ (по некоторым источникам она не превышает 3% в год).

С развитием микроэлектроники оказалось возможным объединить положительные качества диалогового режима (непосредственное общение с вычислительной машиной и удовлетворительное время реакции системы) и пакетного режима (монопольное использование ресурсов ЭВМ и относительно дешевое математическое обеспечение). Такими качествами обладают персональные компьютеры, выполняемые на базе микроЭВМ и ориентированные на работу в диалоге с пользователями в какой-либо одной или нескольких сферах их профессиональной деятельности.

Электронные игры являются частным случаем использования диалогового режима, они повышают скорость реакции, уровень адаптации пользователя к новым непредвиденным ситуациям и в некоторых случаях его творческую активность. Широкому распространению электронных игр способствует минимальный уровень требований к подготовке пользователей. Возможно несколько модификаций электронных игр.

Игры на автономных устройствах отличаются тем, что на одном устройстие возможно проведение, как правило, одного вида игр зачастую с несколькими вариациями. Обычным является двойное использование электронных игр. Часто их комплектуют дополнительными бытовыми приборами: часами-будильником, термометром, микрокалькулятором и т.п. Наиболее известны следующие игры: шахматы, детская игра "Ну, погоди!" и др.

Игры, проводимые с использованием телевизионной приставки, характеризуются большим разнообразием. Для проведения новой игры достаточно сменить кассету, поэтому по сравнению с первым типом игр использование телеприставки позволяет иметь практически неограниченный банк домашних электронных игр. Однако телеигры отличаются невысокой алгоритмической сложностью и могут служить только для отработки моторных навыков.

Игры с использованием ЭВМ проводятся на специализированных и универсальных ЭВМ, в том числе и на персональных микрокомпьютерах. Банк игр в этом случае хранится на магнитных носителях; в специализированных микроЭВМ игры хранятся также в сменных кассетах постоянного запоминающего устройства (ПЗУ) или полупостоянного (перепрограммируемого) запоминающего устройства (ППЗУ). Кроме того, специализированные игровые микроЭВМ комплектуются органом управления, выполненным в виде рукоятки с двумя степенями свободы. Спектр применения игр на ЭВМ чрезвычайно широк: отработка моторных навыков (например, при обучении машинописи или стенографии), ознакомление с основами различных учебных дисциплин, углубленное изучение одного из разделов дисциплины, различные развлекательные игры.

Электронные игры чрезвычайно интенсивно развиваются. С пользой и интересом этими играми занимаются люди разного возраста — от дошкольников до пенсионеров.

Однако следует помнить, что электронные игры чрезвычайно азартны и вредны при чрезмерном увлечении ими. К отрицательным последствиям относятся: ухудшение зрения, онемение пальцев, повышенная утомляемость, гиподинамия и даже нервные стрессы. Поэтому пользователям чрезвычайно важно контролировать свои действия, а разработчикам электронных игр —предусматривать различные блокировки, например по времени работы программы.

2.2. Правила общения

Использование ЭВМ связано с решением разнообразных и сложных технических, математических и организационных проблем. Ими с большим или меньшим успехом занимаются специалисты в области вычислительной техники. Значительно менее исследованными остаются психологические и социальные факторы. В то же время исследователями установлено, что различия в эффективности деятельности отдельных пользователей перекрывают различия, обусловленные типом и характеристиками используемой вычислительной системы. Поэтому в человеко-машинных системах в первую очередь важны не потенциальные возможности аппаратно-программного компонента по решению прикладных задач, а предоставляемые этими системами сервисные возможности для обеспечения эффективной работы пользователей, в том числе непрофессиональных.

Разрабатываемые диалоговые системы в значительной степени ориентированы на потребности непрофессиональных пользователей. Созданы даже такие системы, которые имитируют привычную рабочую обстановку пользователя: письменный стол, папки с бумагами и т.д. В этих условиях важно умение в полной мере эффективно использовать предоставляемые системой возможности.

Кроме чисто профессиональных качеств для достижения положительных результатов при общении с ЭВМ важны и личные качества пользователя. В наибольшей степени способствуют успеху: настойчивость, интравертивность (обращенность вовнутрь, ориентация на собственные идеи), внутренняя управляемость пользователя, высокая терпимость к неопределенности (обычно на начальном этапе работы с машиной), умение быть точным

завершающей стадии работы), скромность, способность переносить стресс.

Общению между людьми и между человеком и ЭВМ присущи как отличительные, так и общие черты. Выяснив, какие правила ведения диалога являются общими, а какие — разными, можно значительно быстрее освоить диалоговую систему.

Под *диалогом* будем понимать непосредственный, соизмеримый по скорости с процессом мышления, обмен сообщениями между двумя субъектами для достижения определенной цели, при котором в любой момент времени возможна смена ролей информатора и реципиента (приемника информации).

Особенности ведения диалога между людьми в некоторых случаях могут быть распространены на диалог в системе "человек—ЭВМ".

Вступая в общение друг с другом, участники диалога преследуют определенные цели: пытаются получить ответ на какой-либо вопрос, поделиться тем, что их волнует, произвести определенное впечатление, просто "скоротать" время. Многие, кроме того, в важных случаях заранее продумывают "сценарий" диалога, с тем чтобы с большим успехом достичь намеченной цели. Поэтому выделим следующие основные особенности ведения диалога:

1. У вступающих в общение есть определенная цель и, возможно, заранее определен "сценарий" диалога. В процессе общения обычно выясняются цели, мнения участников диалога по обсуждаемой проблеме, выявляется общность и отличие взглядов. Как правило, минимизируются отличия в подходах к рассматриваемой проблеме и даже в отношении к ней, в том числе и путем изменения "сценария" диалога.

2. Участники диалога адаптируются друг к другу. В процессе диалога или на отдельных его этапах каждый из собеседников выполняет определенную разговорную роль: один спрашивает, другой отвечает; один активно пытается склонить другого к своей точке зрения, второй отстаивает свое мнение и т.д.

3. Происходит распределение разговорных ролей, каждый из участников диалога выполняет определенные обязательства по отношению к теме общения. В процессе ведения диалога разговорные роли могут неоднократно перераспределяться. Они подчиняются нормам поведения и

лингвистическим правилам, т.е. принимают определенные формы, которые, в известной степени, оказывают влияние на результат отдельных фрагментов или диалога в целом.

4. Принятые формы диалога и распределение разговорных ролей ведут к определенной предсказуемости результатов диалога.

Известная предсказуемость результатов диалога активизирует его участников на вступление в то общение, которое потенциально ведет к положительному результату и, наоборот, заставляет избегать контактов, которые могут привести к нежелательным последствиям. Усилия, затрачиваемые на вступление в общение и ведение диалога, в определенной степени пропорциональны важности предполагаемых результатов для его участников.

5. Участники диалога минимизируют свои усилия, предпочитая вступать в общение с заведомо определенным положительным результатом. При этом зачастую даже ожидаемый положительный результат может быть нивелирован предполагаемыми при общении отрицательными эмоциями, которые могут заставить одного из потенциальных участников диалога отказаться от общения.

Кроме внутренней логики, присущей сценарию диалога, успех общения во многом зависит от того, в каком настроении вступают в диалог его участники. Поэтому вначале важно выяснить это, добиться доброжелательного отношения к себе и заинтересовать в предполагаемой теме общения второго участника диалога. Чрезвычайно важно и самому относиться доброжелательно к собеседнику. Особое значение для каждого из участников диалога имеет оценка его разговорной роли собеседником. Поэтому ведение диалога в положительном эмоциональном тоне, возможно ближе к интересам собеседника, с максимальной высокой объективной оценкой его роли с большей вероятностью ведет к достижению желаемого результата.

6. Люди эмоциональны и преувеличенно чувствительны к недооценке их роли, в значительной мере форма общения оказывается для них важнее содержания. Поведение участников диалога существенно зависит от внешнего вида собеседника и его умения держаться.

Кроме отмеченных существуют, конечно, и другие особенности ведения диалога между людьми. Здесь были отмечены только наиболее существенные для целей организации диалога в системе "человек—ЭВМ". Пункты

1—5 присущи как диалогу в человеко-машинной системе, так и общению между людьми. Пункт 6 свойствен диалогу в системе "человек—ЭВМ" только в той степени, в которой разработчики диалоговой системы сумели передать свои эмоции вычислительной машине. При этом ведение диалога с ЭВМ регламентируется более строгими правилами и ограничениями, чем диалог между людьми.

Диалог «человек—ЭВМ» представляет собой разновидность диалога, где в качестве одного из субъектов выступает человек, а в качестве другого — диалоговая система, и обмен интерактивными посланиями ведется в соответствии с заранее определенными языком и формой общения.

При освоении системы особое внимание надо уделить уточнению своих действий в тупиковых ситуациях. После изучения системы, заказа терминального времени и других машинных ресурсов (дисковой памяти, архивной магнитной ленты и др.), подготовки исходного материала к первому сеансу работы с ЭВМ (пусть это будет небольшая проблемная задача) можно приступить к работе за терминалом.

Во время работы с системой пользователь должен помнить, что она защищена от ошибочных действий, которые могут ее разрушить или причинить пользователям большой ущерб. Если система все-таки будет разрушена — значит она просто неудачно спроектирована, и в следующих версиях разработчики должны учесть эту недоработку.

В процессе работы может возникнуть ситуация, когда пользователю неясно, что от него требуется, каковы его дальнейшие действия. В большинстве диалоговых систем реализованы функции помощи, обратившись к которым пользователь может преодолеть свои затруднения.

При работе с системой следует обращаться к машине за помощью. Важно помнить, что машина помогает пользователю только в рутинной части работы, оставляя ему творческую часть работы и принятие решений. Кроме того, у вычислительной машины отсутствуют эмоции, поэтому многократные повторы и "непонятливость" пользователя, как правило, не отражаются на состоянии диалога с ЭВМ.

Диалоговая система может работать в темпе, превышающем потребности пользователя. В этом случае следует работать в своем темпе, просматривая все сообщения системы и не "гонясь" за ЭВМ.

Новые сообщения, выдаваемые системой, следует внимательно изучить, выделив те из них, которые влияют на дальнейший ход диалога. При неоднократной неудаче полезно переформулировать запрос к ЭВМ, подойти к проблеме с другой стороны. В случае неудачной попытки выполнить функционально сложный запрос следует разбить его на более элементарные действия, что дает, как правило, хороший результат.

Понимание логики построения машинного диалога поможет пользователю, получив удачный вариант взаимодействия в одном режиме и проведя аналогию, с большим успехом достичь результата в похожей ситуации другого режима работы системы.

При отсутствии новых идей по отладке программы следует прекратить бесплодные попытки, чтобы не испортить полученные за сеанс предварительные результаты. Не следует пытаться решить все проблемы за экраном: опыт показывает, что это, как правило, малоэффективно. Наиболее правильные решения пользователь находит обычно за столом.

В конце сеанса необходимо получить распечатки, которые могут понадобиться для анализа результатов работы за экраном дисплея. Это могут быть: текст последней версии программы, результаты тестирования и т.д.

После окончания работы с системой необходимо проанализировать результаты, если этого не было сделано во время сеанса, на порядок полученных величин. Ошибочные результаты машинного расчета в большинстве случаев возникают из-за ошибок пользователя. Крайне редко они появляются или из-за машинного сбоя или из-за выхода ЭВМ из строя. При этом ошибки достигают, как правило, нескольких порядков, поэтому их легко заметить.

Необходимо также анализировать весь ход диалога с ЭВМ, особенно в случае отрицательного результата, найти свои ошибки, наметить пути их устранения: более детально ознакомиться с документацией по системе, проконсультироваться со специалистами, разработать новую стратегию ведения диалога и т.д. После реализации

намеченных мероприятий следует подготовить материал к следующему сеансу, а также план работы за дисплеем.

Итак, резюмируя правила ведения диалога, можно посоветовать выполнение следующих рекомендаций как при работе с ЭВМ, так и в общении с людьми:

— знайте, что вы хотите;
— умейте слушать собеседника;
— пользуйтесь помощью при затруднениях;
— обучайтесь в процессе диалога;
— перестраивайтесь в процессе диалога при неудачной попытке установить контакт;
— анализируйте результаты общения.

2.3. Типы диалога

В прикладной диалоговой системе можно выделить два функциональных компонента:

— собственно прикладную систему, с которой работает пользователь;
— диалоговый компонент, управляющий диалогом между пользователем и прикладной системой.

Примерами прикладных систем являются все системы, с которыми работают пользователи ЭВМ в сфере своих профессиональных интересов, — различного рода автоматизированные системы управления (АСУ), системы автоматизированного проектирования (САПР), пакеты прикладных программ (ППП) и др. Эти системы характеризуются тем, что диалог ведется в терминах проблемной области и на выходе пользователь получает конечный результат, являющийся целью его общения с вычислительной машиной. Некоторые системы могут работать не только в диалоговом, но и в пакетном режиме. Семантика конечного результата при этом не зависит от используемого режима работы.

Диалоговый компонент является надстройкой над прикладной системой и помогает пользователю уточнить постановку проблемной задачи, представляя для этого соответствующие лингвистические формы и набор сервисных функций. Диалог в этом случае ведется в терминах диалоговой системы и называется *метадиалогом*. Используемая терминология практически не зависит от проблемной области. Наличие в прикладной системе развитого метадиалога дает возможность пользователю,

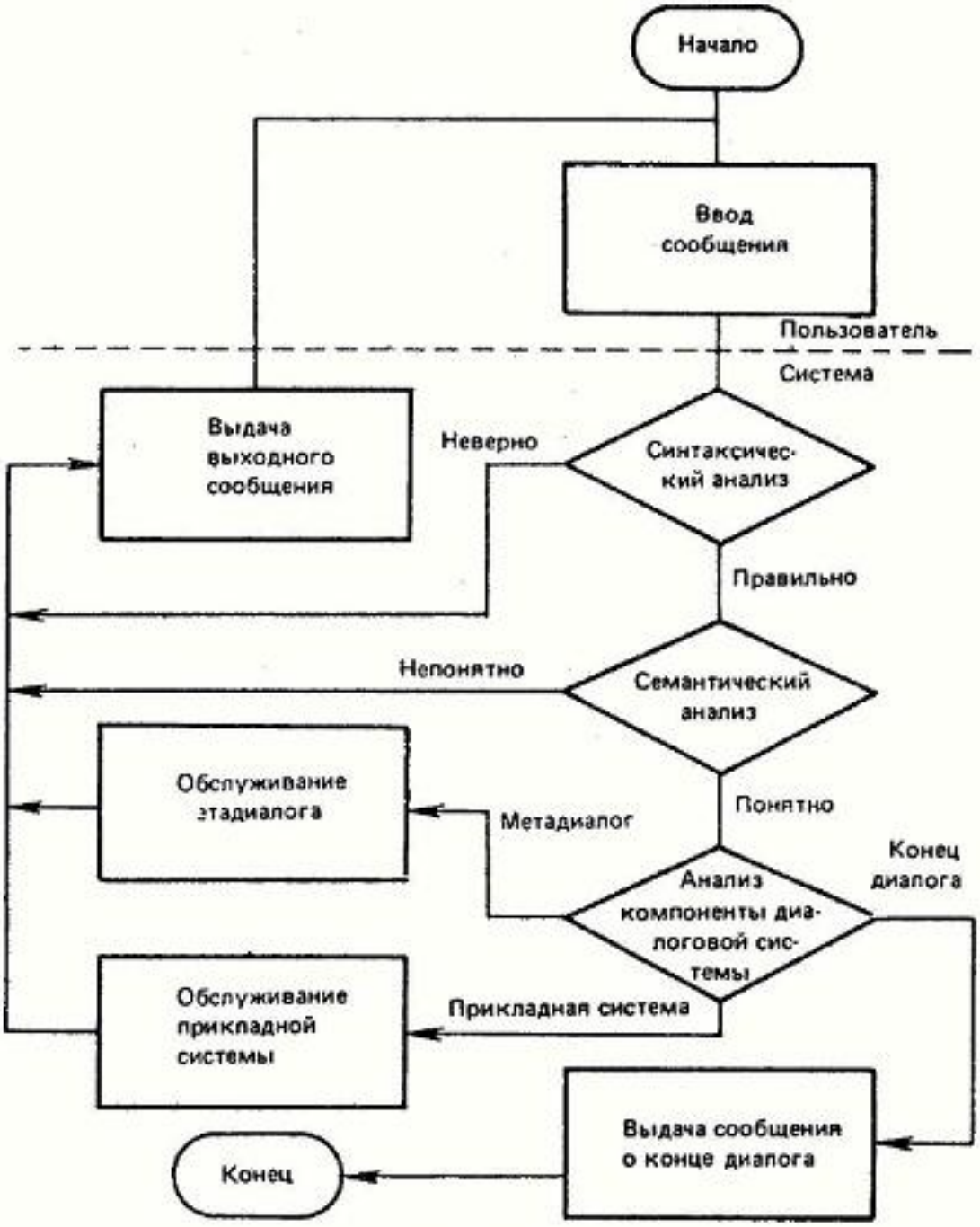

Рис. 2.2

знакомому только с проблемной областью, получать требуемые результаты без предварительного детального изучения диалоговой системы. Поэтому доступность и распространенность прикладной системы во многом зависят от качества построения диалоговой надстройки.

С учетом наличия двух компонентов в диалоговой системе на рис. 2.2 представлена обобщенная схема алгоритма простейшего диалога с машиной. При этом однократное прохождение по одной из ветвей схемы алгоритма от блока НАЧАЛО до блока КОНЕЦ или

до возврата к началу схемы алгоритма будем называть *шагом диалога,* или *транзакцией.* Шаг диалога характеризуется интерактивным взаимодействием человека с ЭВМ, т.е. в данном случае в ответ на ввод сообщения пользователя после его обработки диалоговой системой выдается выходное сообщение, отражающее состояние системы или диалога.

Для более быстрого освоения диалоговой системы пользователю необходимо знать основные типы диалога и формы его реализации на вычислительной машине. Наиболее известны в человеко-машинной среде следующие типы диалога:

— меню;
— вопросы, требующие ответа ДА /НЕТ;
— шаблон;
— простой запрос;
— команда;
— взаимодействие на естественном языке.

Меню — наиболее распространенный тип диалога. При обращении система инициирует диалог. Таким образом, шаг диалога начинается с выдачи системой входного сообщения. После выдачи пользователем выходного сообщения и обработки его системой шаг диалога завершается.

Возможно несколько форм машинной реализации данного типа диалога. Во всех случаях в качестве входного сообщения на экране дисплея высвечивается подмножество (как правило, нумерованное) функций системы, реализация которых возможна в текущем состоянии диалога. Такое входное сообщение будем называть меню.

Пользователь может выбрать требуемую функцию одним из следующих основных способов:

а) набором требуемой директивы или ее аббревиатуры;

б) набором номера выбранной директивы;

в) подведением курсора в строку экрана с требуемой директивой;

г) нажатием фотоселектора в соответствующей строке экрана.

Наиболее экономичным по затратам времени пользователя является способ г), затем способы б) и в). Однако иногда применяется способ а), так как при его использовании происходит подтверждение директивы и возможность

случайного набора соседней директивы меньше, чем при других способах.

Рассмотрим примеры шага диалога двух возможных форм реализации меню.

Пример 2.1

```
СООБЩЕНИЕ СИСТЕМЫ:      ВЫБЕРИТЕ ТРЕБУЕМУЮ ФУНКЦИЮ
                        1.ВВОД ДАННЫХ
                        2.КОРРЕКТИРОВКА МАССИВА
                        3.ВЫВОД ИНФОРМАЦИИ
                        4.УНИЧТОЖЕНИЕ МАССИВА
СООБЩЕНИЕ ПОЛЬЗОВАТЕЛЯ: 3
```

Пример 2.2

```
СООБЩЕНИЕ СИСТЕМЫ:      ВЫБЕРИТЕ ТРЕБУЕМУЮ ФУНКЦИЮ
                        ВВОД ДАННЫХ - ВВД
                        КОРРЕКТИРОВКА МАССИВА - КРМ
                        ВЫВОД ИНФОРМАЦИИ - ВЫИ
                        УНИЧТОЖЕНИЕ МАССИВА - УНМ
СООБЩЕНИЕ ПОЛЬЗОВАТЕЛЯ: ВЫИ
```

В обоих примерах система перейдет в режим вывода информации. Если возможно несколько вариантов вывода, то в следующих шагах диалога система будет уточнять: какую именно информацию, в каком виде и на какое устройство надо вывести.

Графически диалог можно представить в виде графа. Его узлы соответствуют подмножеству функций системы, высвечиваемому на экране дисплея (входные сообщения), а дуги, направленные вниз, — возможным альтернативам пользователя (выходные сообщения). При этом число дуг равно числу предлагаемых системой функций. Дуга вместе с вершиной, из которой она исходит, соответствует шагу диалога. Верхний узел (корень) дерева соответствует начальному сообщению системы.

На рис.2.3 приведен граф диалога меню в виде дерева. Корень дерева (верхний узел, соответствующий начальному сообщению системы) и исходящие из него дуги соответствуют шагу диалогов, приведенных в примерах 2.1 и 2.2; предлагаемые в меню функции — дугам дерева в

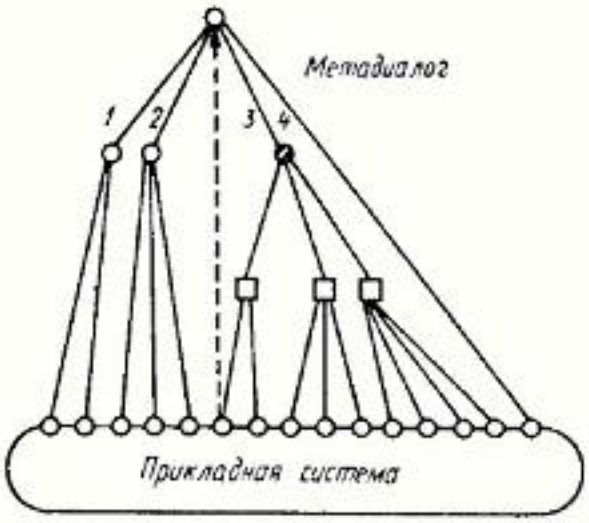

Рис. 2.3

и 2.2; предлагаемые в меню функции — дугам дерева в направлении слева направо. После выполнения шага диалога примера 2.1 или 2.2 входное сообщение (меню) системы на дисплее будет соответствовать заштрихованному узлу графа. После выполнения следующего шага диалога система высветит одно из меню, обозначенных прямоугольником. Движение вниз по дереву графа меню в направлении прикладной системы называется *погружением.*

Для выполнения функции, соответствующей соседним ветвям дерева графа, при метадиалоге или после работы с прикладной системой используется механизм "всплытия". Его можно реализовать двумя способами:

а) последовательным движением вверх по дугам графа диалога до узла, непосредственно содержащего заданную функцию или содержащего ее в узлах, в которые из него можно перейти, используя "погружение";

б) непосредственным переходом в корень дерева графа диалога с последующим погружением.

Графически второй способ соответствует наличию в каждом узле ориентированной к вершине дерева дуги графа. Одна из таких дуг изображена на рис.2.3 штриховой линией.

Механизм "всплытия" может быть представлен пользователю одним из двух способов:

а) его описание содержится только в документации на систему (в этом случае из любой вершины "всплытие" происходит, как правило, единообразно);

б) кроме документации одна или несколько альтернатив каждого меню системы описывает механизм "всплытия".

Вопросы, требующие ответа ДА/НЕТ, являются частным случаем диалога типа "меню", когда в шаге диалога в качестве альтернатив пользователю предлагаются два ответа: ДА и НЕТ.

В диалоговой системе возможны разные реализации ответа пользователя:

а) ответ дается с помощью первых букв слов ДА и НЕТ;

б) в качестве ответа используются какие-нибудь символы, соответствующие ДА и НЕТ (например, 1 и 0);

в) ответ набирается полностью (ДА или НЕТ);

г) одна из альтернатив (ДА или НЕТ) предполагается по умолчанию в комбинации с одной из трех описанных реализаций.

Пример 2.3

После обработки сообщения пользователя в примере 2.3 система перейдет в исходное состояние.

Графическая интерпретация рассматриваемого типа диалога изображена на рис. 2.4 — это граф бинарного дерева, из каждого узла которого выходит вниз (к прикладной системе) по две ветви. "Погружение" и "всплытие" в этом типе диалога аналогичны меню. В данном типе диалога для работы с прикладной системой приходится выполнять, как правило, больше шагов по сравнению с диалогом типа "меню". Основным достоинством вопросов, требующих ответа ДА/НЕТ, является простота выполнения каждого отдельного шага диалога.

Шаблон — это инициированный системой диалог, на каждом шаге которого система воспринимает только синтаксически ограниченное входное сообщение пользователя в соответствии с заданным форматом. Как и в меню, диалог начинает система. Данный тип диалога называют также параметрической системой. Возможные варианты ответа пользователя ограничиваются форматами, предъявляемыми ему на дисплее. Поэтому гибкость пользования системой с диалогом такого типа относительно невысока, однако достаточно низка ее операционная сложность.

Из нескольких возможных вариантов реализации такого типа диалога назовем два:

1) указание системой на экране дисплея формата вводимого пользователем сообщения;

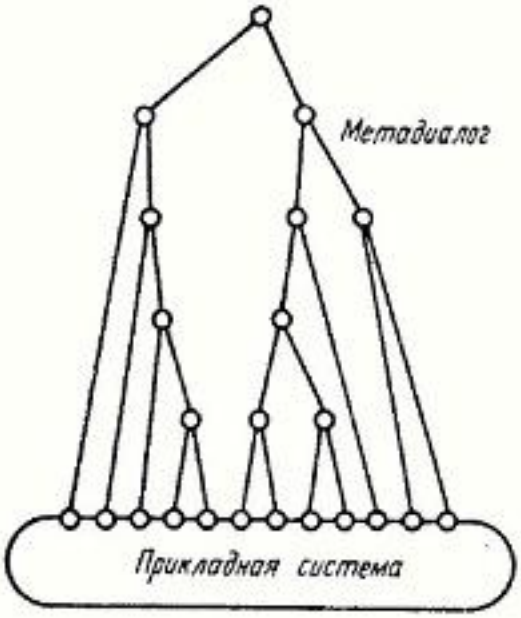

Рис. 2.4

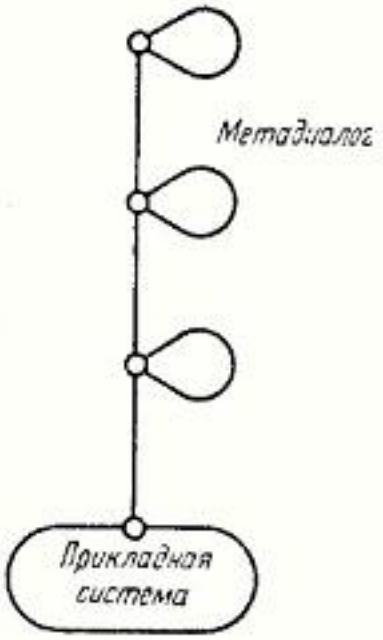

Рис. 2.5

2) резервирование места для сообщения пользователя в тексте сообщения системы на экране дисплея (например, с помощью поля повышенной яркости).

Шаблон обычно используется для ввода данных, значения которых или "прозрачны" (например, значение пола при заполнении анкеты или текущая дата), или являются профессиональными терминами и поэтому они известны пользователю. Однако форма подачи этих данных не определяется их значениями. Например, дата 1 июля 1987 года может быть введена в систему в записанном виде и в более компактных форматах: 87/182 (182-й день 1987 года), 87/07/01, 01/07/87 и др. Символ "/" здесь используется в качестве разделителя. Значение пола можно ввести полностью, например: "мужской" сокращенно — "муж" и по первой букве — "м".

Пример 2.4

СООБЩЕНИЕ СИСТЕМЫ:	ВВЕДИТЕ ВАШИ ДАННЫЕ В СЛЕДУЮЩЕЙ
(ПРЯМОУГОЛЬНИКИ СООТВЕТСТВУЮТ ПОЛЯМ ПОВЫШЕННОЙ ЯРКОСТИ НА ЭКРАНЕ ДИСПЛЕЯ)	СТРОКЕ В ЗАДАННОМ ФОРМАТЕ □□□□КМ□□□М
СООБЩЕНИЕ ПОЛЬЗОВАТЕЛЯ:	□360КМ500М
(В ОТФОРМАТИРОВАННОМ ПОЛЕ)	

Пример 2.5

СООБЩЕНИЕ СИСТЕМЫ:	ЗАДАЙТЕ ДАТУ В ФОРМАТЕ ДД/ММ/ГГ
СООБЩЕНИЕ ПОЛЬЗОВАТЕЛЯ:	01/07/87

Графически диалог типа "шаблон" можно интерпретировать с помощью графа, приведенного на рис. 2.5. Из каждой вершины графа выходит по две дуги. Дуга, направляющаяся к следующему узлу, соответствует семантически правильному сообщению пользователя, а дуга, замыкающаяся на одном узле, — семантически неправильному сообщению пользователя. При неверном сообщении система выдаст предупреждение об ошибке и останется в начале текущего шага диалога.

На рис. 2.6 проиллюстрирована реализация примера 2.5 с помощью двух типов диалога: меню и шаблона. Из рисунка видно, что в данном случае более компактным и удобным для пользователя будет диалог типа "шаблон".

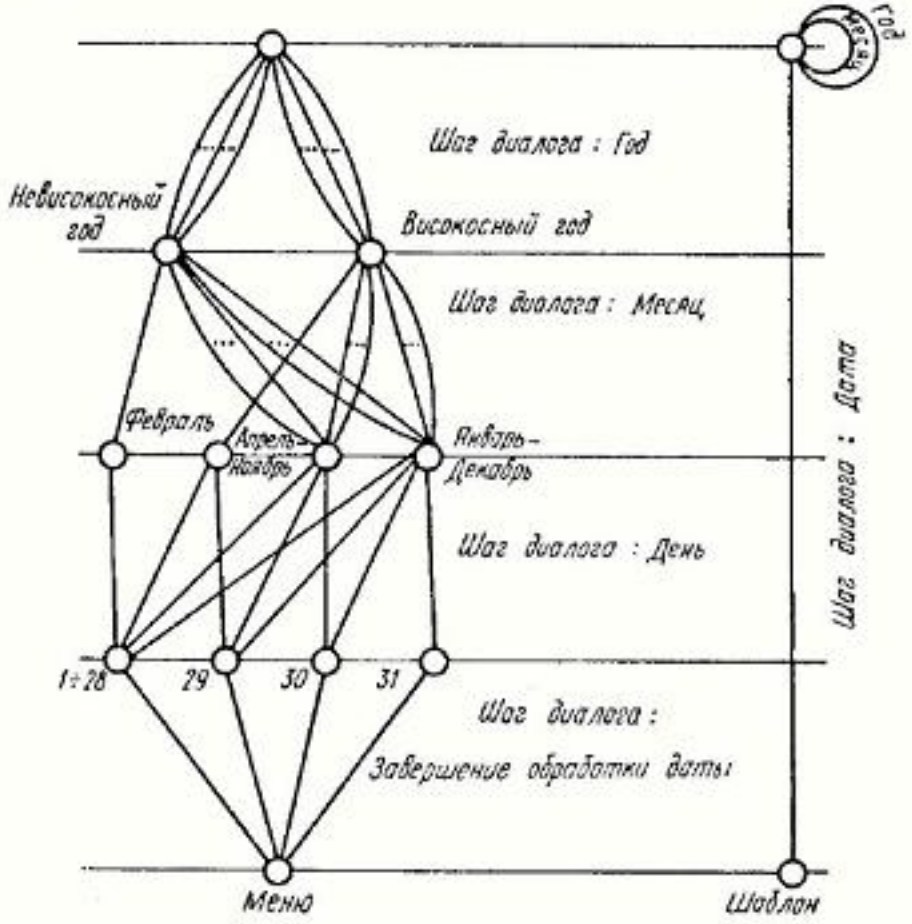

Рис. 2.6

Каждому фрагменту сообщения, заданному своей частью формата, соответствует шаг диалога меню. В примере 2.5 — это "год" (формат ГГ), "месяц" (формат ММ), "день" (формат ДД).

Принципиальным отличием использования шаблона в примерах 2.4 и 2.5 является то, что в примере 2.4 вводятся семантически независимые (для диалоговой системы) данные, а в примере 2.5 — семантически зависимые: число дней зависит от месяца, а число дней в феврале зависит еще и от года.

Простой запрос является частным случаем шаблона. Обычно используется в различного рода автоматизированных системах при сборе данных. Пользователю предоставляется возможность вводить массив данных по формату, заданному системой. Весь диалог сводится к одному шагу. Формат может выводиться на экран, а может быть описан только в руководстве по работе с системой.

Простой запрос — это тип диалога, инициируемый системой и состоящий из одного шага, где пользователю предоставляется возможность ввести более одного сообщения по заранее определенному системой единому формату.

Диалог завершается "всплытием", которое выполняется при задании пользователем определенной директивы.

Пример 2.6

```
СООБЩЕНИЕ СИСТЕМЫ:        ВВОДИТЕ ДАННЫЕ В СООТВЕТСТВИИ  С
                           ЗАДАННЫМ ФОРМАТОМ
                           ГОДНЫЕ ДЕТАЛИ БРАК
СООБЩЕНИЕ ПОЛЬЗОВАТЕЛЯ:    ГОДНЫЕ ДЕТАЛИ БРАК
                           ГОДНЫЕ ДЕТАЛИ БРАК
                             .....
                           ГОДНЫЕ ДЕТАЛИ БРАК
```

Примечание. Символ * использован в данном случае в качестве указателя конца (шага) диалога.

Рис. 2.7

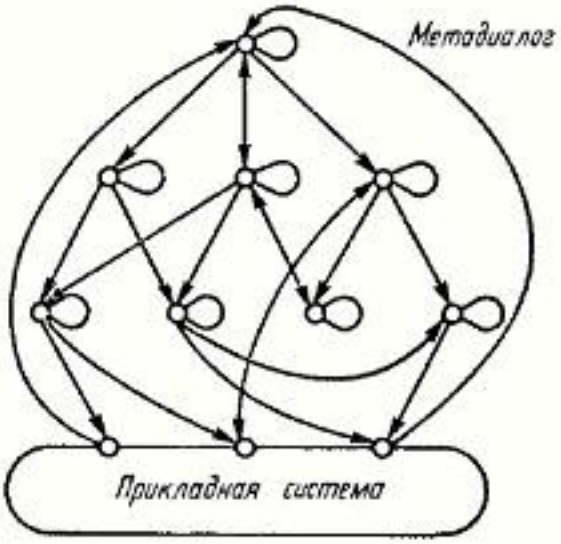

Рис. 2.8

Графическая интерпретация простого запроса приведена на рис.2.7.

Команда — это инициированный пользователем тип диалога, при котором выполняется одна из допустимых на данном шаге диалога директив (команд) пользователя.

Набор допустимых команд, как правило, отсутствует на экране, однако с помощью специальных директив его можно вывести для ознакомления. В случае задания недопустимой директивы выдается предупредительное сообщение об ошибке, и система остается в начале текущего шага диалога.

Пример 2.7

СООБЩЕНИЕ ПОЛЬЗОВАТЕЛЯ:	ВВЕСТИ ДАННЫЕ
СООБЩЕНИЕ СИСТЕМЫ:	ЗАДАЙТЕ ФОРМАТ

Графически данный тип запроса может быть представлен с помощью ориентированного графа (рис. 2.8). Характерным является наличие дуг, замыкающихся на каждом

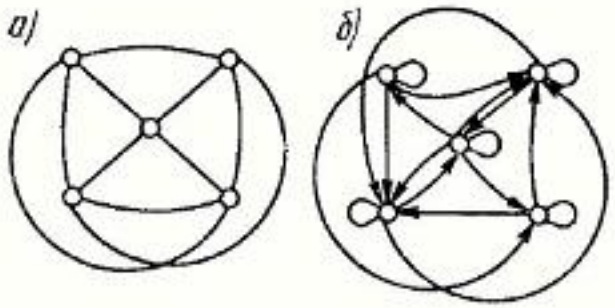

Рис. 2.9

узле графа. Такие дуги соответствуют недопустимым директивам на данном шаге диалога.

Взаимодействие на естественном языке возникает по инициативе пользователя. Этот тип диалога налагает наименьшие ограничения на форму ведения общения со стороны пользователя, которому предоставляется возможность свободно выбирать формулировку задачи, используя естественный язык. В связи с неоднозначностью естественных языков со стороны системы возможны вопросы, уточняющие формулировки пользователя и предметную область рассматриваемой проблемы. Взаимодействием на естественном языке будем называть тип диалога, инициированный пользователем и приводящий к решению поставленной им задачи, когда общение со стороны пользователя ведется на естественном языке.

Научные и технические трудности создания системы, взаимодействующей с пользователем на естественном языке, приводят к реализации квазиестественного (похожего на естественный) или проблемно-ориентированного (для общения в одной узкой проблемной области) языка. Поэтому пользователь должен быть готов к тому, что диалоговая система не распознает запрос с первого раза (если распознает его вообще). Развитие диалоговых систем в направлении повышения их интеллектуальности приводит к повышению их гибкости и адаптивности, а следовательно, и большей восприимчивости к запросам на квазиестественном языке.

Графическая интерпретация взаимодействия на естественном и квазиестественном языках приведена на рис. 2.9, *а, б* соответственно. Отличительной особенностью графа

диалога на естественном языке является наличие парных противоположно направленных дуг графа между любыми парами узлов (на рисунке каждая пара противоположно направленных дуг обозначена одной неориентированной дугой). Граф диалога на квазиестественном языке является ориентированным, характерная его особенность — наличие в каждом узле дуг, замыкающихся на этом же узле, что соответствует недопустимым директивам на данном шаге диалога. В отличие от графа диалога на естественном языке на квазиестественном не все пары узлов соединены противоположно направленными дугами.

Анализ описанных типов диалога показывает, что применение только одного из них (за исключением последнего) не позволяет создать достаточно гибкую и эффективную систему. Рассмотрим некоторые ограничения на структуру диалога, накладываемые разными его типами:

а) вопросы, требующие ответа ДА/НЕТ — мала эффективность диалога, велика избыточность графа диалога при усложнении темы общения, чрезмерное их использование приводит к переутомлению пользователя;

б) шаблон — эффективен только в случае ввода фрагмента табличных данных, при ограниченном (от двух до четырех) выборе возможных альтернатив и малом числе шагов диалога;

в) команда — мала эффективность при значительном числе (не меньше пяти) шагов диалога и альтернативных команд на каждом его шаге.

Использование в диалоговой системе нескольких типов диалога повышает гибкость системы и снижает ее операционную сложность. Работая с диалоговой системой, пользователь должен быть готов к тому, что ему придется встретиться с несколькими типами диалога, в том числе и с такими, которые не были описаны в этом разделе.

Вопросы для самопроверки

2.1. Какая составляющая вычислительной системы дешевле: программная или аппаратная?

2.2. В чем принципиальное отличие диалогового режима работы с ЭВМ от пакетного?

2.3. Назовите основные терминальные устройства при работе в диалоговом и пакетном режимах.

2.4. Какой режим: пакетный или диалоговый, более экономичен:

а) по использованию аппаратных средств;

б) по потреблению программных средств;

в) по затратам времени пользователя?

2.5. Приведите пример диалога на естественном языке.

2.6. В каких случаях при построении диалоговой системы неэффективно использовать простой запрос, а в каких — меню?

2.7. Разработайте алгоритм определения дня недели (понедельник, ..., воскресенье) в зависимости от даты, используя знания о типах диалога.

2.8. Нарисуйте схему алгоритма шага диалога для меню.

Глава 3
ПРОДОЛЖЕНИЕ ДИАЛОГА

3.1. Использование функциональной клавиатуры

В процессе первоначального ознакомления с дисплеем основное внимание уделяется изучению клавиатуры и приобретению навыков работы с ней. Действительно, чем лучше пользователь знает клавиатуру, тем лучше и быстрее он набирает на экране дисплея требуемый текст. Зададимся вопросом: как часто приходится набирать на клавиатуре одинаковые комбинации символов (слова, словосочетания)?

Ответ на этот вопрос неоднозначен. Все зависит от содержания набираемого на клавиатуре текста. Для программы на алгоритмическом языке характерно наличие повторяющихся ключевых слов (названий операторов и переменных). Поэтому в применении к текстам программ довольно часто приходится набирать одинаковые слова. Представим теперь, что пользователь набирает текст на клавиатуре со скоростью хорошей машинистки. Однако неограниченно увеличивать скорость печатания нельзя, так как физические возможности человека ограничены. Можно сказать, что ручной ввод с клавиатуры имеет сравнительно низкую производительность.

Существуют оптические считывающие устройства. Они позволяют распознавать даже рукописный текст, написанный достаточно неразборчиво. Уже разработаны устройства распознавания речи. В будущем средства общения с ЭВМ получат техническую поддержку — можно будет говорить с ЭВМ, предлагать ей прочитать свои рукописи, общаться, как с человеком. Однако пока эти устройства мало распространены и при общении с ЭВМ чаще всего пользуются дисплеем с клавиатурой. Как же повышать производительность ручного ввода? Один из способов — научиться печатать "слепым" десятипальцевым методом.

Для этого нужно пройти курс машинописи. Другой способ (дополняющий первый) — использовать функциональную клавиатуру, представляющую собой набор специальных клавиш или средств, обеспечивающих специальное использование обычных клавиш, предназначенных для повышения производительности ручного ввода с клавиатуры дисплея. Специальные клавиши дают возможность пользователю прежде всего ускорить ввод информации. При этом диалоговая система, с которой общается пользователь, имеет возможность различать обычные и функциональные клавиши. Как правило, функциональные клавиши имеют отличный от остальных клавиш код; может быть дополнительная информация о том, что данная клавиша — функциональная.

Различают два процесса ввода: ручной с клавиатуры и функциональный.

Если при *ручном вводе* нажатие каждой клавиши клавиатуры означает ввод соответствующей буквы (ее кода) и только, то при *функциональном вводе* можно выполнить любое действие, отличное от ввода буквы. Мало того, функциональная клавиатура и создана для того, чтобы выполнять специальные действия. Рассмотрим одно из них — резервирование ключевых слов. Каждой определенной функциональной клавише можно сопоставить определенное ключевое слово языка программирования.

Допустим, на клавиатуре дисплея имеется дополнительное поле с 12 функциональными клавишами, при этом у каждой функциональной клавиши есть свой порядковый номер (от 1-го до 12-го) (на примере дисплея ЕС-7927).

Предположим далее, что пользователь набирает на клавиатуре программу на языке ПАСКАЛЬ. Как известно, в языке ПАСКАЛЬ довольно часто встречаемыми словами являются: BEGIN, END, REPEAT, UNTIL, WHILE, FOR, IF. Сопоставим каждому слову функциональную клавишу:

BEGIN	ФК1
END	ФК2
REPEAT	ФК3
UNTIL	ФК4
WHILE	ФК5
FOR	ФК6
IF	ФК7

Справа написаны соответствующие каждому ключевому слову функциональные клавиши (ФКi — функциональная клавиша с номером i).

Операция сопоставления функциональной клавише определенной текстовой комбинации называется резервированием данной функциональной клавиши. При нажатии зарезервированной клавиши, начиная с текущей позиции курсора, выводится соответствующее клавише ключевое слово. Производительность ручного ввода при этом существенно повышается. Обеспечение функциональных клавиш полностью зависит от диалоговой системы. Если программирование функциональных клавиш заложено в диалоговой системе, то пользователь имеет возможность зарезервировать функциональные клавиши для своих целей. Действительно, в языке ФОРТРАН или БЕЙСИК ключевые слова отличаются от принятых в языке ПАСКАЛЬ и им можно сопоставить соответствующие функциональные клавиши. Сделав это один раз в начале сеанса работы, можно обеспечить возможность ускоренного ввода исходной программы на данном алгоритмическом языке.

Программирование функциональных клавиш — дополнительная возможность, предоставляемая пользователю диалоговой системы. Очевидно, что в диалоговой системе в этом случае предусматривается режим программирования функциональных клавиш, который обеспечивает просмотр и изменение их назначений.

В процессе работы диалоговая система, обеспечивающая работу с функциональной клавиатурой, должна допускать динамическое переопределение функциональных клавиш. Таким образом, можно определить набор функциональных клавиш, которые по мере необходимости могут заменять друг друга.

Однако возможности функциональной клавиатуры не ограничиваются вставкой в текст ключевых слов. Нажатие определенной функциональной клавиши может порождать одно или несколько действий или процессов в диалоговой системе. Имея возможность назначить конкретной функциональной клавише одно действие или совокупность действий в диалоговой системе, пользователь тем самым существенно сокращает число рутинных клавиатурных операций. Кроме того, диалоговая система может обес-

печивать выполнение определенных действий с функциональной клавиатурой по умолчанию. Набор таких действий определяется набором стандартных сменных функциональных клавиатур, который представляет собой совокупность названий функциональных клавиш и соответствующих им назначений, хранящихся в памяти диалоговой системы.

Сменная функциональная клавиатура может храниться в памяти под именем (номером) и вызываться оттуда по имени (номеру) или нажатием определенной функциональной клавиши. Нажатие определенной функциональной клавиши может привести к изменению состояния распределения остальных функциональных клавиш.

Таким образом, находясь на каком-либо уровне диалога, можно пользоваться определенной для этого уровня функциональной клавиатурой.

Различают два способа ввода с помощью функциональных клавиш без подтверждения и с подтверждением.

При вводе без подтверждения нажатие функциональной клавиши приводит к немедленному выполнению зарезервированного за ней действия. Действия может не быть, если за данной клавишей ничего не зарезервировано. Примером использования ввода без подтверждения могут служить функции управления экраном дисплея: стирание экрана, установка курсора в левый верхний угол или другое фиксированное место и т.п. Функциональная клавиша помощи также соответствует вводу без подтверждения.

Основное различие вводов без подтверждения и с подтверждением состоит в том, что при вводе с подтверждением у пользователя есть право выбора: выполнять или не выполнять зарезервированное за клавишей действие.

При вводе с подтверждением после нажатия функциональной клавиши может произойти одно из возможных действий:

1) начиная с текущей позиции курсора выводится текст, зарезервированный за данной клавишей. Диалоговая система переходит в состояние ожидания. При этом пользователь может внести определенные изменения в текст и затем нажатием клавиши ВВОД послать его в машину (обратите внимание, что использование

функциональных клавиш для ввода ключевых слов соответствует вводу с подтверждением);

2) содержимое экрана стирается и выводится сообщение, требующее подтверждения начала зарезервированного за клавишей действия. В случае отказа от запрошенного действия можно вернуть систему в текущее состояние;

3) содержимое экрана стирается и выводится список всех зарезервированных клавиш. Если после этого нажать клавишу ВВОД, то производится возврат системы в текущее состояние. Такое действие соответствует режиму подсказки или помощи. Обычно диалоговая система резервирует одну функциональную клавишу как клавишу помощи. Если пользователь забыл какое-либо назначение функциональной клавиши, он может получить справку в любой момент работы с помощью этой клавиши;

4) содержимое экрана стирается и выводится список зарезервированных за данной функциональной клавишей действий системы. Если после этого нажать клавишу ВВОД, то выполняется действие. При этом имеется возможность внести требуемые изменения.

Рассмотрим еще несколько особенностей функциональной клавиатуры. Существуют специализированные дисплеи, содержащие сменную клавиатуру. При этом смена производится не программным способом, а путем замены всех названий клавиш (сменное поле клавиатуры). Если при этом используется сенсорная клавиатура, то одна пластинка с названием клавиш заменяется на другую и с помощью переключателя или программным способом осуществляется настройка на новую функциональную клавиатуру. Это удобно, так как названия клавиш в этом случае всегда находятся перед глазами пользователя (одним из примеров использования специализированных дисплеев со сменной клавиатурой является автоматизированная телевизионная обучающая система АТОС Белорусского государственного университета).

В качестве примера рассмотрим варианты использования функциональной клавиатуры дисплея ЕС-7927 в системе "Фокус". Все функциональные клавиши дисплеев в данной системе могут иметь двойное назначение: стандартное и пользовательское. Первое назначение предусмотрено для программного расширения функций управляющей клавиатуры. Например, отсутствие клавиши установки курсора в начало экрана компенсируется наличием соответствующей программы, вызываемой при нажатии клавиши

ПФ3. Клавиша ПФ1 предназначена для установки на экране специального символа начала текста, а клавиша ПФ7 — для вывода копии экрана на печатающее устройство. Все эти назначения имеют общесистемный характер и действуют во всех диалоговых программах системы.

Кроме стандартных назначений пользователь может определять собственные назначения клавиш. Если в процессе работы с системой приходится часто вводить один и тот же запрос, то для удобства текст такого запроса может быть назначен конкретной функциональной клавише. Тогда при нажатии данной клавиши необходимый запрос будет автоматически вводиться в машину и набирать его на клавиатуре не требуется. Такое назначение имеет тип запроса и называется R-назначением. Если текст запроса необходимо выдать на экран для возможной модификации, а потом ввести его в машину с помощью клавиши ВВОД, то такое назначение имеет тип вывода и называется D-назначением. Кроме того, существуют R- и L-назначения, предназначенные для сцепления текста, набранного на клавиатуре, с текстом, назначенным определенной функциональной клавише. R-назначение осуществляет сцепление справа, а L-назначение — слева. Еще один тип назначения — P-назначение — используется для более сложной программной обработки.

Переключение между стандартным и пользовательским назначениями обеспечивается клавишей ВЫЗОВ ТЕСТА. Если при нажатии этой клавиши в позиции курсора высвечивается буква F, то функциональная клавиатура переведена в стандартное назначение, а если высвечивается буква U — то клавиатура переведена в пользовательское назначение.

Как отмечалось ранее, на стандартных алфавитно-цифровых дисплеях ЕС-7927 используется дополнительное поле функциональной клавиатуры. Обычные функциональные клавиши, таким образом, строго разделены. Это обусловлено тем, что сообщение, посылаемое в ЭВМ с экрана ЕС-7927, сначала набирается на экране дисплея и передается в машину только после нажатия соответствующей клавиши. Иногда запрос пользователя не содержит текста сообщения. Клавиши, которые вызывают передачу сообщения в ЭВМ, — это клавиша ВВОД и функциональные клавиши. Клавиша ВВОД и каждая из функциональных клавиш имеют собственный код, передаваемый вместе с сообщением диалоговой системе, которая, в свою очередь, распознает, какая из клавиш была нажата. При этом двенадцати функциональных клавиш может оказаться мало. Поэтому следующим шагом в развитии функциональной клавиатуры было введение возможности использования каждой клавиши обычной клавиатуры в качестве функциональной.

Рассмотрим это на примере микроЭВМ ДВК. Микропроцессор во время набора символов на клавиатуре постоянно работает. При этом код каждой нажатой клавиши сразу поступает в диалоговую систему. В ДВК имеется возможность управлять отображением данной

клавиши, т.е. если код клавиши поступает в систему, то она может вывести этот код на экран с помощью команды вывода. Таким образом, клавиатура в ДВК является только устройством ввода информации, а вывод осуществляется программным способом. Очевидно, что в этом случае каждая клавиша может быть функциональной.

Из рассмотренного можно сделать следующие выводы:

1) функциональная клавиатура позволяет существенно повысить производительность ручного ввода;

2) использование функциональных клавиш позволяет упростить работу с диалоговой системой путем замены многих директив.

3.2. Сервисные возможности

В настоящее время все операционные и диалоговые системы содержат большое число программ сервисного назначения. Сервисные программы возникли на ранних стадиях развития вычислительной техники и сейчас представляют собой самую развитую и представительную часть программного обеспечения.

Рассмотрим некоторые примеры диалоговых сервисных возможностей как для профессионального использования (низший уровень), так и для широкого круга пользователей (высший уровень). Первоначально сервисные программы создавались профессиональными программистами для получения всевозможных справок о текущем состоянии вычислительной системы, об автоматизации наиболее часто используемых операций, что существенно повышало эффективность вычислительного процесса. Это был низший уровень сервисных программ. Постепенно программы модернизировались и объединялись в развитые комплексы сервисного программного обеспечения, приобретая при этом все более "интеллектуальный" характер. В настоящее время они достигли такого уровня, что стали доступны непрофессиональным пользователям.

Рассмотрим н и з ш и й у р о в е н ь сервисного программного обеспечения. При работе на вычислительной машине профессионального программиста часто интересует содержимое регистров и памяти машины как оперативной, так и внешней. Такая информация называется дампом

```
Адрес:                  Дисассемблер:                 |    CVT - 00C360
00C896 90DE 02D0        STM   13,14,704               |   SCVT - 00C5B0
00C89A 05D0             BALR  13,0                    | SVCTBL - 01ABBC
00C89C 9180 0360        TM    864,128                 |   JSCB - C0F4D4
00C8A0 4710 D09A        BC    1,154(13)               |   CSCB - FF68CB
00C8A4 58E0 0010        L     14,16                   | FOXTCB - C0E4C8
00C8A8 B611 02C8        DC    X'B61102C8'             | FXLIST - 059CB0
00C8AC D503 02C8 E188   CLC   712(4,0),392(14)       | FOXPDA - 08A000
00C8B2 4780 D01E        BC    8,30(13)                | FOXBFO - 08A174

Адрес:  Дамп оперативной памяти:
000000  000C0000 0000879C 00000000 00000000  * .....gm..... *
000010  0000C360 00000000 070E0000 00000000  * ..C-..........- *
000020  070D3000 0006203C 070C1000 00FB7066  * ..............ш.. *
000030  040C0000 00004560 060C0000 0000415A  * .......-.......] *
000040  00C0BB68 0D000001 00006BC0 0000C360  * ..Ц...........C- *
000050  F4D38DFF 00055C94 040C0000 0000C692  * 4Lг...*m......Fk *
000060  040C0000 0000D212 000C0000 0000C966  * .......K.......I. *
000070  00000000 00035030 040C0000 0000CB96  * ......&.......Ho *

===>
```

Рис. 3.1

памяти. Иными словами дамп — это распечатка или вывод на экран дисплея содержимого памяти и (или) регистров.

Как правило, дамп выдается на экран дисплея или печатающее устройство в шестнадцатеричном и/или символьном виде с указанием соответствующих шестнадцатеричных адресов памяти. На рис. 3.1 представлен пример дампа оперативной памяти, выданный соответстующей программой (автор Ефремов С.В.) в системе "Фокус".

Режим дампа оперативной памяти реализован по принципу оконной технологии, применяемой в настоящее время во многих современных системах. Этот режим содержит пять логически независимых информационных окон, каждое из которых имеет определенное расположение на экране и следующее назначение:

— окно ДАМП (в нижней части экрана) используется для просмотра, поиска и корректировки содержимого оперативной памяти;

— окно ДИСАССЕМБЛЕР используется для просмотра загруженных в память программ в формате команд языка ассемблера;

— окно СИСТЕМА (в правом верхнем углу) содержит адреса часто используемых системных таблиц;

— окно ABEND содержит информацию об аварийном завершении приватной задачи системы "Фокус". Используется только при аварийных завершениях (на рисунке отсутствует);

— командное окно (на предпоследней строке) служит для ввода управляющих команд.

Последняя строка экрана является информационной, куда выводятся, например, сообщения об ошибках.

При работе с программой используются следующие команды управления:

D_addr — выдать дамп памяти, начиная с адреса addr. Адрес задается шестнадцатеричным числом (1-6 шестнадцатеричных цифр). Если адрес не указан, то принимается нулевое зачение. Если в качестве адреса указан символ "R", то принимается значение сумматора (см. ниже);

A_addr — выдать ассемблерный код, начиная с адреса addr. Команда A без параметров служит для активизации окна ДИСАССЕМБЛЕР. В качестве адреса может быть указан символ "R" (значение сумматора);

B — активизировать окно ABEND. Окно ABEND может быть активно только при аварийном завершении приватной задачи системы "Фокус";

S_key — искать данные по ключу. По команде S осуществляется поиск данных, указанных в параметре key, начиная с текущего адреса, увеличенного на единицу. Параметр key задается в виде списка значений, разделенных пробелом. Каждое значение представляет собой символьную строку, заключенную в апострофы, или байт, заданный двумя шестнадцатеричными цифрами, например, S_01_02_'DUMP'_03_FF. Команда S без параметров продолжает поиск по ключу от предыдущего запроса.

Нажатие клавиши ВВОД без модификации каких-либо полей экрана вызывает выдачу дампа памяти или ассемблерного кода в зависимости от положения курсора на экране. Если клавиша ВВОД нажата в поле адреса окна ДИСАССЕМБЛЕР, то будет выдан ассемблерный код, иначе будет выдан дамп памяти. Начальный адрес определяется от позиции курсора на экране до первого символа, отличного от шестнадцатеричной цифры.

Каждое окно содержит незащищенные поля, которые можно модифицировать путем ввода информации с клавиатуры дисплея. Модификация поля адреса вызывает выдачу информации в окне, в котором данное поле расположено. Начальный адрес определяется значением модифицированного поля.

Замена содержимого оперативной памяти производится непосредственным внесением информации в поле данных окна ДАМП с последующим нажатием клавиши ВВОД. За одно нажатие на клавишу ВВОД можно заменить до 4 байт шестнадцатеричного представления или до 16 байт символьного представления данных.

В режиме дампа оперативной памяти функциональные клавиши имеют следующие назначения:

ПФ1 — получение справки;
ПФ3 — установка курсора в 1-ю позицию экрана;
ПФ4 — листание кодов назад;
ПФ5 — листание кодов вперед;
ПФ7 — листание дампа назад на 128 байт;
ПФ8 — листание дампа вперед на 128 байт;
ПФ10 — листание дампа назад на 16 байт;
ПФ11 — листание дампа вперед на 16 байт;
ПФ12 — переключение между окнами;

СТРН.ЭКР — выход.

В режим дампа памяти встроен вычислитель, позволяющий выполнять четыре основные арифметические операции над двумя целыми числами (+, −, *, /). Число может иметь одно из трех форматов представления: десятичный, шестнадцатеричный, десятичный, выраженный в килобайтах. В шестнадцатеричном формате последним символом должен быть указан символ H, а в числе, выраженном в килобайтах — символ K. Например: 8AH, 128K. Результат вычисления выдается в трех форматах, а также помещается в специальный сумматор R, содержимое которого можно использовать при последующих вычислениях или в качестве параметра в командах A и D. Для операции деления дополнительно в круглых скобках выдается частное от деления.

```
     *****    Т А Б Л И Ц А    В Е К Т О Р О В   С В Я З И   ( C V T      **********
-008  FFB  01CD2C   0000                ЗАРЕЗЕРВИРОВАНО
-006  FFA  01CD2E   1061        CVTMDL   ШИФР МОДЕЛИ ЭВМ
-004  FFC  01CD30     61 _ _ _  CVTRELNO  НОМЕР ИЗДАНИЯ ОС ЕС
0000  000  01CD34   00012290    CVTTCBP   АДРЕС ДВОЙНОГО СЛОВА СО СЛЕД.И ТЕК.TCB
0004  004  01CD38   00011888    CVTOEF00  АДРЕС ПРОГ.ПЛАН.АСИНХР.ВЫХОДОВ
0008  008  01CD3C   0001CC14    CVTLINK   АДРЕС DCB ДЛЯ НАБОРА ДАННЫХ SYS1.LINKLIB
0012  00C  01CD40   0002D8A0    CVTJOB    АДРЕС БЛОКОВ УПРАВЛЕНИЯ РАБОЧЕЙ ОЧЕРЕДЬЮ
0016  010  01CD44   00000000    CVTBUF    АДРЕС БУФЕРА РЕЗ.ПРОГ.ПРЕРЫВАНИЙ С КОНСОЛИ
0020  014  01CD48   00008CFC    CVTXAPG   АДРЕС ТАБЛ. ДОПОЛНЕНИЙ СУПЕРВИЗОРА В/В
0024  018  01CD4C   00011FFA    CVTOVL00  АДРЕС ПРОГ. ПРОВЕРКИ КОРРЕКТНОСТИ АДРЕСОВ
0028  01C  01CD50   0001EAF6    CVTPCNVT  АДРЕС ПРОГ. ПРЕОБРАЗОВАНИЯ TTR В MBBCCHHR
0032  020  01CD54   0001EA70    CVTPRLTV  АДРЕС ПРОГ. ПРЕОБРАЗОВАНИЯ MBBCCHHR В TTR
0036  024  01CD58   00009066    CVTILK1   АДРЕС ЧАСТИ (КАНАЛ) И БЛОКОВ В ТАБЛИЦЕ UCB
0040  028  01CD5C   000090E6    CVTILK2   АДРЕС ЧАСТИ СО СПИС. АДРЕСОВ UCB В ТАБЛ.UCB
0044  02C  01CD60   00011E4E    CVTXTLER  АДРЕС ПРОГ. XCTL ДЛЯ ОБРАБОТКИ ОШИБОК
0048  030  01CD64   000023B0    CVTSYSAD  АДРЕС ВХОДА В ТАБЛ.UCB ДЛЯ РЕЗ.СИСТЕМЫ
0052  034  01CD68   00010620    CVTBTERM  АДРЕС ТОЧКИ ВХОДА В ПРОГРАММУ ABTERM
0056  038  01CD6C   00861 58F   CVTDATE   ТЕКУЩАЯ ДАТА В УПАКОВАННОМ ФОРМАТЕ
0060  03C  01CD70   0002D938    CVTMSLT   АДРЕС РЕЗ.ОБЛАСТИ ГЛАВНОГО ПЛАНИРОВЩИКА
0064  040  01CD74   0001EBCC    CVTZDTAB  АДРЕС ТАБЛ.ХАРАКТЕРИСТИК УСТРОЙСТВ В/В
0068  044  01CD78   0002DA00    CVTXITP   АДРЕС ПРОГ.ИНТЕРПРЕТАТОРА ОШИБОК
0072  048  01CD7C   00041660    CVTDAR    АДРЕС БЛОКОВ УПРАВЛЕНИЯ В/В НЕИСПРАВН.DAR
```

Рис. 3.2

Запрос для вычислителя вводится в командной строке. Первым символом запроса должна быть десятичная цифра или символ R. Представленный вычислитель позволяет не только выполнять арифметические вычисления, но и переводить числа из одного формата представления в другой, что довольно часто требуется при работе с дампом оперативной памяти.

Информация в памяти имеет определенную логическую организацию, поэтому развитые системы обладают возможностью получить на экране дисплея содержимое управляющих таблиц в наиболее удобном для работы виде (с указанием символических имен и описаниями полей, которые представлены в соответствующей документации). Такая информация называется *форматизованным дампом.* На рис. 3.2 показан пример форматизованного дампа одной из основных таблиц связей операционной системы ЭВМ.

При работе с дампом памяти оперируют с данными в шестнадцатеричном, двоичном, десятичном и символьном представлениях, поэтому разработаны специальные команды для соответствующего перевода данных из одного представления в другое, а также для вычисления некоторых арифметических операций над кодами.

Дамп внешней памяти еще более связан с логической организацией данных на магнитных носителях. Рассмотрим иерархическую схему работы с дампом внешней памяти. На самом верхнем уровне система выдает на экран дисплея перечень томов подключенных магнитных носителей (диски и ленты). Выбирая конкретное имя тома, пользователь переводит систему на следующий уровень просмотра. Для магнитных дисков — это просмотр оглавления тома. Информация, выдаваемая при этом, содержит для каждого набора данных: имя, организацию, формат записей, размер физических блоков и длину логических записей, процент заполнения и т.д. На рис. 3.3 показан пример вывода на экран оглавления тома. На рассматриваемом уровне имеется достаточно большой набор команд, предназначенных для создания, переименования и удаления наборов данных. На следующем уровне просмотра, например для наборов данных с библиотечной организацией, на экран выводится справочник с именами разделов библиотеки, т.е. оглавление. Здесь также существует развитый набор команд, обеспечивающих управление просмотром, переименование и удаление разделов и др. Последний уровень просмотра — это вывод на экран из разделов библиотек и других наборов данных.

	ИМЯ НАБОРА ДАННЫХ	ORG	RECF	BLKSZ	LRECL	EXT	CL+TR	%%%
*	SYS1.SYSJOBQE	PS	U	00000	00000	1	20+00	100
*	SYS1.SVCLIB	PO	U	13030	00000	1	13+00	64
*	SYS1.LINKLIB	PO	U	13030	00000	1	29+00	85
*	SYSCTLG	UN	UN	00000	00000	1	00+02	0
*	SYS0.PROCLIB	PO	FB	03360	00080	1	01+16	35
*	SYS1.MACLIB	PO	FB	03360	00080	2	21+13	100
*	SYS1.PARMLIB	PO	F	00080	00080	1	00+12	54
*	SYS1.PL1LIB	PO	U	13030	00000	1	01+18	72
*	SYS1.SORTLIB	PO	U	13030	00000	1	01+14	65
*	SYS1.COBLIB	PO	U	13030	00000	1	00+08	57
*	SYS1.FORTLIB	PO	U	13030	00000	1	02+04	60
*	SYS1.TELCMLIB	PO	U	13030	00000	1	01+05	69
*	SYS1.ALGLIB	PO	U	13030	00000	1	00+07	66
*	SYS1.DCMLIB	PO	U	13030	00000	1	00+05	50
*	SYS1.SYSVLOGX	PS	VB	00888	00130	1	01+01	0
*	SYS1.SYSVLOGY	PS	VB	00888	00130	1	01+01	15
*	SYS1.MANX	PS	VBS	00544	00544	1	03+03	81
*	SYS1.MANY	PS	VBS	00544	00544	1	03+03	0
*	SYS1.OLDNUC	PO	U	13030	00000	1	02+08	68
*	SYS1.ROLLOUT	PS	F	00000	00000	1	35+10	100
*	SYS1.GTFGO	PO	U	13030	00000	1	02+12	6
*	SYS1.TRACE	UN	UN	00000	00000	1	01+01	0

Рис. 3.3

Программы, позволяющие просматривать и корректировать данные с последовательной логической организацией, называются *текстовыми редакторами.* Такие программы относятся к высшему уровню сервисного обеспечения.

Рассмотрим один из наиболее развитых представителей профессионального сервисного обеспечения — средство отладки программ или отладчик.

Отладчик — это интерактивный модуль программного обеспечения, который используется для управления выполнением программ с целью выявления и локализации в них ошибок. Отладчик имеет команды запуска и останова выполнения программы с конкретных точек, просмотра и изменения содержимого памяти и регистров.

Чтобы отладчик начал работу, пользователь должен указать имя необходимого задания, имя отлаживаемого в нем модуля и один или несколько адресов останова. После ввода этой информации отладчик выходит в состояние ожидания. Когда в системе появится указанное задание и будет найден требуемый модуль, отладчик остановит выполнение этого модуля в указанной точке останова, если, конечно, логика отлаживаемого модуля предусматривает проход через эту точку. После того как выполнение отлаживаемого модуля остановлено, на экран дисплея выдается следующая информация:

— имя отлаживаемого модуля;
— адрес точки входа и адрес точки загрузки модуля;
— адрес точки останова;
— слово состояния отлаживаемого модуля;
— содержимое основных регистров в точке останова;
— адреса основных блоков управления задачей.

Дальнейшая работа в отладчике осуществляется командами управления дампом памяти и отладчиком. Наиболее существенными являются команды пошагового выполнения отлаживаемого модуля, изменения содержимого общих регистров, установки новых адресов останова.

Для большего пояснения концепции отладчика приведем минимальный набор команд отладчика, используемый в микрокомпьютерах:

Команда	Параметры	Назначение
LOAD	Имя программы	Загрузить отлаживаемую программу в память компьютера
SET_REG	Регистр, значение	Записать в указанный регистр указанное значение

SET_MEM	Адрес, значение	Записать в память по указанному адресу указанное значение
DISPLAY_REG	Регистр	Выдать содержимое указанного регистра
DISPLAY_MEM	Адрес	Выдать содержимое из указанного адреса памяти
RUN		Начать или продолжить выполнение программы
SET_BREAKPOINT	Адрес	Установить адрес точки останова. Отладчик отмечает указанный адрес таким образом, что при попытке выполнить команду, расположенную по этому адресу, управление будет передано в отладчик, а отлаживаемая программа остановлена
CLEAR_BREAKPOINTS		Отменить адреса точек прерывания. Все адреса, отмеченные ранее командами SET BREAKPOINT, освобождаются

В реальных программах отладчиков допускаются как полные, так и сокращенные мнемонические коды команд.

Кроме перечисленных сервисных средств, следует упомянуть также различные программы копирования информации из одних наборов данных в другие. Такие программы копирования помогают считывать и переписывать данные на различные носители информации.

Сервисные программы высшего уровня возникли в результате естественного процесса эволюции программного обеспечения и стали понятны и доступны пользователям, не являющимся профессиональными программистами.

Текстовые редакторы. Наиболее распространенными сервисными операциями являются: подготовка, хранение и редактирование различных текстов. В этих применениях ЭВМ, и в особенности микроЭВМ, пришли на смену пишущим машинкам. Текстовые редакторы разработаны для получения максимальной гибкости при корректировке текстов. Работу с текстовыми редакторами легко освоить,

так как они рассчитаны на широкий круг пользователей и могут быть полезны ученым, писателям, журналистам и людям многих других профессий в тех случаях, когда необходимы пишущие машинки. Текстовой редактор и пишущая машинка функционально очень похожи, хотя текстовой редактор обладает большими преимуществами.

Одно из преимуществ — простота исправления ошибок. Так как набираемый текст не печатается непосредственно на бумаге, можно делать любые исправления. Распечатать текст можно после внесения всех исправлений при уверенности в корректности текста. Дополнительным сервисным средством при этом является программная проверка синтаксиса подготавливаемого текста.

Другое преимущество текстовых редакторов — автоматическое выравнивание правой границы текста. При работе на пишущей машинке для завершения каждой строки используется клавиша НОВАЯ СТРОКА. Текстовой редактор выполняет эту функцию автоматически. Как только вводимый с помощью клавиатуры текст достигает конца строки, происходит переход на новую строку. А после нажатия клавиши ВВОД текстовой редактор выполняет выравнивание текста в соответствии с заданными параметрами выравнивания. Эти параметры могут быть изменены пользователем в зависимости от необходимых требований к оформлению текста.

При работе с текстовым редактором всегда можно запросить помощь у системы, введя соответствующую команду.

Рис. 3.4

Рассмотрим пример текстового редактора, реализованного в системе "Фокус" и имеющего название базового редактора.

Базовый редактор имеет общецелевое назначение и применяется для просмотра и корректировки файлов. Экран базового редактора разделен на три зоны: информационную, управляющую и рабочую (см. рис.3.4).

Информационная зона находится в верхней части экрана. В нее входят 2 строки: информационная и разметочная. Любая из этих строк или обе

строки могут быть исключены из зоны по специальной команде. В этом случае управляющая и рабочая зоны увеличатся на одну или две строки.

Информационная строка содержит определители позиций управляющей зоны "*>( )+>-!", идентификацию файла, вызванного в базовый редактор, объем файла в записях "О=...", номера граничных строк выделенного фрагмента "Ф=(...,...)", ключевое слово команд поиска "К=...", символ пробела и символы схлопывания "Р=...", номер файла для режима работы с двумя файлами.

Разметочная строка содержит определители позиций управляющей зоны "*>( )+>-!" и разметку для рабочей зоны.

Управляющая зона расположена в левой части экрана и содержит семизначные номера записей, находящихся в рабочей зоне базового редактора. При нажатии клавиши ВВОД при курсоре, установленном в любую из позиций управляющей зоны, происходит определенное действие по отношению к записям файла.

Рабочая зона содержит записи файла, вызванного в базовый редактор. Нумерация в управляющей зоне и разметка в информационной зоне определяют, какие именно записи находятся в рабочей зоне.

Нажатие клавиши ВВОД при курсоре, установленном в определенную позицию экрана, для краткости, будем называть "ввод в позиции". Термин "запись" имеет отношение к файлу, термин "строка" — к редактору.

Управление из информационной зоны. Ввод в первых восьми позициях информационной строки приводит к удалению этой строки из информационной зоны. Ввод в остальных позициях инициирует запрос на установку ключевого слова для команд поиска. Ввод в первых восьми позициях разметочной строки, отмеченных символами ">( )+>-!", исключает разметочную строку из информационной зоны. Ввод в позициях разметки приводит к сдвигу строки разметки влево так, что указанная позиция становится первой. Вместе с этим происходит сдвиг записей файла в рабочей зоне. Таким образом просматриваются и корректируются файлы с длиной записи до 255 байт. Последняя позиция записи отмечается в строке разметки звездочкой "*". Если нажат ВВОД в этой позиции или правее ее, то произойдет начальная установка разметки, т.е. в начале будет первая позиция.

Позиционное управление из управляющей зоны. *Ввод в первой позиции* "*" любой строки в зоне кроме первой выводит указанную запись файла на первую строку. Таким образом осуществляется листание записей файла вперед с любым шагом в пределах экрана. Ввод в первой позиции первой строки осуществляет листание назад. Если на экране находится последняя запись файла, а ВВОД нажат в первой позиции ниже этой записи, то происходит добавление пустых записей в конец файла. С помощью функциональных клавиш ПФ7 и ПФ8 можно выполнять постраничное листание файла назад и вперед соответственно. *Ввод во второй позиции* ">" приводит к установке перед указанной записью командной строки, в которой можно задавать сложные запросы обработки файла. *Ввод в третьей позиции* "(" объявляет указанную запись "началом фрагмента". Вместо номера записи высвечивается слово "НАЧАЛО", а номер записи заносится в информационную строку в поле индикатора фрагмента. Повторный ввод в этом же месте отменяет начало фрагмента. *Ввод в четвертой позиции* ")" определяет "конец фрагмента". При этом вместо номера записи появляется слово "КОНЕЦ", а номер записи заносится в индикатор фрагмента в информационной строке. Повторный ввод в этом же месте отменяет конец фрагмента. Фрагмент текста считается выделенным, если определены его начало и конец. При совпадении этих значений фрагментом считается одна строка. Визуально фрагмент выделяется повышенной яркостью (на дисплеях ЕС-7927). *Ввод в пятой позиции* "+" осуществляет вставку копии фрагмента перед указанной строкой.

Если фрагмент отсутствует или отключен, то происходит дублирование записи на указанной строке. *Ввод в шестой позиции* "—" (аналогично второй позиции) приводит к установке командной строки для сложных запроосов. *Ввод в седьмой позиции* "—" в строке начала или конца фрагмента приводит к удалению фрагмента, в любой другой строке — к удалению записи файла на указанной строке. *Ввод в восьмой позиции* "!" осуществляет замену информации файла в рабочей зоне и удаление записи файла на указанной строке.

Командное управление из управляющей зоны. Команды в управляющей зоне указываются в начале любой строки. После набора команды должно непосредственно следовать нажатие клавиши ВВОД. Команда В выводит указанную запись файла на первую строку рабочей зоны. Команда Н устанавливает начало фрагмента. При этом вместо номера записи появляется слово НАЧАЛО, а номер записи заносится в индикатор фрагмента в информационной строке. Команда К устанавливает конец фрагмента. При этом вместо номера записи появляется слово КОНЕЦ, а номер записи заносится в индикатор фрагмента в информационной строке. Если конец фрагмента уже устаювлен, или не было установлено начало фрагмента, то осуществляется выход из базового редактора. Команда Д осуществляет вставку копии фрагмента перед указанной строкой. Если фрагмент отсутствует или отключен, то происходит дублирование записи на указанной строке. Команда У, введенная в строке начала или конца фрагмента, приводит к удалению всего фрагмента. В противном случае происходит удаление записи файла на указанной строке. Допускается перед нажатием клавиши ВВОД указать несколько команд У в последовательных строках, в этом случае происходит одновременное удаление записей на всех указанных строках. Команда Ф отключает выделенный фрагмент файла. Две команды П осуществляют перестановку записей на помеченных строках. Команда _ (пробел) вставляет в указанное место файла запись, состоящую из пробелов. Команда С вставляет в указанное место специальную запись для комментария. (Прим няется для оформления текстов программ). Команда S вставляет в указанное место специальную запись, в которую при копировании в библиотеку система заносит текущую дату и время. Команда ? восстанавливает на экране информационную зону, если последняя была ранее удалена. При наличии информационной зоны по команде ? на экран выдается справочная информация о командах базового редактора. Команда Р переключает (ВКЛ-ВЫКЛ) режим "схлопывания" текста. Этот режим устанавливается запросом из командной строки. Команда Т переключает режим табуляции. Табулирование устанавливается запросом из командной строки. Команда А выводит в рабочую зону первые записи файла. Команда Я выводит в рабочую зону последние записи файла. Команда = переключает базовый редактор в режим текстового редактора и обратно. Текстовый редактор осуществляет дополнительную обработку файла, выполняя при этом выравнивание текста по правому краю с грамматическим переносом слов и равномерное распределение пробелов. Позиция выравнивания устанавливается запросом в командной строке и выделяется в разметочной строке знаком =. TXT-файлы попадают при вызове в текстовой редактор автоматически. Позиция выравнивания по умолчанию устанавливается равной 60. Команда L включает и выключает "листинговый" (просмотровый) редактор. Листинговый редактор служит обычно для просмотра результатов выполнения заданий. В листинговом редакторе зона управления не отображается, однако управление из этой зоны возможно. LST-файлы при вызове попадают в листинговый редактор автоматически, при этом устанавливается ключ поиска "ERROR". Команда Э включает экранный редактор (для файлов с длиной записей, равной 80). Команда Х выполняет переключение базового редактора с файла 1 на файл 2 и обратно. Индикация осуществляется в конце информационной строки. Данная команда предостав-

ляет возможность одновременной работы с двумя файлами. Команды И, ИВ, ИН осуществляют поиск ключевого слова, установленного ранее по запросу из командной строки или из информационной зоны. Мнемоника этих команд предполагает следующие операции: И, ИВ — "искать вперед", начиная со следующей записи в направлении конца файла; ИН — "искать назад", начиная с предыдущей записи в направлении начала файла. Команды М, МВ, МН аналогичны командам И, ИВ, ИН, но поиск осуществляется в поле метки для текстов программ на языке Ассемблера. При успешном поиске в рабочую зону выдается фрагмент текста так, что на строке, где была введена команда, помещается запись с найденным ключевым словом. Команда, представляющая собой любой номер (без лидирующих нулей), выводит в рабочую зону записи в соответствии с указанным номером. В этом смысле команда 1 аналогична команде А, а любой номер, больший, чем число записей в файле, аналогичен команде Я. Команды ПЧ формируют запрос на вывод файла на устройство печати.

Управление из рабочей зоны. Нажатие клавиши ВВОД после пробела приводит к записи содержимого рабочей зоны в файл, т.е. происходит замена текста. Если клавиша ВВОД нажата после символа, отличного от пробела, осуществляется замена текста и вставка следующей строки из пробелов. Курсор устанавливается в начало этой новой строки. При включенной табуляции нажатие клавиши ВВОД вызывает замену текста и установку курсора в следующую позицию табуляции. Вставка строки из пробелов не происходит.

Управление запросами из командной строки. Установка командной строки осуществляется вводом во второй или шестой позиции (">") любой строки управляющей зоны. Установка разделителей, т.е. изменение символа пробела и символов "схлопывания" осуществляется запросом РПНОК, где: Р — команда (обозначает установку разделителей), П — символ пробела, Н — символ начала группового схлопывания, О — символ одиночного схлопывания, К — символ конца группового схлопывания. Установка управляющих символов по умолчанию соответствует запросу "Р [^]", в чем можно убедиться, введя команду Р в управляющей зоне. Пользователь может изменить разделители, например, по запросу "Р.(@)". При этом все пробелы в тексте отображаются на экране точкой, а части строк, ограниченные круглыми скобками, а также одиночные символы "@", "схлопываются" справа налево. Установка позиций табуляции выполняется по запросу Т, в котором указывается список позиций в возрастающем порядке, например, "Т 1 10 16 40". Символ "Т" и номера позиций в запросе разделяются пробелами. Автоматически устанавливается табуляция для ASM-файлов, FTN-файлов и TXT-файлов. Установка позиции выравнивания в текстовом редакторе осуществляется запросом "=NN", где NN — номер позиции выравнивания. Для TXT-файлов позиция выравнивания устанавливается равной 60 по умолчанию. Операции поиска осуществляются с помощью запросов: ИВ/Что искать/ — искать вперед (к концу файла); И/Что искать/ — искать вперед (к концу файла); ИН/Что искать/ — искать назад (к началу файла). После выполнения любой операции поиска ключевое слово запоминается и высвечивается в информационной строке. Последующие операции поиска по тому же ключу можно выполнять командами из управляющей зоны, т.е. без установки командной строки. Аналогично выполняются операции поиска метки для ASM-файлов: МВ/Что искать/ — искать вперед, М/Что искать/ — искать вперед, МН/Что искать/ — искать назад. Для выполнения операций подстановки нового значения вместо старого используются запросы: ПВ/старое/новое/ — подстановка вперед, ПН/старое/новое/ — подстановка назад, ПФ/старое/новое/ — подстановка внутри фрагмента. В качестве символов разделителей, используемых в запросах поиска и подстановки ("/"), могут

быть любые отображаемые символы, в том числе алфавитно-цифровые. Важно лишь, чтобы они были одинаковыми в одном запросе. Вывод на печать всего файла или выделенного фрагмента инициируется запросом "ПЧ/CUU", где CUU — адрес устройства. В запросе можно опускать адрес устройства. Работа с двумя файлами обеспечивается подключением второго файла с помощью запроса "X файл". В конце информационной строки имеется индикация активного файла (цифра 1 или 2). Переключение файлов выполняется командой X из управляющей зоны. При работе с двумя файлами возможно копирование информации из одного файла в другой. Для этого служит запрос COPY. Если в копируемом файле выделен фрагмент, то копируется только информация из фрагмента, в противном случае копируется весь файл. Копирование осуществляется в файл между строками, где установлена командная строка. С помощью команд OVER и ROLL осуществляется наложение второго файла на первый, причем команда ROLL делает это многократно по всей длине первого файла.

Режим текстового редактора представляет собой расширение базового редактора для специальной обработки файлов с текстовой информацией. Специальная обработка заключается в выравнивании текста по правому краю, при этом выполняется грамматический перенос слов и равномерное распределение пробелов. Выравнивание текста осуществляется только для записей с первым символом, отличным от пробела или записей, содержащих информацию за позицией выравнивания. Позиция выравнивания устанавливается запросом в командной строке (например, "=52") и выделяется в разметочной строке знаком "=". По умолчанию в системе установлено выравнивание в позиции 60. Ввод команды "=" в управляющей зоне переключает базовый редактор в режим текстового редактора. После выполнения одного выравнивания (в пределах экрана и до нового абзаца за экраном) происходит автоматическое переключение в базовый редактор. Постоянно режим текстового редактора устанавливается только для TXT-файлов. Текстовой редактор допускает любое управляющее воздействие, принятое для базового редактора, за исключением управления из рабочей зоны, которое предназначено для инициирования выравнивания.

Большую помощь при работе с текстами оказывают синтаксические анализаторы и табличные калькуляторы.

Синтаксические анализаторы представляют собой достаточно большие по объему словари и программы поиска и сравнения слов исходного текста с элементами словаря. В результате такого сравнения слова текста, написанные с ошибкой, не будут найдены в словаре и пользователь сможет быстро исправить их. Если же слово написано правильно, но в словаре не найдено, то, выполнив соответствующие операции, следует добавить такое слово в словарь синтаксического анализатора.

Табличный калькулятор — это "интеллектуальная" таблица, применяемая в планировании, инженерных расчетах, научной деятельности, управлении, финансовых отчетах и во многих других областях деятельности человека. Рабочая область табличного калькулятора —

двухмерное табулированное пространство, разделенное, например, на 255 рядов и 64 колонки. На экран дисплея выводится только часть такого пространства, называемая окном. Используя специальные команды управления окном, можно перемещать окно в пространстве калькулятора. Все колонки имеют буквенные идентификаторы, ряды — цифровые. В пересечении рядов и колонок находится более 16000 ячеек. Ячейки используются для хранения числовых величин или символьных строк. "Интеллектуальность" табличного калькулятора заключается в использовании правил и формул, которые связывают различные блоки, ряды, колонки или конкретные ячейки таблицы. Это значит, что информация, введенная в одну область, автоматически преобразуется и устанавливает значения в других областях таблицы. Например, можно использовать 12 колонок таблицы для представления месяцев года и затем ввести данные по рядам в соответствии со статьями расхода. Следующие два ряда могут содержать формулы для подсчета, например, суммарных величин или процентных соотношений. Эти значения подсчитываются автоматически при введении новых чисел в таблицу. Дополнительную сервисную возможность представляют собой табличные калькуляторы, обеспечивающие графическое представление результирующей информации в виде гистограмм.

Кроме перечисленных сервисных возможностей большой интерес представляют различные архивные системы, информационные массивы, базы данных, на основе которых строятся информационно-поисковые системы типа картотек, записных книжек, справочников.

Дальнейшее развитие сервисных средств приведет к объединению разрозненных сервисных программ в единые комплексы, доступные широкому кругу пользователей.

3.3. Примеры диалога

Проиллюстрируем общение пользователя с ЭВМ на примере диалоговой системы "Фокус", использующей некоторые типы диалога из описанных в п. 2.3.

Для того чтобы начать работу, пользователь должен "войти" в систему, т.е. ввести индивидуальный код (обычно фамилию) и пароль. Для этого на соответствующий запрос

системы пользователь набирает на клавиатуре свою фамилию и с помощью клавиши ВВОД вводит ее в систему. После ввода фамилии система проверяет, во-первых, наличие ее в регистрационном списке, и во-вторых, не работает ли уже пользователь под этой фамилией в системе. Фамилии пользователей обычно известны, поэтому дополнительно вводится индивидуальный пароль, который защищает фамилию пользователя в системе и запрещает пользователям работать под чужими фамилиями. Таким образом, осуществляется защита от несанкционированного доступа и контроль действий пользователей в системе по статической и учетной информации. Пароль вводится в неотображаемом поле. Для ввода пароля обычно предоставляются две попытки.

После входа в систему пользователю на экран дисплея выдаются общие системные объявления и персональные "письма" от других пользователей. Общие системные объявления информируют об изменениях в системе и о появлении новых функциональных программ, а возможно, и о событиях, не имеющих отношения к системе "Фокус" (объявления о собраниях, семинарах, знаменательных датах и пр.). Персональные "письма" позволяют обмениваться информацией пользователям, работающим в различных сеансах.

Процедура входа в систему завершается выдачей на экран списка разделов системы (рис.3.5).

Выбор необходимого раздела представляет собой диалог типа "меню" и осуществляется вводом соответствующего идентификатора раздела, представленного номером в списке. Существует также возможность выбора с помощью фотоселектора или функциональной клавиши ПФ4, которая вызывает последовательное выделение повышенной яркостью отдельных элементов списка, представленного на экране. Если впоследствии нажать клавишу ВВОД, то произойдет переход в раздел, соответствующий элемент списка которого был выделен повышенной яркостью с помощью клавиши ПФ4. Предположим, что необходимо выполнить назначения функциональным клавишам. Такая возможность представлена в разделе "Сервисные программы и функции" (рис. 3.5). После выполнения ряда операций по выходу в режим резервирования функциональных клавиш на экран дисплея будет выдана информация, представленная на рис. 3.6.

```
                СПИСОК РАЗДЕЛОВ СИСТЕМЫ "ФОКУС"

1. РАЗДЕЛ ПРОГРАММИСТА                    9. ИГРЫ И ТЕСТЫ
2. ОБУЧАЮЩАЯ СИСТЕМА (АОС ВУЗ)           0. ПРОСТЕЙШИЙ РЕЖИМ РАБОТЫ
3. МАССИВЫ И БАЗЫ ДАННЫХ                  Д. ПРИМЕРЫ ДИФОЛ=ПРОГРАММ
4. СИСТЕМА ДОКУМЕНТАЦИИ (АСВД)            М. СВОБОДНОЕ МЕНЮ
5. ПОЧТА И ПЕРЕДАЧА СООБЩЕНИЙ             А. *.........
6. СЕРВИСНЫЕ ПРОГРАММЫ И ФУНКЦИИ          Б. *.........
7. КОНФИГУРАЦИЯ СИСТЕМЫ "ФОКУС"           Т. СРВ (TSO)
8. ДИСКУССИОННЫЙ КЛУБ                     К. КОНЕЦ РАБОТЫ

   НАБЕРИТЕ НОМЕР РАЗДЕЛА, НАЖМИТЕ ВВОД     >
```

Рис.3.5

```
ПФ1   УСТАНОВКА.SOM
ПФ2   СТИРАНИЕ.ЭКР.
ПФ3   КУРСОР,ДОМОЙ!
ПФ4   ВВОД-СЕЛЕКТОР
ПФ5   НОВАЯ.СТРОКА.
ПФ6   ВЕРХНИЙ.ЭКРАН
ПФ7   ПЕЧАТЬ.ЭКРАНА
ПФ8   ОТВЕТ."ДА"...
ПФ9   НИЖНИЙ.ЭКРАН.
ПФ10  СПРАВКА.(?)..
ПФ11  ОТВЕТ."НЕТ"..
ПФ12  ПОЛНЫЙ ЭКРАН
```

Рис.3.6

```
        ВЫБРАНА КЛАВИША - ПФ1                          КОДЫ ТИПА НАЗНАЧЕНИЙ:

        СТАНДАРТНОЕ НАЗНАЧЕНИЕ:                          ^ - ЗАПРОС
УСТАНОВКА СИМВОЛА НАЧАЛО СООБЩЕНИЯ (SOM)                 L - СЦЕПЛЕНИЕ СЛЕВА
                                                         R - СЦЕПЛЕНИЕ СПРАВА
        ТИП И ТЕКСТ ТЕКУЩЕГО НАЗНАЧЕНИЯ:                 D - ВЫВОД СООБЩЕНИЯ
                                                         P - ПРОГРАММА

УКАЖИТЕ ТИП И В СКОБКАХ НОВЫЙ ТЕКСТ НАЗНАЧЕНИЯ:

>
```

Рис.3.7

```
ПФ1   УСТАНОВКА,SOM     D START SIGNALH,V=FXDISK,D=MASTER,V=FXNFSTED
ПФ2   СТИРАНИЕ,ЭКР.     Q !!TESTTED
ПФ3   КУРСОР,ДОМОЙ!
ПФ4   ВВОД-СЕЛЕКТОР
ПФ5   НОВАЯ,СТРОКА.     P FXKEYPF5
ПФ6   ВЕРХНИЙ,ЭКРАН
ПФ7   ПЕЧАТЬ,ЭКРАНА     Q !P7934
ПФ8   ОТВЕТ,"ДА"...
ПФ9   НИЖНИЙ,ЭКРАН.
ПФ10  СПРАВКА,(?)..     Q !!?
ПФ11  ОТВЕТ,"НЕТ"..
ПФ12  ПОЛНЫЙ ЭКРАН
```

Рис.3.8

На рисунке обозначены: в левой колонке — наименования функциональных клавиш (ПФ1, ПФ2 и т.д.), в средней колонке — стандартные назначения системы "Фокус", в правой колонке — пользовательские назначения. При первом входе в данный раздел правая колонка не содержит информации. Для того чтобы выполнить назначение какой-либо клавише, необходимо нажать ее. В результате система предложит ввести информацию, которая состоит из типа назначения и текста назначения (рис. 3.7). Тип назначения определяет способ обработки текста назначения при нажатии соответствующей клавиши. Тип описывается следующими идентификаторами:

Q — запрос (текст назначения поступает на обработку без вывода на экран);

D — сообщение (текст назначения выводится на экран);

L — сцепление слева (текст назначения сцепляется с введенным запросом слева и поступает на обработку);

R — сцепление справа (текст назначения сцепляется с введенным запросом справа и поступает на обработку);

P — программа (система выполняет указанную программу при нажатии соответствующей клавиши).

После установки необходимых назначений правая колонка будет содержать соответствующие типы и тексты назначений (рис. 3.8). В процессе дальнейшей работы в системе "Фокус" можно использовать двойное назначение функциональных клавиш: стандартное и пользовательское. Для переключения используется клавиша ВЫЗОВ ТЕКСТА.

Вопросы для самопроверки

3.1. Какие устройства ввода данных в ЭВМ вам известны?

3.2. Чем отличается функциональная клавиатура от обычной?

3.3. Перечислите зарезервированные функциональные клавиши, применяемые в используемой вами диалоговой системе.

3.4. Используете ли вы в работе с диалоговой системой режим ввода с функциональной клавиатуры с подтверждением?

3.5. На рис. 3.1 показан пример дампа оперативной памяти. Определите, в какой колонке находится шестнадцатеричное представление, а в какой символьное. Где находятся шестнадцатеричные адреса?

3.6. Найдите на рис. 3.2 в первых четырех колонках шестнадцатеричные адреса и значения ячеек памяти, а также относительные адреса элементов таблицы.

3.7. Придумайте механизм работы отладчика для выполнения останова отлаживаемой программы в требуемой точке.

3.8. Какие типы диалога встречаются в параграфе 3.3?

Глава 4
СРЕДА РАЗВИТОГО ДИАЛОГА

4.1. Операционное окружение

В процессе решения задачи на ЭВМ пользователю требуются определенные ресурсы: аппаратные средства, программное обеспечение, данные.

Под *аппаратными средствами* понимаются: вычислительная система, включающая в себя центральный процессор, каналы, линии связи и периферийные устройства ЭВМ. На вычислительной системе функционирует программное обеспечение, осуществляющее обработку данных пользователя.

На первых стадиях применения вычислительной техники программист работал непосредственно за машиной и чаще всего пользовался ей единолично. Ресурсы вычислительной системы при этом использовались неэффективно, а часто вообще не использовались, так как не требовались в данный момент.

По мере внедрения вычислительной техники увеличилось число выпускаемых ЭВМ, но еще больше росло число программистов. Появилась проблема более эффективного использования ресурсов ЭВМ. Так возникла идея совместного использования ресурсов одной ЭВМ несколькими программистами. Очевидно, что при этом возникает проблема распределения и управления ресурсами. Выполнение этой задачи взяли на себя операционные системы (ОС).

Операционная система — совокупность программных средств, обеспечивающих оптимальную организацию использования ресурсов ЭВМ. Операционная система выступает в качестве посредника между программистом и ЭВМ. Для этого в состав ОС включаются программные

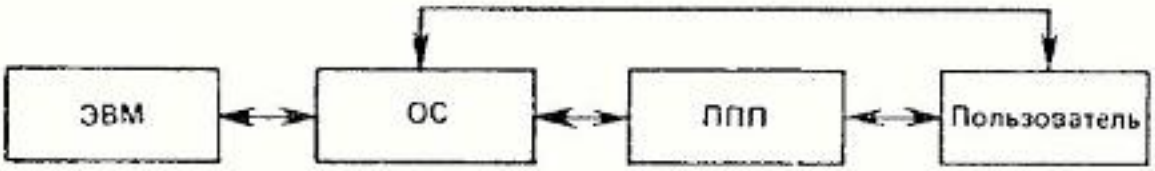

Рис. 4.1

средства общения с пользователем, которые обеспечивают требуемый уровень сервиса. Кроме средств общения с пользователем имеются программные средства, облегчающие доступ к вычислительной системе и повышающие производительность программиста, т.е. в конечном счете улучшающие уровень сервиса. Эти дополнительные средства называются *пакетами прикладных программ* (ППП). Они функционируют под управлением ОС и являются посредниками между ОС и пользователем.

В соответствии со схемой, представленной на рис. 4.1, пользователь может "контактировать" с ОС и без участия ППП. Примером ППП является диалоговая система, функционирующая под управлением ОС.

Операционная система в совокупности со средствами, обеспечивающими более эффективный доступ и использование ресурсов ЭВМ, составляет *операционное окружение* пользователя.

Взаимодействие с ОС можно подразделить на следующие уровни: команды, язык управления заданиями.

На уровне команд пользователь может запускать на выполнение определенные задания, получать информацию о текущем состоянии ОС и состоянии выполнения своего задания и изменять эти состояния.

Задание — это единица работы, которую должна выполнить операционная система по запросу пользователя. Примером задания является трансляция программы с алгоритмического языка. С помощью команды, заданной с дисплея, можно указать:

— какой транслятор нужно вызвать для выполнения;

— где находится исходный текст программы на алгоритмическом языке;

— куда помещать результаты и протокол трансляции.

Взаимодействие с ОС на уровне команд характерно для режима персонального доступа к ЭВМ (работа на персональном компьютере). В мультипрограммной ОС пользова-

управления машиной и операторскому пульту, но имеет возможность выполнять ограниченное число команд.

Рассмотрим функционирование мультипрограммной операционной системы для ЕС ЭВМ — ОС/ЕС. Операционная система ориентирована на пакетную обработку заданий пользователей. Пакет заданий формируется пользователем на перфокартах, перфоленте, магнитной ленте, а в данном случае — с дисплея. В состав ОС/ЕС входит ППП "Диалоговый удаленный ввод заданий", который обеспечивает ввод задания с клавиатуры, редактирование задания и запуск задания на выполнение. С точки зрения ОС/ЕС задание — это совокупность операторов языка управления заданиями (ЯУЗ) и данных.

Язык управления заданиями характерен для мультипрограммных операционных систем, функционирующих на ЭВМ третьего поколения в условиях конкурентного выполнения нескольких заданий. В этих условиях ЯУЗ берет на себя выполнение многих функций оператора ЭВМ. Основная цель применения ЯУЗ — автоматизация процесса выполнения заданий. Чтобы лучше себе представить необходимость языка управления заданиями, рассмотрим пример взаимодействия пользователя и ОС/ЕС.

Рассмотрим задачу пользователя. Предположим, что ему необходимо произвести расчеты с применением языка ФОРТРАН. Для этого нужно написать программу на языке ФОРТРАН, ввести ее с помощью клавиатуры и записать на диск. Но этого недостаточно с точки зрения ОС/ЕС. Необходимо оформить всю работу, требуемую от ОС в виде задания. Объясняется это тем, что ОС/ЕС выполняет одновременно несколько работ, заданных пользователем, поэтому для различения этих работ необходимо оформить каждую из них в виде задания. Целью настоящей книги не является подробное изложение какого-либо языка управления заданиями. Для этого существуют специальные руководства и методические пособия. Поэтому рассмотрим только функциональную часть операторов языка управления заданиями ОС/ЕС. Как отмечалось выше, задания — совокупность операторов ЯУЗ и данных пользователя. В качестве данных рассматривается вся информация пользователя. Для того чтобы отличать данные от операторов ЯУЗ, в начале карты с оператором ЯУЗ должны стоять символы //. Обратите внимание, что при работе с дисплее-

ем пользователь имеет дело не с самими перфокартами, а с их образами. Образ перфокарты соответствует одной строке дисплея.

К основным операторам языка управления заданиями относятся следующие три оператора:

1. Оператор JOB предназначен для идентификации задания. Задания идентифицируются с помощью имен заданий. Имя задания должно быть уникальным. Кроме имени в операторе JOB можно задавать:

— учетную информацию (необходимую для учета ресурсов, затраченных на выполнение данного задания);

— имя программиста (фамилия программиста; можно указать учебную группу — всего до 24 символов, заключенных в апострофы);

— режим печати управляющих операторов ЯУЗ в выходном листинге (можно задавать компактную или расширенную печать);

— приоритет выполнения (низший —0, высший — 13);

— время выполнения задания;

— объем требуемой для выполнения оперативной памяти;

— другие параметры.

Таким образом, в операторе JOB пользователь сообщает ОС о том, что нужно выполнить задание с указанным именем, при этом он дополнительно перечисляет параметры, которые влияют на процесс выполнения задания в ОС/ЕС. Оператор JOB идентифицирует задание для ОС/ЕС.

2. Оператор EXEC предназначен для определения имени программы, которая должна быть выполнена. Он отмечает начало каждого шага задания, соответствующего выполнению определенной в операторе EXEC программы. В данном примере следует указать системе имя программы — транслятора с языка ФОРТРАН.

Транслятор с языка ФОРТРАН входит в состав ОС/ЕС. Вызов транслятора оформляется в виде каталогизированной процедуры, которая представляет собой набор написанных заранее управляющих операторов языка управления заданиями.

Каталогизированные процедуры служат для облегчения работы с языком управления заданиями. Они содержат все наиболее повторяющиеся операторы ЯУЗ, связанные с выполнением определенного действия. Например, каталогизированная процедура печати исходного текста

программы содержит управляющие операторы для вызова программы печати.

В рассматриваемом примере оператор EXEC осуществляет вызов каталогизированной процедуры трансляции с языка ФОРТРАН и выполнение программы пользователя

```
//  EXEC    FORTGCG
```

3. Оператор DD определяет данные, необходимые для выполнения программы, а также устройства ввода-вывода. Обычно в задании присутствует несколько операторов DD. Операторы DD находятся и в каталогизированной процедуре FORTGCG. Чтобы добавить операторы ЯУЗ к процедуре, необходимо в имени оператора DD указать имя соответствующего задания каталогизированной процедуры. Итак, для данного примера задание может иметь вид:

```
//PRIMER    JOB    'ИВАНОВ'
//  EXEC    FORTGCG
//FORT.SYSIN DD  *
  ИСХОДНЫЙ ТЕКСТ
  ПРОГРАММЫ НА ФОРТРАНЕ
/*
//
```

Два новых оператора ЯУЗ, приведенные в примере задания:

/* — конец файла (обозначает конец исходного текста);

// — конец задания.

Диалоговая система обеспечивает ввод и корректировку задания с дисплея и запуск задания на выполнение. Довольно распространенные ошибки — это ошибки в языке управления заданиями. Некоторые диалоговые системы обеспечивают входной контроль. Обычно контроль осуществляется после ввода очередной строки и сопровождается соответствующим диагностическим сообщением. Средства диалоговой системы обеспечивают контроль за выполнением задания в среде ОС/ЕС. Окончание выполнения задания наблюдается визуально через средства отображения активных заданий ОС/ЕС или сопровождается сигнальным сообщением на экран дисплея. Результаты можно просмотреть на экране или распечатать. С помощью символьного

сообщением на экран дисплея. Результаты можно просмотреть на экране или распечатать. С помощью символьного редактора текстов исправляются обнаруженные ошибки и повторно запускается задание на выполнение. Таким образом, диалоговая система в данном случае является надстройкой над операционной системой пакетной обработки ОС/ЕС.

Начиная с середины 70-х годов в развитии операционных систем наметились тенденции к усовершенствованию средств диалогового взаимодействия с пользователем. Появились операционные системы, которые обеспечивали на уровне команд или меню непосредственное управление своей работой. В ОС/ЕС командный уровень взаимодействия предназначен в основном для оператора ЭВМ и обеспечивает управление пакетной обработкой. Вмешательство пользователя в этом случае сильно ограничено, а часто и невозможно, так как одним из принципов пакетной обработки является ограничение доступа пользователя. В диалоговых операционных системах пользователь имеет непосредственный доступ к управлению операционной системой, при этом команды взаимодействия обеспечивают тот сервис, который предоставляла диалоговая система в ОС/ЕС. Перенесение взаимодействия с пользователем на уровень управляющей программы (монитора) операционной системы позволяет более гибко управлять ее ресурсами, обеспечивает "непосредственный контакт".

Диалоговые операционные системы не исключают пакетную обработку. Всегда имеется возможность включить команды монитора в пакетное задание. Совмещение диалоговой и пакетной обработки характерно для системы малых ЭВМ — СМ ЭВМ. Для семейства этих машин разработано более десяти операционных систем, ориентированных на различное применение. Наиболее распространенными для миниЭВМ (СМ-3, СМ-4, СМ-1420) являются: операционная система с распределенными функциями РАФОС и операционная система реального времени ОС РВ. Для ОС РАФОС характерно преобладание диалоговых форм общения с монитором на уровне команд; ОС РВ является развитой системой мультипрограммной обработки, ориентированной на высокопроизводительные модели СМ ЭВМ с большим числом пользователей. Эти системы обеспечивают каждому пользователю выделение

взаимодействия с монитором ОС, что имеет важное психологическое значение. В то же время пользователи могут обмениваться информацией друг с другом.

Широкое распространение персональных компьютеров привело к модернизации существующих и разработке новых диалоговых операционных систем для микроЭВМ. Работа на персональном компьютере происходит в режиме постоянного диалогового взаимодействия пользователя с монитором операционной системы. В настоящее время имеется ряд выпускаемых промышленностью микроЭВМ, наиболее распространенными из которых являются:

— диалоговый вычислительный комплекс ДВК (ДВК-2, ДВК-3,ДВК-4) с операционной системой ОС ДВК;

— микроЭВМ "Искра-226" с БЕЙСИК-интерпретирующей ОС;

— микроЭВМ СМ-1800, СМ-1810 с операционной системой микроДОС;

— микроЭВМ ЕС-1840 с операционной системой М86.

Общей особенностью всех перечисленных ОС является их ориентация на диалоговое взаимодействие с пользователем, при этом применяются как команды, так и меню, помощь (команда HELP), дополнительные запросы ОС в случае неполной информации от пользователя. Как правило, все эти системы являются "дружественными" пользователю и характеризуются высоким уровнем сервиса.

Однако многообразие операционных систем вызывает значительные трудности при переходе с одной ЭВМ на другую, так как при этом приходится осваивать новые команды монитора и сервисное программное обеспечение.

Необходимость стандартизации предопределило создание мобильных ОС, т.е. переносимых с одной ЭВМ на другую без изменения функциональных возможностей системы и интерфейсов с пользователем. Наиболее ярким представителем мобильных ОС является широко распространенная в мире ОС UNIX (в СССР используется совместимая с ней инструментальная мобильная операционная система ИНМОС).

Операционная система UNIX реализована практически полностью на языке высокого уровня С, который не зависит от конкретной ЭВМ, тем самым имеется возможность "переносить" ОС UNIX на различные машины (в настоящее время она реализована для всех типов

распространенных микро-, мини- и больших ЭВМ). Важной особенностью ОС UNIX является наличие простого интерфейса с пользователем и широкого набора сервисных средств. В нашей стране ОС UNIX и система ИНМОС реализованы для ЕС ЭВМ, СМ ЭВМ и ряда микроЭВМ (ДВК, "Искра-226", ЕС-1840, СМ-1810). Таким образом созданы предпосылки для принятия ОС UNIX и ОС ИНМОС в качестве стандартной ОС для большинства ЭВМ. Это удобно для пользователя, так как в этом случае он работает в одной операционной системе, и что очень важно, программы, написанные на одной ЭВМ, могут практически без изменений выполняться на другой машине. Необходимо, однако, отметить, что существуют определенные трудности при переносе мобильных ОС на другие ЭВМ, в частности перенесенная ОС далеко не всегда работает эффективно и быстро. Например, большое число обращений к внешней памяти на магнитных дисках сдерживает распространение ОС UNIX для микроЭВМ, оснащенных накопителями на гибких магнитных дисках. В связи с низким быстродействием этих устройств ОС UNIX работает на таких ЭВМ неэффективно. Поэтому применение ОС, ориентированных на конкретные микроЭВМ и учитывающих специфику их внешних устройств, еще достаточно распространено.

Дальнейшее развитие технической базы ЭВМ приведет к созданию новых операционных систем, еще более приближенных к пользователю. Интерес представляет проект создания ЭВМ пятого поколения, в которых будут применяться средства общения с пользователем на естественном языке и ЭВМ сможет решать задачи на логическом уровне в режиме, максимально приближенном к человеческому мышлению. Основной тенденцией в развитии системного программного обеспечения является его дальнейшее "сближение" с пользователем и, как результат, упрощение работы с ЭВМ, ускорение решения поставленных задач.

4.2. Программные средства диалога

Машинный диалог реализуется с помощью диалоговых программ и представляет собой пошаговый циклический процесс (см. 2.3). Допустим, что в некоторый момент времени система вышла в состояние ожидания запроса поль-

зователя. Пользователь подготавливает текст запроса, набирает его на клавиатуре и вводит в систему. При этом система выходит из состояния ожидания и приступает к обработке введенного запроса. В процессе обработки запроса система выдает пользователю информацию, предусмотренную алгоритмом обрабатывающей программы. По окончании обработки система вновь выходит в состояние ожидания следующего запроса пользователя. Действие системы в каждом шаге диалога определяется соответствующей диалоговой (интерактивной) программой, написанной на некотором языке программирования. Такой язык программирования должен иметь, во-первых, средства для описания сообщений и вывода их на экран дисплея и, во-вторых, средства для ввода и анализа запросов пользователя.

Современные наиболее распространенные языки программирования не содержат средств, в достаточной степени отвечающих указанным требованиям, поэтому на практике обычно используются различные расширения языков. Рассматриваемое ниже средство программирования интерактивных фрагментов представляет собой достаточно эффективное расширение языков Ассемблер, ПЛ/1, ПАСКАЛЬ и других для разработки развитых диалоговых программ. Описываемое средство применяется для разработки функциональных программ в системе "Фокус" и называется языком ДИФОЛ.

Язык ДИФОЛ появился в результате анализа элементов информационного обмена в диалоговых системах и накопленного опыта разработок программного обеспечения системы "Фокус".

Любую символьную информацию, выводимую на экран дисплея, можно представить в виде отдельных элементов или сообщений. Каждое такое сообщение определяется:

а) адресом оперативной памяти, содержащей текст сообщения;

б) длиной этого текста;

в) координатами на экране дисплея в виде номера строки и позиции в строке, где должен быть расположен данный текст.

С сообщением связаны определенные визуальные свойства текста, его статус и команды управления терминалом. Под термином «визуальные свойства» понимаются такие

параметры, как уровень яркости (интенсивность), цвет, режим мерцания или бегущая строка.

Кроме визуальных свойств сообщение обладает некоторым статусом по отношению к различным операциям, таким как:

— определение фотоселектором;
— удаление или модификация текста на экране;
— указание цифровых полей;
— табуляция.

Задавая то или иное значение параметра, определяющего статус, программист разрешает или запрещает применение к тексту сообщения соответствующей операции.

Сообщение несет в себе определенную информацию об управляющем воздействии на терминал, например: предварительное гашение экрана, сброс признаков модификации полей, разблокировка клавиатуры или подача звукового сигнала.

Таким образом, сообщение в общем виде можно представить следующей пятеркой:

$$M_{txt} = (T, K_t, V, S, C),$$

где $T = (A, L)$ — параметры, характеризующие расположение текста в оперативной памяти (A — адрес области памяти, L — длина текста);

$K_t = (l, p)$ — координата расположения текста на экране (l — строка на экране, p — позиция на строке);

V — параметры, характеризующие визуальные свойства текста;

S — статус сообщения по отношению к различным операциям;

C — управляющие параметры, характеризующие воздействие на терминал.

Возможны следующие частные случаи сообщения:

1) сообщение является текстом, состоящим из одних пробелов, необходимых для частичного стирания информации на экране дисплея:

$M_{clr} = (L, K_t, S, C)$;

2) сообщение несет в себе информацию об установке курсора на экране дисплея в определенную позицию, текст при этом отсутствует:

$M_{crs} = (K_c, C)$,

где $K_c = (l, p)$ — координата установки курсора на экране;

3) сообщение выполняет только управляющее воздействие на дисплей, например стирание экрана:

$M_{ers} = (C)$.

При использовании принципа умолчания многие параметры (в частности, атрибуты V, S, C) можно опустить для большинства сообщений, например:

$M_{out} = (T, K_t)$; $M_{clr} = (L, K_t)$; $M_{ers} = (K_c)$.

Кроме того, при использовании системы программного слежения за текущими координатами текста или курсора на экране дисплея опускается и указание параметров K_t и K_c:

$M_{out} = (T)$; $M_{clr} = (L)$.

Сообщения, передаваемые от дисплея в центральный процессор, представляют собой запросы пользователя; их можно представить в следующем виде:

$M_r = (I, K_c, K_t, T)$,

где I — код индикатора сигнала внимания; $K_c = (l_c, p_c)$ — координаты курсора на экране дисплея; $K_t = (l_t, p_t)$ — координаты одного или нескольких модифицированных полей на экране дисплея; $T = (A, L)$ — адрес расположения считанного текста в оперативной памяти и его длина.

При использовании фотоселектора или специальных клавиш программного прерывания текст не передается (частный случай);

$M_{fls} = (I, K_c, K_t)$.

Реакция системы на запрос пользователя однозначно определяется содержанием запроса и предысторией работы пользователя, т.е. предыдущими запросами:

$R = F(M_r)$.

Реализация описанного подхода достаточно наглядно представлена в языке ДИФОЛ.

Все операторы языка можно условно разбить на три группы: позиционирования и вывода информации на экран дисплея, ввода одного или нескольких модифицированных полей, предварительного анализа введенной информации.

К операторам позиционирования и вывода информации на экран дисплея относятся следующие:

ERASE — стереть все данные на экране;
CLEAR — стереть часть информации на экране;
CURSOR — установить курсор в указанную позицию экрана;
DISPLAY — выдать данные на экран дисплея.

К операторам ввода информации относятся:

BREAK — выполнить операцию чтения ручного ввода одного или нескольких модифицированных полей;
READSCR — выполнить операцию чтения буфера экрана.

К операторам анализа введенной информации относятся операторы:

PICK — выделить информацию одного модифицированого поля;
FORK — выполнить предварительный анализ считанной информации;
FORKA — определить идентификатор функциональной клавиши;
FINDCURS — определить положение курсора.

Операторы позиционирования и вывода информации содержат ключевой оператор AT, в котором записываются координаты установки курсора или начала вывода сообщения на экран дисплея, например:

```
CURSOR     AT=(8,20)
```

устанавливает курсор в 20-ю позицию на 8-й строке;

```
DISPLAY     'НАЧАЛО СЕАНСА',AT=(3,10)
```

выводит сообщение НАЧАЛО СЕАНСА на 3-ю строку, начиная с 10-й позиции.

Чтобы определить визуальные свойства и статус текста, а также управляющие параметры, кодируется операнд ATR, который называется атрибутом. Значения атрибута могут быть следующими:

UPR — установить поле, не защищенное от вмешательства оператора;
NUM — установить поле цифровое;
BR0 — установить поле неотображаемое;
BR1 — установить поле, отображаемое нормальной яркостью;
BR2 — установить поле, отображаемое повышенной яркостью;
FS1 — установить поле выбора фотоселектором (нормальной яркости);
FS2 — установить поле выбора фотоселектором (повышенной яркости);
FT1 — установить поле прерывания фотоселектором (нормальной яркости);
FT2 — установить поле прерывания фотоселектором (повышенной яркости);
MOD — установить поле модифицированное;
SND — подать звуковой сигнал;
DEB — разблокировать клавиатуру;
DEM — сбросить признак модификации.

Если поле имеет несколько атрибутов, то в операнд ATR значения записываются через запятую и заключаются в скобки. Например:

```
DISPLAY  'ВВЕДИТЕ ЗАПРОС',ATR=(BR2,SND,DEB)
```

выводит сообщение ВВЕДИТЕ ЗАПРОС повышенной яркости с подачей звукового сигнала и разблокировкой клавиатуры.

Переход в состояние ожидания и последующий ввод запроса осуществляется с помощью оператора BREAK. Оператор READSCR выполняет считывание информации с экрана дисплея без перехода в состояние ожидания. Для считывания информации с определенного места экрана можно выполнить предварительное позиционирование адреса буфера дисплея с помощью операнда AT, например

```
READSCR      AT=(10,1)
```

осуществляет считывание с 10-й строки экрана. Если позиционирование не используется, то считывание информации выполняется с начала экрана.

Анализ введенной информации может быть выполнен средствами используемого языка программирования, однако на практике удобно воспользоваться операторами PICK, FORK, FORKA и FINDCURS, имеющимися в языке ДИФОЛ. Операторы языка ДИФОЛ подробно описываются в документации по системе "Фокус". Дополнительно отметим, что язык ДИФОЛ представляет собой достаточно развитое средство программирования интерактивных фрагментов для программ на ассемблере, на языках высокого уровня и для непосредственного выполнения в системе. Последняя возможность позволяет автоматизировать работу пользователя системы путем программирования отдельных сценариев диалога. Такой режим работы в системе "Фокус" называется режимом автопилота.

Для программирования диалога на персональных микрокомпьютерах широко используется язык БЕЙСИК (начальные буквы Beginners Allpurpose Symbolic Instruction

Code), разработанный в Дортмундском колледже (США) в 1964 г. Для разных компьютеров существуют различные версии и расширения языка БЕЙСИК, связанные со спецификой аппаратной реализации компьютеров.

При программировании на языке БЕЙСИК необходимо указать компьютеру последовательность выполнения команд. Для этого в каждой строке перед последовательностью команд указывается номер, который обычно начинается с номера 10 и увеличивается на 10 для каждой новой строки. Это позволяет вставлять в текст программы дополнительные строки при необходимости модификации программы.

Рассмотрим следующую программу:

```
10 LET     A%= "СКОЛЬКО ВАМ ЛЕТ?"
20 LET     B%= "ВАШ ВОЗРАСТ -   "
30 LET     C%= "ЛЕТ"
40 INPUT  (A%);X
50 PRINT   B%;X;C%
```

Ввод такой программы в компьютер осуществляется обычным набором на клавиатуре, который завершает каждую строку нажатием клавиши ВВОД. Чтобы выполнить такую программу, используется команда RUN, вводимая без указания номера.

Таким образом, если вводится команда с указанием номера строки, то компьютер не выполняет ее немедленно, а запоминает в программе (в порядке возрастания номеров строк). Чтобы выполнить команду непосредственно после ввода, ее следует ввести без номера строки.

Для организации циклов в языке БЕЙСИК используется конструкция FOR ... TO ... STEP и NEXT.

Следующая программа выводит на экран числа от 1 до 10 и их квадраты:

```
10 FOR X=1 TO 10 STEP 1
20 PRINT X,X*X
30 NEXT X
```

Приведенные примеры программ характеризуют две сферы применения языка: организацию диалога, математические вычисления. Рассмотрим программу, в которой представлены обе сферы применения языка БЕЙСИК:

```
10  REM   ПРЕОБРАЗОВАНИЕ ТЕМПЕРАТУР
20  PRINT "ГРАД.С", "ГРАД.F"
30  PRINT
40  INPUT "ВВЕДИТЕ ГРАД.С", C
50  PRINT C, 32+C*9/5
60  GO TO 40
```

Данная программа преобразует значения температур из шкалы по Цельсию в шкалу по Фаренгейту. Если выполнить эту программу, то на экране появится заголовок, выведенный командой в строке 20. Строка 10 игнорируется компьютером и представляет собой ремарку или комментарий. При выполнении команды в строке 40 компьютер выйдет в состояние ожидания ввода значения переменной С. После того как это значение будет введено, компьютер выведет результат (строка 50) и вновь по команде GO TO 40 перейдет на ожидание ввода следующего значения. Очевидно, что строки 40—60 представляют собой бесконечный цикл. Для выхода из этого цикла необходимо ввести команду STOP.

Оператор PRINT может иметь более сложную форму, например:

```
PRINT   AT 10,26; "*"
```

выводит звездочку в 26-ю позицию на 10-й строке. Таким образом, наличие параметра "AT строка, позиция" позволяет осуществлять позиционирование выводимых данных по указанным координатам.

Кроме параметра AT можно указать параметры, определяющие цвет (INK, PAPER), яркость (BRIGHT) и другие атрибуты сообщения. Однако эти параметры относятся к расширению языка БЕЙСИК и их синтаксис отличается для различных компьютеров.

Еще один язык программирования АПЛ — диалоговый язык для описания процессов обработки информации, рассчитанный на широкий круг пользователей различной квалификации и специализации. Этот язык был разработан

К.Э. Айверсоном в 1962 г. и является одним из наиболее выразительных, последовательных и мощных языков программирования. АПЛ имеет педагогическую ценность, так как обучаемый приступает к работе, не знакомясь с широким набором операторов языка. АПЛ обеспечивает простую возможность вычисления различных математических функций, действия с матрицами, организации систем кодирования и защиты информации, обработки многомерных массивов, ведения файлов.

4.3. Графические возможности дисплея

Рассмотрим графические средства дисплеев и особенности программирования графики, а также графические возможности экрана дисплея с матрицей размера 192×640 точек.

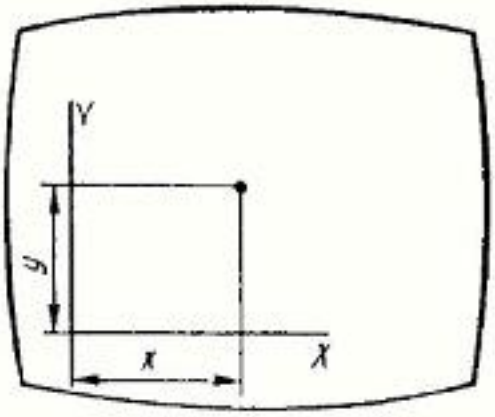

Рис. 4.2

Положение любой точки на экране дисплея определяется координатами. Координата x указывает, насколько далеко расположена данная точка от левого края поля экрана, а координата y — расстояние от нижнего края экрана до данной точки, т.е. x — координата, определяемая по горизонтали слева направо, а y — координата, определяемая по вертикали снизу вверх (рис. 4.2). Будем записывать эти координаты через запятую и в скобках (x, y). Таким образом, пары координат (0,0), (639,0), (0,191) и (639,191) определяют положение угловых точек на экране.

Оператор PLOT—простейший оператор экранной графики. Он выводит на экран дисплея точку в соответствии с указанными координатами. Формат оператора:

PLOT *x—координата, y—координата.*

С помощью оператора CIRCLE можно изображать на экране полные окружности. Формат оператора:

CIRCLE *x*-координата, *y*-координата, радиус.

Операнды в операторе CIRCLE определяют координаты центра (x, y) и радиус окружности.

Операторы PLOT, DRAW и CIRCLE представляют собой пример так называемой экранной графики, которая использует для изображений всю площадь экрана дисплея.

Ниже приведена программа на языке БЕЙСИК, которая выводит на экран один период синусоиды (при значении аргумента от 0 до 2π):

```
10  FOR  N=0 TO 639
20  PLOT N,96+90*SIN(N/320*PI)
30  NEXT N
```

Следующая программа выводит на экран дисплея график параболы при изменении аргумента от 0 до 4:

```
10  FOR  N=0 TO 639
20  PLOT N,90*SQR(N/160)
30  NEXT N
```

Обратите внимание на различие координат графической точки (x, y) и координат строки и позиции (c, n) в операнде AT (см. параграф 4.2).

Использование единственного оператора PLOT для сложных графических изображений представляется достаточно трудоемким, поэтому наряду с ним применяются операторы для вычерчивания прямых линий и дуг (DRAW) и окружностей (CIRCLE).

Оператор DRAW для изображения прямых линий имеет следующий формат:

```
DRAW X,Y
```

Начало линии определяется координатами конечной точки в результате выполнения предыдущего оператора PLOT, DRAW или CIRCLE, а конец линии — операндами x и y, которые в операторе DRAW являются относительными координатами.

Рассмотрим последовательность операторов

```
PLOT    0,40 :DRAW  200,  110
PLOT  400,160:DRAW -100, -140
```

и соответствующий график на рис.4.3. Операнды в операторе DRAW могут принимать отрицательные значения в отличие от операндов в операторе PLOT.

С помощью оператора DRAW кроме прямых линий можно вычерчивать на экране дуги. Для этого указывается третий операнд, определяющий угол дуги. Формат оператора DRAW для изображения дуг имеет следующий вид:

DRAW *x*, *y*, *a*

Операнды *x* и *y* определяют координаты положения конечной точки дуги, как и для прямых линий; операнд *a* — угол дуги, рад. Если операнд *a* имеет положительное значение, то в процессе вычерчивания дуги осуществляется поворот налево, а для отрицательных *a* — направо. При $a = 2\pi$ будет начерчена полная окружность, при $a = \pi$ — полуокружность, при $a = 0.5*\pi$ — четверть окружности и т.д. Например, для $a = \pi$ независимо от значений x и y будет начерчена полуокружность (рис. 4.4) операторами

```
PLOT  200,50; DRAW 120,90,PI
```

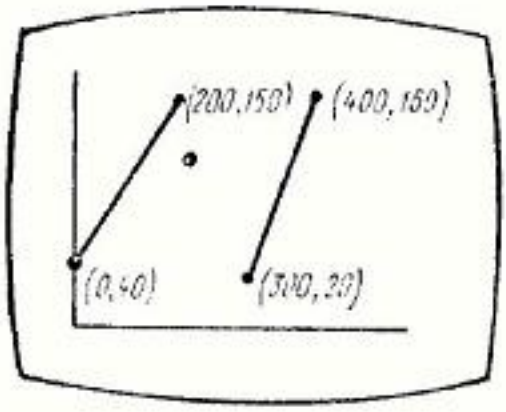

Рис. 4.3

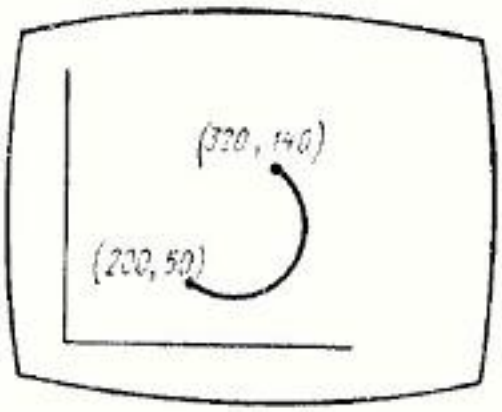

Рис. 4.4

Вычерчивание начинается в юго-восточном направлении и заканчивается в северо-западном, т.е. происходит поворот на 180° или π рад (значение a).

Графические возможности в языке БЕЙСИК являются одним из простейших примеров машинной графики.

Другим примером является "масштабная оконная графика", которая в отличие от "точечной графики" использует произвольное двухмерное пространство, не ограниченное размером экрана. На экран дисплея, называемый в данном случае окном, выводится только часть

Вычерчивание фигур осуществляется графическим курсором (указателем), который может быть неотображенным на экране. Последовательное выполнение графических операций осуществляется относительно положения курсора (относительная операция) или относительно осей координат (абсолютная операция).

Для лучшего понимания масштабной графики введем понятие *мировых координат*, которые позволяют задавать объекты в двухмерном или трехмерном мире пользователя, при этом пользователь определяет единицы измерения, например: график движения в километрах при времени в часах или зависимость выхода продукции в тысячах штук от количества дней.

Масштабный фактор, т.е. полное расстояние в системе мировых координат по вертикали окна, имеет значение 100 по умолчанию и может быть изменен с помощью оператора SCALE. Масштабы координат *x* и *y* одинаковы. Однако длина линии, которую можно начертить вдоль координаты *x*, зависит от масштабного фактора окна. Увеличивая или уменьшая масштабный фактор окна, можно увеличить или уменьшить размер начерченной в окне фигуры.

Важно понять, что в окне просматривается либо все графическое пространство, либо его фрагменты в зависимости от выбираемого масштабного фактора.

Ниже перечислены основные операторы "масштабной оконной графики" и их назначение:

CIRCLE — вычерчивание эллипса или окружности;
LINE — вычерчивание линии;
ARC — вычерчивание дуги или окружности;
POINT — вычерчивание точки;
SCALE — установка масштаба и перемещение окна;
FILL — штриховка;
CURSOR — позиционирование текста.

Оператор SCALE позволяет изменять масштабный фактор и перемещать окно в области экрана дисплея. Масштаб определяет длину вертикальной линии, которая полностью помещается на экране дисплея. Масштаб 100 устанавливается по умолчанию. Формат оператора:

SCALE масштаб, координата *x*, координата *y*

Координаты (*x*, *y*) определяют положение левого нижнего угла окна в графическом пространстве.

Примеры оператора SCALE:

```
SCALE 0.5, 0.1, 0.1
```

— устанавливает масштабный фактор 0.5 и позиционирует окно по координатам (0.1, 0.1) графического пространства;

```
SCALE 10, 0, 0
```

— устанавливает масштабный фактор 10 и позиционирует окно по координатам (0, 0) графического пространства;

```
SCALE 100, 50, 50
```

— устанавливает масштабный фактор 100 и позиционирует окно по координатам (50, 50) графического пространства.

Для того чтобы высветить на экране точку, используется оператор POINT. Координаты точки задаются операндами *x* и *y*. При использовании масштабного фактора 100 (по вертикали) точка в центре окна устанавливается с помощью оператора

```
POINT  60,50
```

Для вычерчивания линий используется оператор LINE_R. Например:

```
LINE  60,50  TO  80,90
```

—выводит на экране линию с координатами начала (60,50) и координатами конца (80,90).

Оператор LINE вычерчивает любую ломаную линию или, например, квадрат (рис. 4.5):

```
LINE  60,50  TO  70,50  TO  70,60  TO  60,60  TO 60,50
```

Значения операндов определяют абсолютные координаты в графическом пространстве. Иногда удобнее чертить относительно текущей позиции курсора. В этом случае используется относительный оператор LINE_R. Например,

квадрат, начерченный в приведенном примере, определяется в относительных координатах с помощью оператора

```
POINT 60,50

LINE_R 0,0 TO 10,0 TO 0,10 TO -10,0 TO 0,-10
```

Чтобы начертить окружность, необходимо указать координаты центра, например, (50,50) и радиус 40. Тогда с помощью оператора

```
CIRCLE  50,50,40
```

вычерчивается окружность с координатами центра (50,50) и радиусом (или "высотой") в 40 ед. (рис. 4.6).

Если добавить еще два параметра, например:

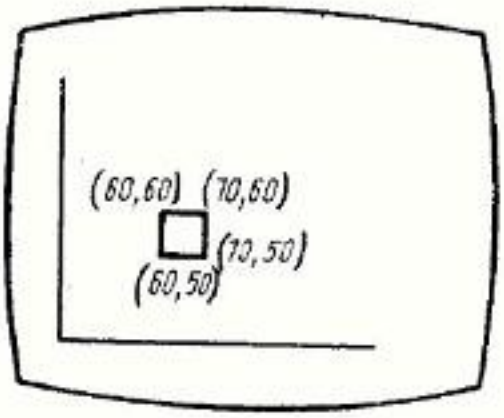

Рис. 4.5

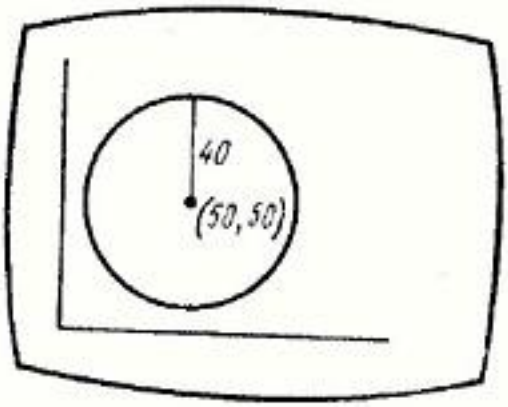

Рис. 4.6

```
CIRCLE  50,50,40,0.5,0
```

то получится эллипс (рис. 4.7).

"Высота" эллипса составляет 40 ед., но горизонтальный "радиус" составляет только половину (0,5) от высоты и равен 20 ед. Число 0,5 в этом примере представляет собой эксцентриситет эллипса. Если эксцентриситет равен 1, то получается окружность, а если не равен 1, то вычерчивается эллипс. Если требуется повернуть эллипс, то следует изменить 5-й параметр, например:

```
CIRCLE  50,50,40,0.5,1
```

В результате вычерчивается эллипс, повернутый против часовой стрелки на угол в 1 рад (приблизительно 57°) (рис. 4.8).

Таким образом, параметры для изображения эллипса и окружности должны указываться в операторе CIRCLE в определенной последовательности: *x*, *y* — координаты центра, высота, [эксцентриситет], [угол поворота]. Последние два параметра можно опустить, поэтому они указаны в квадратных скобках. Значения этих параметров по умолчанию 1, 0 соответственно.

Если требуется начертить дугу окружности, то необходимо использовать оператор ARC, в котором следует указать координаты начала и конца дуги, а также угол дуги в радианах. Например:

```
ARC  10,30  TO  60,90,1
```

изображает дугу, приведенную на рис.4.9.

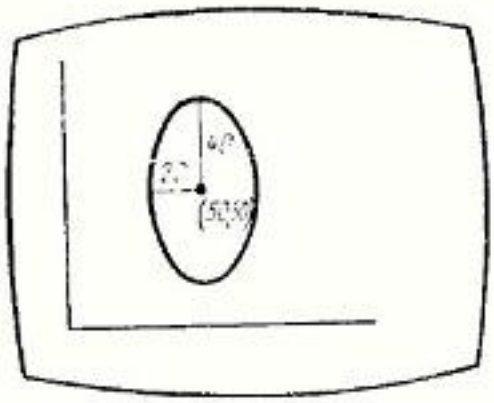

Рис. 4.7

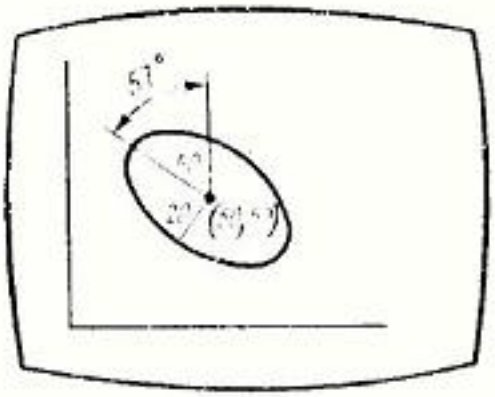

Рис. 4.8

Любую замкнутую фигуру можно заштриховать с помощью оператора FILL. Этот оператор имеет единственный параметр, значение которого может быть либо 1 (включить штрихование), либо 0 (выключить штрихование). Для изображения заштрихованного круга используется последовательность операторов FILL и CIRCLE, например

```
FILL 1
CIRCLE 50,50,30
```

Штриховка выполняется горизонтальными линиями. Для отключения штриховки используется оператор FILL 0.

Масштабная оконная графика достаточно проста, но позволяет создавать на экране сложные чертежи с высокой разрешающей способностью.

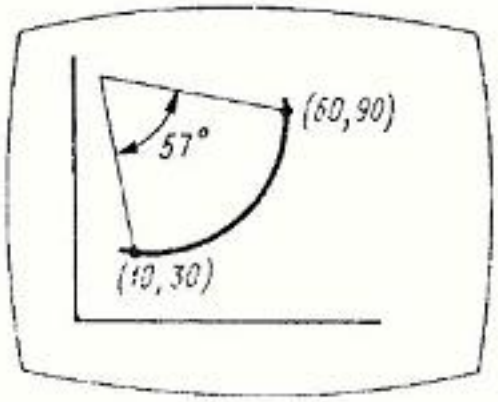

Рис. 4.9

Покажем еще один способ представления графической информации, достаточно распространенный в микрокомпьютерах. Представим себе маленькую невидимую черепаху, которая имеет пишущее перо и может перемещаться в плоскости экрана.

Существуют операторы управления пером и черепахой:

PENUP — поднять перо;
PENDOWN — опустить перо;
MOVE — переместить черепаху;
TURN — повернуть черепаху (относительно);
TURNTO — повернуть черепаху (абсолютно).

Будем называть описываемый способ представления графической информации Т-графикой (от английского turtle graphic — черепашья графика). Особенность ее заключается в том, что, как известно, черепаха не умеет прыгать и поэтому перемещается по непрерывной траектории. Первоначально черепаха находится в левом нижнем углу экрана и имеет направление 0° (вправо). При этом перо поднято. С помощью указанных операторов можно поворачивать и перемещать черепаху по экрану, причем в зависимости от того, опущено или поднято перо, черепаха оставляет за собой след или нет. Оператор MOVE имеет один параметр, указывающий расстояние, на которое перемещается черепаха. Параметр оператора TURN определяет поворот черепахи на необходимый угол относительно текущего направления. Если значение параметра положительное, то поворот осуществляется против часовой стрелки, в противном случае — по часовой стрелке. Оператор TURNTO с помощью параметра указывает направление движения черепахи.

Описанный набор операторов Т-графики является минимальным и обычно используется в процедурах для получения расширенного набора операторов. Например, процедура

```
10  DEFINE PROCEDURE FORWARD (DISTANCE)
20    MOVE DISTANCE
30  END DEFINE
```

определяет новый оператор FORWARD для движения черепахи вперед на расстояние, значение которого определяется параметром. Следующие три процедуры

```
40  DEFINE PROCEDURE BACKWARD (DISTANCE)
50    MOVE -DISTANCE
60  END DEFINE
70  DEFINE PROCEDURE LEFT (ANGLE)
80    TURN ANGLE
90  END DEFINE
100 DEFINE PROCEDURE RIGHT (ANGLE)
110   TURN -ANGLE
120 END DEFINE
```

определяют операторы BACKWARD, LEFT и RIGHT для движения черепахи назад, поворотов налево и направо соответственно.

На основе Т-графики разработан графический язык программирования ЛОГО, который является одним из наиболее простых и доступных для изучения языков программирования. Основу языка составляет интерактивная графика и достаточно простое средство создания новых команд языка. Таким образом, в процессе программирования на ЛОГО язык обогащается новыми словами и все более отвечает требованиям пользователя. Язык характеризуется рекурсивностью, наличием процедур, возможностью струк-

турного программирования, обработки списков, вычисления математических функций.

Многочисленные расширения языка обеспечивают имитацию физических явлений, применяются в робототехнике, игровых и обучающих программах.

В качестве развитого графического языка высокого уровня рассмотрим язык Калсруда, который имеет средства обработки таких графических типов данных, как точки, отрезки прямых, рисунки, подэлементы и области. Так как синтаксис языка принимает различные формы в зависимости от вкуса разработчика, рассмотрим только семантику операторов.

Команды для описания изображения

SCALE Параметры — имя и масштаб. Определяет размер видимой области и генерирует сетку с указанным шагом.

POINT Параметры — имя и список точек (пар координат). Выводит на экран точки в соответствии с координатами. Приписывает заданное имя всей совокупности точек.

LINE Параметры — имя и список точек. Рисует ломаную с вершинами в заданных точках. Точки соединяются в том порядке, в котором они встречаются во входном списке. Приписывает ломаной заданное имя.

ARC Параметры — список точек. Через две точки проводит дугу, центр которой находится в третьей точке. Приписывает дуге заданное имя.

FUNCT Параметры — имя и выражения, представляющие собой функциональные зависимости.

CURVE Параметры — имя и числовые данные, определяющие крайние и промежуточные точки интервала, в котором вычисляются функции, идентифицируемые именем. Рисует кривую.

Команды для манипуляции изображением

PRINT Выводит на экран сформированное изображение. Можно задать список параметров, определяющих дополнительные элементы, накладываемые на результирующее изображение, но отсутствующие в памяти (например, координатные оси, сетка и т.д.).

STORE Сохраняет в стеке изображений под заданным именем графическую информацию, с которой в данный момент ведется работа.

GET Выводит на экран изображение, хранящееся в стеке под заданным именем.

NAME Параметр — список имен. Команда помечает подэлементы текущего изображения именами из заданного списка. Подэлементы связываются друг с другом, их совокупности присваивается общее имя, также заданное в списке.

COPY Параметр — список имен объектов, хранящихся в стеке изображений. Эти объекты добавляются к изображению, находящемуся в данный момент на экране.

ROTATE Параметры — список имен подэлементов и числовые данные, которые задают преобразование вращением, применяемое к указанным подэлементам.

ERASE Параметр — список имен, запомненных в стеке изображений или подэлементов текущего изображения. Объекты с этими именами уничтожаются.

Команды для анализа областей

REGION Идентифицирует область, содержащую указанную точку, и присваивает этой области заданное имя. Может быть также задано условие, которому эта область должна удовлетворять.

CONNECT Проверяет, принадлежат ли две указанные точки одной и той же области и лежит ли в этой области кривая, соединяющая эти точки.

ADJAC Проверяет смежность двух указанных объектов. Результат выдается как логическое значение.

INTERX Находит точку пересечения двух прямых.

DIST Находит число шагов между двумя точками. Под шагом понимается перемещение от точки к ближайшей соседней точке.

Команды для топологического анализа

WITHIN Проверяет включение некоторого подэлемента в другой подэлемент.

SEPAR Проверяет наличие или отсутствие пересечения двух подэлементов.

SIMPLY Проверяет односвязность области, образуемой изображением или его подэлементом.

Язык Калеруда можно рассматривать как модель графических языков высокого уровня. Помимо обычных для языков высокого уровня типов данных этот язык обрабатывает практически полный набор графических

типов данных: точки, отрезки прямых, рисунки, подэлементы и области.

Вопросы для самопроверки

4.1. Приведите определение операционной системы и назовите известные операционные системы.

4.2. Что такое операционное окружение пользователя?

4.3. Что такое задание для операционной системы с точки зрения ОС/ЕС?

4.4. Перечислите наиболее часто употребляемые операторы языка управления заданиями ОС/ЕС.

4.5. Для чего используются каталогизированные процедуры?

4.6. Что такое графическая точка и сколько их на экране графического дисплея?

4.7. Объясните, как работает программа на языке БЕЙСИК, приведенная на с. 77 (преобразование температур).

4.8. Можно ли с помощью оператора DRAW начертить на экране окружность?

4.9. Что определяют операнды в операторах: а) PLOT, б) DRAW, в) CIRCLE?

4.10. Напишите программу на языке БЭЙСИК для вычерчивания на экране прямых линий. Все линии должны иметь случайные длину и направление. Каждая линия должна начинаться в точке завершения предыдущей. Для получения случайной величины от 0 до 1 используйте встроенную функцию RND.

4.11. Напишите программу для вычерчивания на экране прямых линий случайной длины. Все линии должны быть горизонтальными.

4.12. Аналогично п. 4.11, но для вертикальных линий.

4.13. Напишите программу для изображения "квадратной спирали".

Глава 5
РЕАЛИЗАЦИЯ ДИАЛОГА

5.1. Постановка задачи автоматизации конторской деятельности

В настоящей главе рассматриваются этапы жизненного цикла диалоговых программных средств на примере отдельных простых программных модулей. Применительно к диалоговой программе такими этапами являются: разработка сценария диалога, собственно программирование, тестирование программы, внедрение нового программного модуля, сопровождение диалоговой программы или новой версии диалоговой системы.

Каждый из этих этапов описан подробно (параграфы 5.2—5.6). Качество документирования каждого этапа жизненного цикла программ неразрывно связано с глубиной его проработки. Снижение степени проработки любого этапа ведет к повышению затрат на следующих этапах. Поэтому описание особенностей документирования диалоговых программ выделено в отдельный параграф.

Актуальность автоматизации конторской деятельности, т.е. деятельности работников административно-управленческого аппарата, определяется следующими двумя обстоятельствами:

1. В связи с развитием экономики и повышением уровня ее сложности доля населения, занятого в сфере управления, возрастает. Возможности для этого создаются за счет высвобождения части работающих в производящих отраслях в связи со значительным ростом в них производительности труда, в том числе и за счет автоматизации.

2. Темпы роста производительности труда в сфере управления за последнее столетие отставали ежегодно от

темпов ее роста в сфере производства в среднем в 4—5 раз. Кроме того, стоимость оборудования, приходящаяся на одного конторского работника, в настоящее время примерно в 20 раз меньше аналогичного показателя для промышленного рабочего.

В то же время уровень затрат на обработку информации в различных сферах следующий:

— в сфере управления — около 80%;

— в административной деятельности — до 90%;

— в области управления технологическими процессами — 60%.

Поэтому правомерно следует вывод: снижение, даже на небольшую долю, затрат на обработку информации в административно-управленческой деятельности ведет к значительной экономии трудовых ресурсов.

Многим выпускникам вуза после его окончания предстоит заниматься административно-управленческой деятельностью, где имеются большие резервы повышения производительности труда работающих. Поэтому проблемы реализации диалога целесообразно рассматривать на примере автоматизации этого вида деятельности.

Содержание деятельности работника административно-управленческого аппарата определяет состав автоматизированной системы конторской деятельности в зависимости от специфики предприятий и организаций, традиций и потребностей народного хозяйства. Распределение времени на каждый вид деятельности различных категорий работников этой сферы может быть разным, хотя отличия не превышают нескольких процентов. Ориентировочное распределение рабочего времени на основании проведенных исследований можно принять следующим.

Служащие: управление — 24%, техническая работа — 40, исполнение — 36%.

Специалисты: совещания — 17%, телефонные переговоры — 12, командировки — 6, архивирование и запрос информации — 11, работа за столом — 30, другие виды работ — 24%.

Технические исполнители: расчеты — 25%, написание документов — 19, телефонные переговоры — 18, работа в архиве — 12, коммуникации — 7, размножение документов — 6, другие виды работ — 13%.

Секретари: переговоры с руководством — 6,5%, телефонные переговоры — 7,5, исполнительская работа — 13, размножение документов — 20, запросы — 2,5, архивирование — 2,5, ожидание работы — 18, подготовка почты — 3, диктовка — 2,5, печатание на машинке — 20, другие виды работ — 4,5%.

Анализ деятельности работников административно-управленческого аппарата показывает, что на работу с документацией расходуется от 20 до 50% рабочего времени в зависимости от должности, специфики работы и квалификации исполнителя. В то же время использование дисплея в диалоговом режиме снижает до 50% время на подготовку документов по сравнению с традиционными формами работы.

Наибольший эффект, как и при автоматизации других видов человеческой деятельности (автоматизированное проектирование, гибкие производственные системы и др.), дает не фрагментарное, а системное использование ЭВМ. Такую возможность предоставляют *автоматизированные рабочие места работников аппарата управления* (АРМ/У).

Состав АРМ/У:

операционная система с системными сервисными программными средствами (текстовой и, возможно, графический редакторы, другие средства);

диалоговая система;

СУБД;

генератор отчетов;

библиотека подпрограмм;

сетевые средства (при работе в составе локальной сети ЭВМ);

минимальные технические характеристики АРМ/У на базе универсальной персональной микроЭВМ:

1) быстродействие — десятки тысяч операций в секунду;

2) объем оперативной памяти 56 Кбайт;

3) наличие алфавитно-цифрового дисплея, содержащего 80 символов в каждой из 24 строк;

4) наличие спаренного накопителя на гибких магнитных дисках емкостью примерно 200 Кбайт каждый;

5) наличие печатающего устройства последовательного действия;

с развитием информационной базы и расширением сфер применения могут потребоваться дополнительные технические средства:

1) графический дисплей;

2) мозаичное печатающее устройство;

3) жесткий магнитный диск (или так называемый "винчестеровский" накопитель с жестким магнитным диском);

средства сетевого обеспечения.

Применение АРМ/У определяется следующими сферами:

— обучение работе с программно-аппаратными комплексами рабочего места;

— создание и ведение локальной базы данных;

— генерация выходных форм по оперативным и регламентированным запросам;

— ведение справочной информации, помогающей организовать рабочее время административного работника (справочные списки, календари и др.);

— создание, редактирование и копирование различных документов на "твердую" копию (бумагу, гибкий магнитный диск);

— проведение вспомогательных математических расчетов;

— обмен информацией с центральной базой данных учреждения (через каналы связи — в случае подключения к локальной сети организации, или через гибкий магнитный диск — в противном случае);

— создание новых диалоговых программных средств (при наличии соответствующей квалификации исполнителя);

— проведение заочных конференций, автоматизированная почта (при работе в сети ЭВМ).

Таким образом, при относительной простоте и ясности автоматизируемых функций административно-управленческого аппарата задача автоматизации конторской деятельности требует применения мощных программных и технических средств. Эта задача предъявляет к пользователю требования по владению навыками работы с ЭВМ, знанию основ построения диалоговых программных средств.

Рассмотрим наиболее общие принципы процесса проектирования и реализации системы.

Зададимся вопросом: почему сравнительно мало "хороших" диалоговых систем? Ведь если рассмотреть весь цикл проектирования и реализации, то он представляет собой довольно четкую последовательность требований и действий, правильное выполнение которых должно было бы гарантировать создание "хороших" систем. Однако на деле эти требования часто нарушаются или вообще не выполняются по следующим причинам. Во-первых, это связано со спецификой разработки программного обеспечения. Довольно четкие требования, предъявляемые к разрабатываемой программе, в конкретном применении имеют расплывчатый характер. Действительно, требования к создаваемой программе вырабатываются без малейшего сопротивления со стороны человека: взял чистый лист бумаги и написал, опираясь на эти требования, программу. Ввел ее в компьютер, отладил и получилось, что запрограммирована не та задача! Неверно сформулированные начальные требования и ряд накопившихся неточностей приводят к тому, что нужно переделывать все заново. Этот процесс может повторяться многократно. Причина кроется в том, что многие программисты стремятся скорее приблизить во времени стадию реализации, т.е. быстрее "пойти на компьютер". При этом выработке требований и проектированию уделяется очень мало времени. В то же время подготовка полных и правильных спецификаций — самая ответственная задача в разработке программного обеспечения.

Спецификация — это задание на разработку программы, которое является результатом предварительного этапа — этапа эскизных требований. Эскизные требования определяются на уровне постановки задачи. В процессе дальнейшей детализации и вырабатываются спецификации. На основе спецификаций разрабатываются программы. В то же время в процессе реализации могут выявиться отклонения от исходных требований и понадобятся изменения исходных спецификаций.

Модифицируемость — важное качество хороших спецификаций.

Можно выделить два основных этапа проектирования: выработка эскизных требований и разработка спецификаций.

При этом необходимо уделять особое внимание разработке сценария диалога, представляющего собой описание всей совокупности обменов между пользователем и диалоговой системой.

Будем понимать под обменом операцию, связанную с выводом данных с клавиатуры терминала в ответ на запрос диалоговой системы. Совокупность обменов по выполнению определенной функции системы назовем процедурой. Очевидно, что процедура может состоять и из одного обмена.

Рассмотрим наиболее существенные факторы, которые необходимо учитывать при проектировании сценария диалога:

1. **Конечные цели диалога.** Эти цели вытекают из постановки задачи диалога. Лучше всего изобразить цели в виде графа диалога (см. гл. 2). Граф диалога должен предусматривать асинхронные выходы, т.е. при желании пользователь должен иметь возможность вернуться на верхний уровень из любой точки обмена.

2. **Входной контроль.** В процессе диалога необходимо осуществлять контроль данных, т.е. анализировать информацию, поступающую извне на корректность. Диалоговая система должна обрабатывать ситуацию ошибки и выдавать подробные диагностические сообщения.

Отличительной чертой хорошей диалоговой программы является развитая *диагностика ошибок* ввода данных.

Подробные диагностические сообщения — залог высокопроизводительной работы пользователя с диалоговой системой. Действительно, если система выдает: У ВАС ОШИБКА, то это малоинформативное сообщение. Система должна указывать на конкретную ошибку (обычно для этого используется специальный символ, например восклицательный знак "!") и подробно объяснять ее причину.

3. **Расположение сообщений на экране.** Существенное значение для правильного и быстрого восприятия информации имеет расположение текста сообщений на экране. Основные способы представления сообщений описаны в главе 2.

Сценарий диалога, выраженный в спецификациях, является наиболее конкретным представлением проекта диалоговой системы. Проверка готовой системы облегчается

при наличии хороших спецификаций. Если требуется внести изменения в программу, то необходимо вносить изменения и в соответствующую спецификацию. Основное внимание необходимо уделять проектированию диалоговой системы.

5.3. Программирование

Программирование начинается с выбора языка программирования. При этом предпочтение отдается языку высокого уровня.

Вероятность допущения ошибки на языке высокого уровня гораздо меньше, чем на языке ассемблера. Кроме того, производительность программирования на языке высокого уровня гораздо выше. Для разработки диалоговых систем наиболее подходящими являются такие распространенные языки, как БЕЙСИК, ПАСКАЛЬ, АДА. Если выбор языка сделан (а на это влияет наличие или отсутствие соответствующего компилятора на машине), то следующим этапом является собственно программирование. Характерной чертой разработки программного обеспечения является его технологичность. Использование технологии программирования обеспечивает создание качественного программного обеспечения. Примером такой технологии является широко применяющееся структурное программирование.

При реализации диалоговой системы стуктурное программирование обеспечивает четкую логику программирования и способствует получению программ, удобных для сопровождения и внесения изменений.

Основными составляющими структурного программирования являются: проектирование сверху вниз, модульное программирование, структурное кодирование.

Проектирование сверху вниз предполагает определение задачи в общих чертах с последующей ее детализацией (уточнением). Уточнение задачи приводит к появлению нижних уровней иерархии (рис. 5.1).

Например, общая задача — написать диалоговую систему. Изобразим верхний уровень иерархии в виде прямоугольника, внутри которого запишем: "диалоговая система". Уточним, из чего она состоит. Так появляется второй уровень иерархии, который включает в себя уже три прямоугольника. В свою очередь, каждая программа

на втором уровне иерархии также уточняется. В результате получается древовидная иерархическая структура системы.

Проектирование сверху вниз позволяет точно представить структуру системы.

Очевидно, что сценарий диалога также лучше проектировать методом сверху вниз.

Модульное программирование — это разделение программы на части, называемые модулями. Модуль оформляется в виде подпрограммы или процедуры. При модульном программировании важно знать, какие данные поступают на вход модуля и какие данные должны быть на выходе.

Рис. 5.1

При этом модуль работает как "черный ящик", т.е. с точки зрения других модулей совершенно неважно, что делается внутри модуля, главное, чтобы он выдавал правильные выходные данные. Этим обеспечивается независимость модулей.

Модульное программирование является логическим продолжением проектирования сверху вниз. Действительно, каждый уровень иерархии на рис. 5.1 представляется одним или несколькими прямоугольниками, которые реализуются в виде соответствующих модулей. Разбиение на модули облегчает программирование и отладку.

Структурное кодирование — это метод написания модулей, позволяющий получать программы, удобные для чтения, тестирования и модификации. Основу структурного кодирования составляет принцип написания программ любой сложности с помощью ограниченного числа простых логических структур: последовательности операторов, условного выбора, повторения (цикла).

Последовательность операторов предполагает их последовательное выполнение. Условный выбор — это широко распространенная конструкция IF-THEN-ELSE (если-то-иначе). Повторение — это цикл типа WHILE, выполняющийся до достижения определенного условия.

Таким образом, из рассмотренного выше можно сделать следующие выводы:

1. Применение структурного программирования позволяет внести ясность в структуру программ и данных.

2. Применение при написании программы одних и тех же логических структур стандартизирует стиль программирования, улучшает удобочитаемость программы, делает ее понятной.

3. Модульное программирование — один из наиболее эффективных способов написания больших, в том числе диалоговых, программ и систем.

5.4. Тестирование

Только очень простая программа обычно выполняется сразу без ошибок. К сожалению, большинство программ содержат ошибки. При реализации диалога положение усложняется тем, что диалоговая система должна содержать средства защиты от неправильно введенных данных, поступающих от пользователя.

Обычно ошибки появляются на входе программы по трем причинам:

1) неверно заданы входные данные. При наличии хороших средств анализа правильности входных данных эта ошибка может быть сведена к минимуму. Такая ситуация получила название "мусор на входе — мусор на выходе";

2) допущены логические ошибки в программе, в том числе случайные;

3) неправильно выбран алгоритм. Тестирование позволяет определить ошибки на этапе выполнения. Однако можно избежать этих ошибок.

Верификация — процесс проверки требований, документации и проекта программы и частей программы на этапе проектирования программы с целью выявления ошибок, несогласованностей и неточностей. *Тестирование* — это часть процесса верификации программы, относящаяся к выявлению ошибок во время функционирования программы. Однако лучшим тестированием любой системы является ее эксплуатация. Только в результате достаточно

длительной эксплуатации диалоговой системы в реальных условиях можно устранить ошибки в программах.

Писать программы без ошибок трудно, так как любая хорошо оттестированная и проверенная система может содержать необнаруженные ошибки. Даже если не принимать во внимание личностный фактор и забыть пословицу, что "человеку свойственно ошибаться", все равно остаются проблемы сложности системы в целом, несогласованности в требованиях к отдельным программам, неверные входные данные и т.п. Однако ошибки можно свести до минимума, если учитывать определенные факторы и следовать требованиям, выработанным в процессе написания систем различной сложности.

Рассмотрим более подробно причины появления ошибок во входных данных.

Прежде всего система должна быть защищена от неверных входных данных, т.е. содержать программы анализа данных на правильность. Тестирование этих "анализаторов" осуществляется путем ввода заведомо неверных данных.

В системе должна быть предусмотрена диагностика ошибок ввода, т.е. анализаторы в зависимости от той или иной ошибки во входных данных должны выдавать сообщения.

Существенно, чтобы эти сообщения были достаточно конкретными и понятными для пользователя. Действительно, сообщение: «Мне кажется, что у вас ошибка» — мало информативно.

При тестировании следует добиваться, чтобы анализаторы выдавали все сообщения об ошибках, т.е. во входных тестовых данных должны присутствовать комбинации ошибок, приводящие к выводу всех сообщений об ошибках. Тестовые данные — это набор данных, предназначенный для проверки правильности обработки входных данных и содержащий заведомо неверные данные для проверки программ анализа.

Логические ошибки встречаются во всех стадиях реализации диалоговой системы: при разработке спецификаций и сценария диалога, при проектировании и программировании. Установлено, что 70—80% ошибок может быть обнаружено после тщательного просмотра текста за столом, так как трудно найти ошибки за экраном дисплея.

В условиях быстрого развития вычислительной техники дисплеи станут доступными для всех. В то же время

постоянная работа за дисплеем наносит вред программисту. Работа за экраном стимулирует принятие быстрых, но зачастую недостаточно обдуманных решений, что приводит к увеличению общего времени отладки программ. Для обнаружения логических ошибок необходимо использовать листинг программы.

Наиболее распространенными логическими ошибками в программах являются следующие:

— неправильное задание начальных значений переменных;

— неправильная организация циклов, в том числе несвоевременный выход из цикла;

— ошибки во внутренних данных программы, в типах данных;

— пропуск одного или нескольких условий при организации ветвлений;

— неверные логические условия в ветвлениях;

— неверно вычисляемые значения;

— искажение значений переменных.

К логическим ошибкам проектирования также относится выбор неправильного или не совсем точного алгоритма, несоответствие в соглашениях о связях между программами и т.д.

Все эти ошибки должны быть обнаружены на уровне тестирования отдельных модулей и системы в целом. Если ручная проверка не позволяет больше обнаружить ошибки, то при тестировании используется ЭВМ. Задачей тестирования является обнаружение ошибок, а не демонстрация правильности работы программы. Желательно, чтобы тестовые данные подготавливал человек, не принимавший участия в реализации программы.

Процесс тестирования модуля включает в себя следующие этапы:

— тестирование путей прохождения модуля;

— проверку циклов;

— проверку ветвлений;

— проверку правильности внутренних переменных.

Большую помощь в тестировании модуля оказывает прием фиктивного выполнения программы. В применении к диалоговым системам этот прием называется "фиктивным" сеансом работы. Во время такого сеанса действие пользователя (его часть диалога с машиной) заменяется тестовым файлом, заранее подготовленным в соответствии со сценарием тестирования. Если заранее разработать

сценарий тестирования модулей и системы в целом, то существенно упростится процесс тестирования.

5.5. Внедрение диалоговой системы

Внедрение диалоговой системы — один из наиболее сложных этапов реализации, так как на этом этапе разработанная система становится доступной конечному пользователю, т.е. человеку или группе людей, для которых собственно диалоговая система и создавалась. Именно здесь и выявляется несоответствие желаемого и действительного. Пользователь хотел видеть одно, а получил другое или, точнее, не совсем то, что хотел. Объясняется это различием во взглядах разработчиков системы и ее пользователей на функциональное назначение системы. При этом разработчик и пользователь говорят на одном языке, но в одни и те же слова вкладывают различные значения. То, что кажется простым и естественным разработчику диалоговой системы, может оказаться непонятным и недостижимым для пользователя.

Обычно такая ситуация возникает в том случае, когда конечный пользователь не участвует в процессе проектирования системы и разработки сценария диалога. Все сказанное относится и к разработке диалоговой системы для коллективного пользования. Если разрабатывать систему только для себя, то таких трудностей не возникает. Однако большинство диалоговых систем имеет конечных пользователей. Поэтому диалоговая система должна содержать средства обеспечения контакта с пользователем. Такие системы называют *системами, дружественными пользователю.*

Успех внедрения диалоговой системы во многом зависит от ее возможностей по адаптации к требованиям пользователя. Это означает, что система должна содержать развитую систему диагностики ошибок ввода и по мере необходимости давать "подсказки" и помощь пользователю. Если необходимо было ввести букву О, а вместо нее пользователь набрал цифру 0, либо наоборот, то система должна вывести подробное сообщение об ошибке.

Если в диалоговой системе развиты средства обеспечения контакта и подсказок, то пользователь все равно стремится получить напечатанную документацию по системе, так как не все можно решить, сидя за терминалом. Но даже и при работе за терминалом справочное руководство по системе оказывает существенную помощь.

Большое значение во внедрении имеет правильно составленная и достаточно подробная документация. Документация пользователей призвана служить двум целям: для справок и для обучения. Особенно большим спросом пользуется справочное руководство, которое в краткой форме описывает возможности системы. Обычно справочное руководство содержит вводный раздел, инструкции и приложения. Вводный раздел дает представление о функциях и назначении системы, а также о ее пользователях. Возможно, что после прочтения вводного раздела "случайный" пользователь откажется от работы с диалоговой системой.

Основной раздел справочного руководства содержит инструкции и описания процедур, подлежащих выполнению пользователем.

Приложения, помещаемые в конце, содержат "выжимку" из текста и перечень команд управления системой, требования к формам вводимой пользователем информации и т.п.

Рассмотрим основные работы, проводимые на этапе внедрения.

1. Планирование перехода на новую систему (новая система заменяет старую, а если раньше никакой системы не было, то вполне естественно рассматриваемую систему называть новой):

   1.1. Разработка методики перехода на новую систему, т. е. описание совокупности организационно-технических средств обеспечения перехода на новую систему.

   1.2. Разработка инструкций для пользователей и сотрудников, сопровождающих систему.

   1.3. Разработка методов контроля функционирования системы (на основе сформированных контрольных примеров и файлов данных).

2. Ввод новой системы в действие.

   2.1. Испытания системы (они должны быть всесторонними).

   2.2. Обучение пользователей системы и сопровождающего персонала.

3. Контроль за функционированием системы, исправление ошибок, обнаруженных на этапе внедрения.

Все перечисленные работы характеризуют промышленный подход к процессу внедрения. Рассмотрим пример другого подхода. Предположим, что этап внедрения не планируется вследствие каких-либо причин. При этом

обычно составляется минимальная документация и система вводится в действие для пользователей. В результате такого "кустарного" подхода системой не будут пользоваться или ею будет пользоваться только ограниченный круг пользователей, которые смогут разобраться в системе. К сожалению, такой подход еще достаточно распространен и наносит большой вред.

Альтернативой является промышленный подход, основанный на всестороннем планировании и организационно-технической поддержке внедрения системы. Только решение комплекса проблем, возникающих при внедрении диалоговой системы, обеспечит создание необходимых условий для успешного ее функционирования.

Рассмотрим особенности внедрения диалоговых систем.

В системах пакетной обработки посредником между пользователем и машиной является оператор ЭВМ. В условиях диалоговой системы задачи ввода данных и отбора выходных результатов перемещаются от оператора к пользователю. При этом возрастает роль документации и особенно ее учебного воздействия на пользователя. Задача усложняется тем, что можно лишь приблизительно оценить квалификацию пользователя диалоговой системы, в то время как требования к операторам ЭВМ достаточно однозначны. Широко распространено идеализированное представление о возможностях ЭВМ: достаточно подойти к дисплею, нажать несколько клавиш и "умная" машина сразу же выдаст желаемый результат. В действительности, конечно, все обстоит иначе. Чтобы "заставить" ЭВМ работать, нужно изучать соответствующие языки программирования.

Тенденцией развития информатики является создание новых языков общения пользователей с вычислительными машинами, максимально приближенных к естественным. Пользователь сможет общаться с ЭВМ на уровне постановки задачи, ему не нужно будет при этом иметь специальных навыков в программировании. В результате сфера внедрения диалоговых систем существенно расширится.

5.6. Сопровождение

Внедрение новой диалоговой системы является необходимым, но недостаточным условием ее функционирования. Система может содержать ошибки или не

соответствовать в какой-то степени постоянно меняющимся требованиям.

В первом случае ошибки нужно устранять. Сначала этим занимаются разработчики диалоговой системы. Затем назначается человек, ответственный за сопровождение системы в данной организации. Он не является разработчиком системы и у него возникают определенные трудности в процессе ее сопровождения.

Большое значение для успешного сопровождения системы имеет сопроводительная документация. Наличие хорошо составленной документации позволяет сопровождающему систему программисту достаточно быстро разобраться в структуре системы и устранить обнаруженные ошибки.

Во втором случае имеется в виду развитие системы. Действительно, через некоторое время система может уже не соответствовать быстро меняющимся требованиям. Развитие системы существенно упрощается, если она является открытой. Открытая система содержит средства обеспечения наращивания функциональных возможностей. Обычно эти средства включают в себя программы, ориентированные на функционирование в среде открытой системы и обеспечивающие пользователю или сопровождающему программисту возможность написания диалоговых (или пакетных) программ и включения их в состав системы. Такие средства позволяют вводить в систему новые команды.

Развитие системы предполагает усовершенствование существующих модулей и написание новых с целью последующего включения их в состав системы.

Если существующая система является закрытой, то включение в ее состав новых модулей практически невозможно, поэтому при проектировании системы необходимо закладывать возможности ее расширения.

Однако процесс сопровождения заключается не только в устранении ошибок и развитии. Сопровождающий систему персонал должен выполнять следующие функции:

— поддержание системы в работоспособном состоянии;

— контроль за функционированием системы;

— администрирование системы.

Первые две в основном предполагают следующее:

— копирование (сохранение) программного обеспечения и данных системы на резервном магнитном диске или магнитной ленте;

— периодическую проверку правильности функционирования системы на тестовых примерах и задачах;

— наблюдение за текущим состоянием системы и фиксацию обнаруженных ошибок в специальном журнале;

— фиксацию рекламаций пользователей и их пожеланий по усовершенствованию системы в специальном журнале.

Специальный журнал имеет графы, включающие дату и время событий, описание событий, фамилию пользователя и сопровождающего сотрудника, сделавшего запись в журнале. В данном случае под событием понимается: обнаружение ошибки при работе системы, пожелание пользователя, рекламации пользователя.

Обычно все три функции совмещает и выполняет администратор системы — человек, ответственный за функционирование системы в целом.

Администрирование системы предполагает выполнение следующих функций:

— введение новых пользователей в систему. Новые пользователи регистрируются в памяти системы под своей фамилией. Они сообщают администратору свой пароль доступа к системе. Администратор присваивает новому пользователю приоритет или уровень доступа к тем или иным функциям системы. Кроме фамилии и пароля обычно указывается группа или подразделение, где работает пользователь, телефон, пол, дата рождения (система может поздравлять пользователя с днем рождения и в качестве подарка выделить ему дополнительное место на диске или временно — на один день — повысить приоритет доступа);

— распределение ресурсов системы между пользователями. Администратор должен не только распределять ресурсы, но и разрешать конфликтные ситуации по ресурсам между пользователями системы;

— сбор и обработку информации о функционировании системы. Функции сбора и обработки информации могут быть автоматизированы и могут входить в состав системы. Результаты работы этих средств просматриваются администратором системы и служат исходной точкой для принятия решений по распределению ресурсов между пользователями, перераспределения нагрузки на систему и т.п.;

— восстановление работоспособности системы после аварийного сбоя с ее резервной копии на магнитной ленте или другими способами. (Естественно, что администратор

системы должен следить за созданием необходимого числа резервных копий).

Как видно, процесс сопровождения системы — достаточно трудоемкий и ответственный этап. В диалоговых системах сопровождение особенно актуально, так как пользователи обычно широко используют возможности системы, выполняют различные действия за терминалом, в частности изменяют и добавляют значительные объемы информации. В таких условиях надежное сопровождение позволяет избежать потери информации, а правильное планирование ресурсов системы во время ее функционирования помогает поддерживать постоянные технические характеристики системы и избежать конфликтных ситуаций.

5.7. Документирование

Многие программисты считают, что гораздо легче писать программы, чем документировать их. Некоторые программы не имеют документации либо имеют плохую документацию. Ситуация эта аналогична той, когда покупатель делает покупку в магазине какого-нибудь электроприбора без инструкции по эксплуатации. В этом случае он пробует его подключить, нажимает различные клавиши, возможно в конце концов включает его и в результате создает свою инструкцию, только устную. Если же прибор сгорает при включении, то покупатель идет в магазин и требует инструкцию. К сожалению, редко имеется возможность добиться подробной инструкции к программе.

Но допустим, что инструкция по эксплуатации электроприбора приложена, но не соответствует действительности. Покупатель этого, конечно, не знает и, строго следуя пунктам инструкции, вдруг обнаруживает, что прибор сгорает. Он думает, что ему попался брак. Идет в магазин. В магазине тоже не знают, что в инструкции ошибка. Заменяют товар. Покупатель повторяет все согласно инструкции, и прибор снова сгорает... В реальной жизни такая ситуация маловероятна, так как прибор проверяется на заводе. И все же в применении к программам такая ситуация не только возможна, но и довольно типична. Плохо, когда программа не имеет документации, но еще хуже, когда она есть, но не соответствует действительности.

Конечно, программы не сгорают, как приборы, но они перестают функционировать либо функционируют, но не так, как это написано в документации. Главная причина заключается в том, что программирование как техническая дисциплина долгое время развивалась, когда стандартов еще не было. Написание программы считалось элементом творческим, а следовательно, не подчиняющимся стандартам. Действительно, программист в некотором смысле писатель. Он тоже работает с ручкой и листом бумаги (а часто и просто за дисплеем). Он творит свою программу. Он ее отлаживает, тестирует. Она правильно выполняется! А документацию можно рассматривать как мемуары. Ведь если провести аналогию с писателем, то он тоже сначала пишет книги. А потом на творческих вечерах, в мемуарах он описывает методы своей работы, условия ее, как и откуда он брал информацию из окружающего мира, и перед нами постепенно возникают контуры такого сложного, многогранного явления, как стиль писателя.

Долгое время считалось, что стиль каждого программиста глубоко индивидуален. Это подтверждается на практике, так же как и то, что программисту свойственны такие черты, как увлеченность, эмоциональность, азарт, самоотверженность. Все это верно, так и должно быть. Но существенным является то, что стиль программиста заключается не в отклонении от стандартов, а в следовании им. Только в этом случае его программы будут понятными для других, потому что стиль правильного программирования предполагает тщательное документирование всех этапов жизненного цикла программы.

Возвращаясь к примеру с электроприбором, можно сказать, что программа может продаваться в магазине только в том случае, если она обладает характеристиками товара. А главная характеристика товара — это его доступность для потребителя. По отношению к программам характеристикой товарной программы является наличие правильно составленной документации, соответствующей действительности.

Программный продукт — это программное изделие, соответствующее промышленным стандартам. К нему предъявляются все требования изделия промышленности.

Промышленный подход предполагает тщательное документирование системы на всех этапах ее реализации. Целью настоящей главы является не изложение промышленных стандартов на документацию (для этого существует специальная литература). Поэтому рассмотрим общую

схему документирования на примере реализации диалоговой системы:

1. Исходные требования должны быть согласованы с будущим пользователем диалоговой системы. Они включают в себя также перечень желаемых технических характеристик системы (время реакции, число обслуживаемых терминалов, типы используемых диалогов и т.д.). Исходные требования документально оформляются в виде технического задания на разработку системы.

2. Проектная документация включает в себя проект системы, описания программ и данных. Проект системы содержит описание структуры системы и выполняемых ею функций. Описания программ включают в себя спецификации межпрограммных интерфейсов, описания схемы взаимодействия программных модулей между собой, описания алгоритмов, схемы или другие формы документирования алгоритмов (таблицы решений, псевдокод). Описания данных включают документацию по организации данных в будущей системе.

3. Различные справочные пособия и руководства, необходимые для управления и пользования системой, включают в себя руководство пользователя, руководство системного программиста и руководство оператора.

Руководство пользователя — наиболее тщательно написанный документ, включающий в себя подробное описание функционирования системы, команд управления, сообщений об ошибках.

Руководство системного программиста включает в себя описание установки (внедрения) системы, особенностей ее функционирования и процедур обслуживания, т.е. необходимую для сопровождения системы документацию.

Руководство оператора — документ, описывающий действия, требуемые для запуска системы и поддержания ее в работоспособном состоянии в течение сеанса работы с пользователем.

4. Документация по реализации системы включает в себя собственно исходные тексты всех программ, листинги и другую машинную информацию, выдаваемую в процессе реализации, результаты тестирования системы.

Исходные тексты программ оформляются в соответствии с единым стандартом, который устанавливается перед началом реализации системы. Этот стандарт содержит требования к оформлению пролога ("шапки") программы, расположению исходных строк и комментариев к различным частям программы.

Наиболее важная информация, выдаваемая машиной, обязательно включается в состав документации.

Результаты тестирования системы в виде распечаток тестовых данных и листингов тестовых прогонов также включаются в документацию.

5. Документация по сопровождению системы кроме руководства системного программиста, описанного выше, включает в себя журнал регистрации обнаруженных при работе системы ошибок и изменений, внесенных в систему.

Вопросы для самопроверки

5.1. Дайте определение спецификации программы.

5.2. Разработайте сценарий небольшого диалога, предполагающего выяснение личности пользователя. В процессе диалога дайте пароли различного уровня доступа, основанные, например, на дне и годе рождения, теме занятий и т.п. Задача сценария — минимизировать возможность несанкционированного доступа к системе. Сценарий оформите в виде спецификаций на разработку соответствующей диалоговой программы.

5.3. Зачем необходимы диагностические сообщения? Какие диагностические сообщения являются наиболее полными?

5.4. Применяете ли вы метод структурного программирования при написании программ? В чем его отличительные особенности?

5.5. Дайте определение структурного кодирования.

5.6. Чем отличается верификация от тестирования?

5.7. Что является решающим фактором при внедрении диалоговой системы?

5.8. Приведите особенности вашего стиля программирования.

5.9. Чем отличается программный продукт от других изделий промышленности материальной сферы производства?

5.10. Сколько страниц документации должно приходиться на одну страницу программы?

Глава 6

Перспективы диалога

6.1. Интеллектуальный терминал

Развитие средств общения человека с ЭВМ — одно из основных направлений развития вычислительной техники. И если раньше ЭВМ не были приспособлены к человеку, то сейчас изготовляются ЭВМ и программное обеспечение, максимально адаптивные к конечному пользователю.

Простота общения при постоянно возрастающей функциональной мощности диалоговых систем — вот главное требование к системам "человек — ЭВМ". Упрощение управления работой диалоговой системы позволит существенно расширить круг ее пользователей. Пользователь — не профессионал в области вычислительной техники — сможет работать за терминалом.

Одним из основных направлений развития средств общения с ЭВМ является создание искусственного интеллекта — комплекса аппаратных и программных средств, обеспечивающих решение сложных логических задач принятия решений. Взаимодействие с ЭВМ, наделенной интеллектуальными свойствами, предполагается на естественном языке с использованием средств звукового ввода и синтеза голоса. В сочетании со средствами машинной графики это дает возможность в процессе общения изображать на экране дисплея объекты реального мира, соответствующие решаемым задачам.

Широко вошли в обиход диалоговые операционные системы, ориентированные на пользователя. Ими оснащены все современные мини- и микроЭВМ. Существуют такие системы и на ЕС ЭВМ. Пользователю в диалоговой операционной системе настолько доступны сервисные средства общения, что процесс написания диалоговых программ

многократно ускоряется. Диалоговая операционная система представляет собой функционально законченную систему, она повышает уровень интеллектуальности общения с пользователями за счет включения широких сервисных средств обеспечения диалога.

К таким средствам, в частности, относятся инструментальные системы автоматизации построения диалоговых систем, которые обеспечивают создание диалоговых интерфейсов с проблемно-ориентированными программами пользователей. Например, многие операционные системы содержат средства расширения диалоговых возможностей (специальные языки обеспечения диалога, системы заполнения меню, таблиц и т.д.).

Дальнейшая интеграция средств вычислительной техники привела к созданию сетей ЭВМ — от нескольких ЭВМ до сотен и тысяч ЭВМ, соединенных между собой средствами коммуникации. Информация свободно передается по каналам сети и становится неотъемлемой характеристикой жизни общества. Компьютеризация общества — это новая ступень в развитии научно-технического прогресса человечества.

В настоящее время существуют две основные формы взаимодействия пользователя с ЭВМ: через терминал, подключенный к центральной ЭВМ, одновременно обслуживающей много пользователей, через терминал, подключенный к микропроцессору (в составе микроЭВМ).

Для второй формы взаимодействия характерно использование диалоговой операционной системы, ориентированной на одного пользователя. Кроме того, в распоряжении пользователя находится вычислительная машина, которая работает только на него, что очень удобно, когда пользователь общается с диалоговой операционной системой, имеющей достаточно высокий уровень сервиса. Это обусловило такое широкое распространение персональных компьютеров, которые стали неотъемлемой частью процесса обработки данных на многих предприятиях и в учреждениях.

Терминал, подключаемый к микропроцессору, нельзя рассматривать в отдельности от последнего. В совокупности с устройством внешней памяти они образуют микроЭВМ, функционирующую под управлением диалоговой операционной системы.

МикроЭВМ способна решить достаточно широкий спектр различных задач.

В то же время ресурсы микроЭВМ ограничены. Она имеет сравнительно небольшой объем оперативной и внешней памяти, не всегда удовлетворяющее быстродействие (хотя есть и супермикроЭВМ). Однако обычно терминал, подключенный к большой ЭВМ, используется только для ввода и отображения информации. Диалоговая система, обслуживающая терминал, обеспечивает выполнение с его помощью множества различных функций, но только через большую ЭВМ. В то же время, большая ЭВМ обладает и большими ресурсами. Большой объем внешней памяти позволяет создавать централизованную базу данных, в которой накапливаются значительные объемы информации. Центральная ЭВМ в настоящее время способна работать с быстродействием в несколько миллиардов операций в секунду.

Таким образом, обе формы взаимодействия пользователя с ЭВМ имеют достоинства и недостатки, поэтому возникла идея их совмещения, т.е. подключения терминала со встроенным микропроцессором к большой ЭВМ по каналам связи. В результате повышается степень интеллектуальности подключенного к центральной ЭВМ терминала, так как с помощью микропроцессора он может выполнять гораздо большие функции независимо от центральной ЭВМ.

Интеллектуальный терминал — это терминал со встроенным микропроцессором (или микроЭВМ), подключенный к центральной ЭВМ по каналам связи. Выгоды от использования интеллектуального терминала очевидны. Во-первых, он снимает нагрузку с центральной вычислительной системы, так как большую часть функций выполняет на микропроцессоре независимо, при этом сеансы связи с центральной ЭВМ имеют эпизодический характер. Во-вторых, с интеллектуального терминала доступны все ресурсы центральной ЭВМ. Рассмотрим различные режимы работы с интеллектуальным терминалом.

Дистанционная передача файлов. Ограниченные ресурсы микроЭВМ не позволяют обрабатывать на ней большие объемы информации. Использование микроЭВМ в качестве интеллектуального терминала позволяет использовать внешнюю и оперативную память центральной ЭВМ для хранения данных, подготовленных или обработанных на интеллектуальном терминале. Рассмотрим сеанс работы

интеллектуального терминала в режиме дистанционной передачи файлов.

В начале сеанса пользователь работает в режиме микроЭВМ. Он вызывает с помощью диалоговой операционной системы, под управлением которой работает микроЭВМ, текстовой редактор для подготовки данных. МикроЭВМ в это время отключена от центральной ЭВМ и работает только на пользователя. Через некоторое время пользователь подготавливает данные. Допустим, это программа на языке ФОРТРАН, предназначенная для выполнения на центральной ЭВМ. Программа оформляется в виде файла, который затем передается для выполнения на центральную ЭВМ. Для этого пользователь с помощью специальных средств посылает сигнал запроса на связь с центральной ЭВМ, которая, обработав этот сигнал, осуществляет перепись подготовленного файла на быструю внешнюю память на дисках, т.е. осуществляется дистанционная передача файла.

При этом с интеллектуального терминала могут прийти дополнительные указания о том, что сделать с передаваемым файлом: запомнить в определенной библиотеке, запустить на выполнение, преобразовать, отсортировать и т.п. Результаты посылаются обратно в интеллектуальный терминал в виде одного или нескольких файлов, которые сразу или позже просматриваются на экране терминала. Дистанционная передача файлов позволяет центральной ЭВМ и интеллектуальному терминалу обмениваться большими порциями информации в виде файлов.

Эмуляция терминала центральной ЭВМ. Терминал, используемый в микроЭВМ, обычно отличается от терминала центральной машины. Поэтому, когда происходит подмена, возникает проблема использования интеллектуального терминала в качестве обычного терминала центральной ЭВМ. Это достигается с помощью моделирования, эмуляции терминала центральной ЭВМ. Эмуляция терминала центральной ЭВМ обеспечивается совокупностью программно-аппаратных средств, входящих в состав интеллектуального терминала. Эмуляция позволяет работать как с диалоговой системой центральной ЭВМ (при этом диалоговая система работает как бы с обычным терминалом), так и в режиме микроЭВМ.

Использование интеллектуальных терминалов позволяет повысить производительность пользователя и вычисли-

тельной системы в целом. Сеть интеллектуальных терминалов, подключенных к мощной центральной ЭВМ — такова типичная конфигурация вычислительных средств предприятий и организаций в ближайшем будущем.

6.2. Диалог в сети ЭВМ

Достижения микроэлектроники позволили создать недорогие, но производительные и относительно небольшие вычислительные машины (микроЭВМ) персонального пользования. Эти ЭВМ обладают быстродействием примерно в 1 млн. операций в секунду и внешней памятью до нескольких десятков мегабайт. Поэтому многие задачи, связанные с переработкой информации, получаемой на одном рабочем месте, успешно решаются с помощью таких персональных ЭВМ. Однако при необходимости оперировать большими объемами информации, совместной работе нескольких пользователей с банком оперативно меняющейся информации и потребности большей производительности вычислительной машины одной микроЭВМ недостаточно. В этом случае может оказаться целесообразным создание вычислительной сети ЭВМ — территориально рассредоточенных, но информационно связанных между собой средств вычислительной техники, используемых в комплексе для сбора, хранения и переработки информации.

Пользователи сети ЭВМ, а также подключенные к ней терминальные устройства называются *абонентами*.

Типы сетей отличаются составом и удаленностью средств вычислительной техники, используемыми средствами связи и др. В частности, многомашинные комплексы также можно рассматривать как сеть ЭВМ.

Ознакомимся с основными принципами работы и некоторыми типами сетей.

Организация связи в сети ЭВМ невозможна без надежной передачи информации. Из-за высокой тактовой частоты работы вычислительных машин обмен информацией без преобразования в естественном виде требует дорогостоящих высокочастотных кабелей. Но даже по таким кабелям дальность надежной прямой связи между ЭВМ не превышает нескольких километров. Для обеспечения более дальней связи принято использовать имеющиеся телеграфные и телефонные сети.

Стандартная частота телеграфного (50 Гц) и телефонного (3,1 кГц) каналов обусловлена возможностями по передаче и приему информации соответственно телеграфных аппаратов и человека (его голоса и слуха). ЭВМ является принципиально другим оконечным устройством со значительно большей тактовой частотой работы. Поэтому требуются специальные устройства сжатия (на передающем конце линии связи) и восстановления (на приемном конце) машинной информации. Ими могут служить специальные устройства модуляции и демодуляции сигнала — *модемы*. Приняты различные стандарты скорости передачи информации: от 50 бод (бит/с) для низкоскоростных модемов до десятков килобод для высокочастотных.

Вычислительные машины и отдельные группы терминалов подключаются к сети ЭВМ через *коммутационные процессоры*, которые располагаются в узлах сети. Эти процессоры коммутируют линии связи, преобразуют информацию и управляют ее прохождением по линиям связи сети ЭВМ.

Возможны два способа коммутации абонентов:

1) коммутация каналов — организация на все время обмена информацией постоянного канала связи между абонентами сети. При этом из-за малой скорости передачи информации (особенно с терминалов) канал может оказаться недогруженным;

2) коммутация сообщений — обмен между абонентами сообщениями, временно запомненными в узлах сети.

Второй способ обеспечивает гораздо более экономное по сравнению с коммутацией каналов использование дорогостоящих каналов дальней связи. Однако он имеет и недостатки: при изменении длины сообщений в широком диапазоне память коммутационных процессоров рассчитывается на максимальную длину, что увеличивает стоимость этой памяти. Поэтому сообщения разбиваются на пакеты стандартной длины, которые передаются независимо друг от друга и зачастую по разным линиям. Таким образом, появляются дополнительные задачи сети ЭВМ: разборка сообщений на пакеты и сборка из них сообщений, оптимизация маршрутов передачи пакетов в зависимости от загрузки элементов сети.

Для согласования работы различных элементов сети ЭВМ (вычислительных машин, коммутационных процессоров, линий связи) служат сетевые протоколы — основные

задачи управления и стандарты представления информации в сети ЭВМ. В соответствии с рекомендациями международной организации по стандартам ИСО для обеспечения совместимости различных сетей ЭВМ необходимо реализовать семь уровней связных протоколов.

Прикладной уровень обеспечивает доступ к ресурсам любой прикладной программы, находящейся в сети ЭВМ.

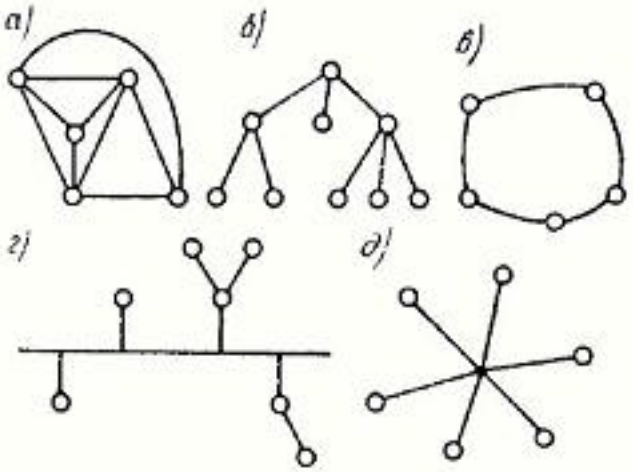

Рис. 6.1

Представительный уровень преобразует прикладную информацию в удобную для транспортировки форму.

Сеансовый уровень определяет средства, необходимые для распределения ресурсов прикладных процессов, синхронизации диалога и управления процессом обмена данных.

Транспортный уровень поддерживает реализацию сеансового уровня с помощью управления процессом транспортировки данных, обеспечивает оптимизацию использования ресурсов следующих (нижних) уровней протокола.

Сетевой уровень поддерживает транспортный уровень, обеспечивает независимость маршрута сообщений от предыдущего уровня и оптимизирует маршруты передачи данных.

Канальный уровень поддерживает сетевой уровень и обеспечивает организацию, поддержание и разъединение

связей, упаковку сообщений, обнаружение и корректировку ошибок и т.д.

Физический уровень определяет физические способы соединения терминалов абонентов в выбранной передающей среде с учетом ее конкретных физических характеристик.

Возможны различные конфигурации подключения терминалов и ЭВМ, называемые *топологией сети* (рис. 6.1): *а* — произвольная (сетевая), *б* — иерархическая, *в* — кольцевая; (цепочная, эстафетная), *г* — типа "общая шина", *д* — типа "звезда".

Терминальная сеть является примером простейшей сети ЭВМ и состоит из вычислительного центра и удаленных терминалов. В этом случае его называют *вычислительным центром коллективного пользования* (ВЦКП). Удаленность терминалов от вычислительного центра при этом не превышает, как правило, нескольких километров.

ВЦКП целесообразно строить на базе достаточно мощной ЭВМ или многомашинного вычислительного комплекса. Территориально близкие друг к другу терминалы могут группироваться для подключения к центральной ЭВМ через терминальные концентраторы, в качестве которых могут использоваться даже миниЭВМ в случае предварительной обработки информации.

В качестве терминалов используются различные средства и комплексы вычислительной техники:

— простейшие терминалы (телетайпы, алфавитно-цифровые дисплеи, телевизоры с клавиатурой и др.);

— интеллектуальные терминалы на базе микроЭВМ;

— абонентские пункты, в состав которых могут входить устройства внешней памяти, выполненные на магнитных лентах и дисках, комплексы устройств ввода-вывода;

— автоматизированные рабочие места проектировщиков и работников аппарата управления с периферийными устройствами.

ВЦКП работают в режимах разделения времени или удаленного ввода пакетных заданий. В этих режимах наряду с сервисом, предоставляемым системой на рабочем месте абонента, достигаются хорошие результаты использования машинных ресурсов ВЦКП.

Примеры сетей ЭВМ характеризуют разнообразие их возможностей и сфер применения. В мире разработаны и действуют несколько тысяч сетей ЭВМ примерно ста различных типов.

Первые *коммерческие иностранные сети ЭВМ* были созданы и начали работать в конце 60-х годов. В их числе сети Cubernet, ARPANET и сеть фирмы General Electric (США). Абоненты из США, Европы и Японии имеют возможность подключаться к информационно-вычислительным ресурсам этих сетей через кабели (в том числе трансатлантический кабель) и через стационарные спутники связи. Услугами только этих сетей пользуются десятки тысяч абонентов.

Среднее время задержки пакета в сетях такого типа составляет десятые доли секунды, а вероятность ошибки при его передаче не превышает 10 в минус десятой на бит. В настоящее время такие сети продолжают развиваться в направлении как улучшения их технических характеристик, так и предоставления абонентам новых информационных услуг.

Широкое распространение в СССР и за рубежом получили различные *специализированные сети*, к которым относятся: банковские сети, позволяющие проводить электронные расчеты между абонентами территориально удаленных друг от друга банков, сети по продаже авиационных и железнодорожных билетов.

Наиболее интенсивно развиваются локальные сети, объединяющие терминалы и ЭВМ одной организации, предприятия для обеспечения информационных и вычислительных услуг. Эти сети объединяют территориально близких абонентов, находящихся в пределах нескольких километров друг от друга. Их отличает большое разнообразие технических решений ввиду широкой номенклатуры используемых средств вычислительной техники, а также сфер применения этих сетей.

Наиболее распространены и известны следующие локальные сети ЭВМ: ARCnet, Ethernet, Net/one, Z-NET, iLNA. Для локальных сетей характерна аппаратная реализация протоколов нижних уровней, а также отсутствие некоторых уровней протокола, что повышает надежность и сокращает стоимость реализации сети. В последнее время произошел отказ от конфигурации типа "общая шина" (Ethernet, Net/one) в пользу кольцевой топологии локальной сети (iLNA, "Эстафета"). Для повышения быстродействия таких сетей могут использоваться волоконно-оптические шины.

Развитие сетей ЭВМ имеет важное стратегическое значение, особенно развитие национальных и транснациональных государственных сетей. В эпоху научно-технической революции создание единой системы информационного обслуживания является даже более актуальной задачей, чем в свое время создание единой энергетической системы.

В нашей стране ведутся работы по созданию государственной сети вычислительных центров для управления экономикой на всех уровнях. Ее основу должна составить сеть опорных региональных ВЦКП. Для обеспечения требуемой пропускной способности опорные центры должны быть подключены в обход каналообразующей аппаратуры непосредственно к линиям дальней связи с необходимой пропускной способностью. При этом сеть фактически превращается в многомашинный комплекс, объединяя территориально удаленные ЭВМ через их собственные устройства: канальные процессоры и адаптеры канал — канал. Устраняется необходимость многих сетевых протоколов, в первую очередь транспортного. Для уменьшения стоимости связи могут быть использованы существующие телевизионные линии, что приведет к требованию минимизации обмена между опорными центрами в дневное и особенно в вечернее время. Это, в свою очередь, может потребовать перестройки работы государственных органов, активно работающих в сети ЭВМ.

Важнейшей задачей при разработке национальной государственной сети является обеспечение требуемого режима секретности работы с информационной базой. Большое внимание уделяется стандартизации работы сетей и перспективам их сопряжения друг с другом.

Состояние линий связи для обеспечения работы сетей ЭВМ во многом будет определять информационный потенциал государства.

6.3. Развитие средств общения

Основными направлениями развития средств вычислительной техники на ближайшие десятилетия является широкое внедрение на основе достижений микроэлектроники, вычислительной и микропроцессорной техники в большинстве сфер жизнедеятельности человека, а также повышение интеллектуальности этих средств. Оба эти на-

правления требуют развития аппаратных и программных компонентов вычислительной техники.

Развитие аппаратных средств вычислительных машин первого, второго и третьего поколений характеризовалось наличием ярко выраженных интенсивных тенденций. Производительность машин росла примерно пропорционально квадрату их стоимости, т.е. самые большие ЭВМ имели самые низкие затраты на единицу вычислений. Данная тенденция была формализована в виде закона Гроша:

$$W=kS^2,$$

где W — производительность ЭВМ, опер/с; k — некоторая среднестатистическая постоянная; S — стоимость модели ЭВМ.

Для микроЭВМ и вычислительных машин следующих поколений характерно также интенсивное развитие. При наличии соответствующих потребностей это способствует созданию более производительных машин: повышению их быстродействия, росту объема оперативной и внешней памяти, появлению внешних устройств, действующих на новых физических принципах.

Рост производительности ЭВМ достигается:

а) за счет создания новой элементной базы;

б) путем разработки новых архитектурных решений (в том числе на базе многомашинных и многопроцессорных систем).

Принципиальным ограничением деятельности цикла работы элементной базы является скорость распространения электромагнитных волн в пространстве (скорость света). В ЭВМ пятого поколения, построенных на базе криогенной электроники, быстродействие близко к этому ограничению и находится на уровне 10^{10} опер/с. Получить более высокое быстродействие можно только за счет распараллеливания алгоритмов и создания новых принципов организации вычислений.

Объем прямо адресуемой оперативной памяти находится за пределами 2^{24} байт (16 М байт) и ограничивается величиной 2^{32} байт (4 Г байта). Существуют и успешно работают ЭВМ с объемом оперативной памяти в 256 М байт.

В области производства микроЭВМ происходит постепенное наращивание разрядности используемых микропроцессорных наборов (от 4-разрядных через 8- и 16-разряд-

ные к 32-разрядным), а также рост оперативной памяти персональных компьютеров (от 16 К байт до нескольких мегабайт).

Неуклонно растет емкость и сокращается время доступа к внешней памяти как за счет улучшения параметров магнитного слоя накопителей, так и вследствие использования новых физических принципов. Так, разрабатываемые оптические накопители на дисках имеют емкость несколько гигабайт.

Создаются внешние устройства с более совершенными техническими характеристиками (надежность, быстродействие, разрешающая способность). Особо следует отметить применение достижений голографии в вычислительной технике. Недалеко то время, когда дисплей, построенный на принципах голографии, станет столь же обычным явлением, как и цветной телевизор. Использование объемного изображения значительно повысит возможности средств общения с ЭВМ, особенно для целей обучения. И в других сферах искусственного интеллекта будут активно использоваться новые типы внешних устройств: синтезаторы речи, устройства распознавания речи и образов и др. Для вычислительных комплексов в целом сохранится тенденция к повышению доли стоимости внешних устройств и уменьшению доли стоимости центральных устройств.

Дальнейшее развитие получают спутниковая и кабельная связи. При этом следует учитывать, что экономические характеристики связи за время существования ЭВМ улучшались примерно на порядок медленнее, чем характеристики средств вычислительной техники. Поэтому связь и дальше будет сдерживать темпы создания и использования крупномасштабных сетей ЭВМ (региональных, национальных, межнациональных, глобальных).

Развитие программного обеспечения будет связано:

а) с широким внедрением средств вычислительной техники в традиционных и нетрадиционных сферах ее применения;

б) с ликвидацией разрыва, существующего между относительно высокими темпами роста возможностей аппаратных средств и низкими темпами повышения эффективности разработки программного обеспечения ЭВМ.

Решение этих задач взаимозависимо, и одно из направлений их решения связано с повышением уровня интеллек-

туальности программных средств. В будущем разработчикам прикладных программ для их создания достаточно будет провести инструктаж машины в диалоге на языке, близком к естественному. При этом будут активно использоваться речевые способы общения с ЭВМ.

Потребности комплексного применения средств вычислительной техники, особенно в гибких производственных системах, выдвигают на одно из первых мест требование по обеспечению мобильности их программного обеспечения. Это касается разработки как операционных систем, так и новых трансляторов и интерпретаторов.

Важным направлением является повышение гибкости аппаратных средств за счет более широкого использования ПЗУ и ППЗУ для хранения микропрограмм (уровень автоматизации разработки которых также нуждается в существенном повышении).

Тенденция переноса центра тяжести при построении центрального процессора с жесткой аппаратной логики на микропрограммную реализацию ведет к снижению производительности ЭВМ. Поэтому на практике широко используются оба подхода.

В будущем будет продолжаться создание более эффективных компиляторов с языков как в традиционных, так и в новых областях применения.

Активно будет разрабатываться программное обеспечение сетей ЭВМ, в первую очередь локальных и общегосударственных.

Перспективы применения вычислительной техники зависят от способа ее использования. Существует два основных способа использования ЭВМ: автоматический (роботизированное производство, различные контрольно-измерительные системы) и автоматизированный. Диалог человека с ЭВМ ведется, главным образом, при автоматизированном использовании вычислительной техники.

Рассмотрим основные сферы ее автоматизированного применения: в науке, производстве, при управлении организационными системами, в социальной области (образование, здравоохранение и др.). В быту, вне зависимости от сфер применения вычислительной техники выделяют секторы (или подсистемы) системы искусственного интеллекта, которые помогают решать традиционно неавтоматизированные, интеллектуальные задачи. Наибольшее развитие получат следующие (в порядке следования) сек-

торы искусственного интеллекта: экспертные системы, программные средства на естественном языке, распознавание и генерация визуальных образов (в том числе трехмерных) и речи, автоматизированное обучение. При этом в составе каждой из автоматизированных систем будет присутствовать подсистема самообучения.

В науке развиваются многоуровневые системы управления экспериментом на базе ЭВМ с сопрягаемой физической аппаратурой и внешними устройствами (в том числе и голографические дисплеи), различные моделирующие системы, а также пакеты прикладных программ и языки программирования, соответствующие различным научным дисциплинам и используемому в них математическому аппарату. Будут активно проводиться научные телеконференции, использоваться электронная почта.

В производстве активно используются комплексные системы автоматизированного проектирования, автоматизирующие все этапы разработки: от технического задания и эскизного проекта до автоматизации изготовления и контроля серийных изделий. В таких системах гибкие автоматизированные производства могут применяться в качестве одной из подсистем САПР.

При управлении организационными системами в качестве программно-аппаратного обеспечения используются локальные сети ЭВМ, подключенные к региональным, а в дальнейшем — к национальной государственной сети. Состав автоматизированного рабочего места работника аппарата управления дополнится различными вспомогательными устройствами: телефоном с памятью и программным управлением, устройством печати с голоса, накопителем на магнитных дисках типа "винчестер" (несъемный высоконадежный НМД) и др.

Обучение с помощью ЭВМ развивается в направлении повышения адаптивности обучающих систем к индивидуальным особенностям учащихся. При этом активно используются различные новые аудиовизуальные средства. В качестве лабораторного практикума ЭВМ может использоваться в тех же формах, что и в научных применениях.

Как показывает опыт применения ЭВМ в *медицинской диагностике*, эффективность их использования здесь чрезвычайно высока. Поэтому вычислительная техника для предварительной диагностики будет использоваться повсеместно. Постепенно на основе всеобщей диспансеризации

будет создан банк медицинских данных населения, что позволит проводить лечение с использованием вычислительной техники с учетом индивидуальных показаний.

Наиболее быстрыми темпами ЭВМ будут внедряться в различные сферы быта: управление теле- и радиоаппаратурой; обеспечение передачи данных из удаленных банков данных на экран видеотерминала (телевизора), компоновка телепередач и электронных газет в соответствии с интересами пользователя; управление работой бытовых приборов, сокращение объема домашнего труда на основе использования бытовых роботов; повышение уровня безопасности жилища; контроль за состоянием здоровья проживающих; решение задач обучения детей и взрослых; проведение телеконференций; телезакупки и безналичные денежные расчеты и др.

Вопросы для самопроверки

6.1. Перечислите известные вам модели микроЭВМ. Используете ли вы их для вычислений и обработки данных ?

6.2. Каковы преимущества использования сетей ЭВМ ?

6.3. Дайте определение интеллектуального терминала.

6.4. Перечислите основные режимы работы с интеллектуальным терминалом.

6.5. Чем дистанционная передача файлов отличается от эмуляции терминала?

6.6. Что такое сетевые протоколы? Перечислите все уровни связных протоколов.

6.7. Каковы числовые величины стандартных частот телеграфного и телефонного каналов? Почему они имеют такие значения?

6.8. Сформулируйте закон Гроша.

6.9. Что такое искусственный интеллект?

СПИСОК ЛИТЕРАТУРЫ

1. *Девис У.* Операционные системы. Функциональный подход .— М.: Мир,1980.

2. *Попов Э.В.* Общение с ЭВМ на естественном языке.— М.: Наука, 1982.

3. *Зиглер К.* Методы проектирования программных систем.— М.: Мир, 1985.

4. *Хьюз Дж., Мичтом Дж.* Структурный подход к программированию.— М.: Мир, 1980.

5. *Венда В.Ф.* Видеотерминалы в информационном взаимодействии.— М.: Энергия, 1980.

6. *Ван Тассел Д.* Стиль, разработка, эффективность, отладка и испытание программ.— М.: Мир, 1985.

7. *Довгялло А.М.* Диалог пользователя и ЭВМ: основы проектирования и реализации.— Киев: Наукова думка, 1981.

8. *Боэм Б.У.* Инженерное проектирование программного обеспечения.— М.: Радио и связь, 1985.

9. *Уэзерелл Ч.* Этюды для программистов.— М.: Мир, 1982.

10. *Грогоно П.* Программирование на языке Паскаль.— М.: Мир, 1982.

11. *Белоногов Г.Г., Кузнецов Б.А.* Языковые средства автоматизированных информационных систем.— М.: Наука, 1983.

12. *Мартин Дж.* Вычислительные сети и распределенная обработка данных.— М.: Финансы и статистика, 1985.

13. *Сакман Г.* Решение задач в системе "человек—ЭВМ": Пер. с англ.— М.: Мир,1973.

14. *Рыбаков Ф.И.* Системы эффективного взаимодействия человека и ЭВМ.— М.: Радио и связь, 1985.

15. *Вирт Н Систематическое программирование. Введение — М.: Мир, 1977.*

16. *Пиотровский Р.Г.* Текст, машина, человек.— Л.: Наука, 1975.

17. *Шнейдерман Б.* Психология программирования: человеческие факторы в вычислительных и информационных системах: Пер. с англ.— М.: Радио и связь, 1984.

18. *Деннинг В., Эссинг Г., Маас С.* Диалоговые системы "человек — ЭВМ". Адаптация к требованиям пользователя: Пер. с англ. — М.: Мир, 1984.

ОГЛАВЛЕНИЕ

Учебное издание

Электронные вычислительные машины

В восьми книгах

Савченко Александр Валентинович
Сальников Юрий Валентинович
Филиппов Александр Николаевич

Книга 6

Средства общения с ЭВМ

Заведующая редакцией *Н. И. Хрусталева.* Редактор *С. М. Оводова.* Младший редактор *Г. Г. Бучина*, Художник *Ю. Д. Федичкин.* Художественный редактор *Т. М. Скворцова. Технический редактор Л. А. Овчинникова.*

ИБ № 8816

Изд. № СТД-702. Сдано в набор 09.04.91. Подп. в печать 23.08.91. Формат 84×108 1/32. Бум. газетная. Гарнитура Таймс. Печать высокая. Объем 6,72 усл. печ. л. 7,14 усл. кр.-отт. 6,86 уч. изд. л. Тираж 100 000 экз. Зак. № 469. Цена 1р.50к.

Издательство "Высшая школа", 101430, Москва, ГСП-4, Неглинная ул., д. 29/14.

Набрано на персональном компьютере издательства.

Государственная ассоциация предприятий, объединений и организаций полиграфической промышленности «АСПОЛ». Ярославский полиграфкомбинат, 150049, Ярославль, ул. Свободы, 97.