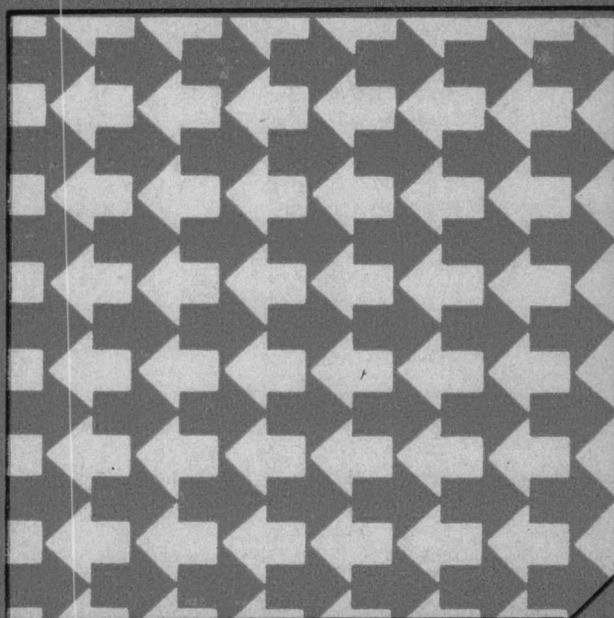


**МАТЕМАТИЧЕСКОЕ  
ОБЕСПЕЧЕНИЕ  
ЭВМ**

---

**Дж. Хенли**

**АВТОМАТИЗИРОВАННАЯ  
БИБЛИОТЕКА  
И ИНФОРМАЦИОННЫЕ  
СИСТЕМЫ**





J. P. HENLEY

**COMPUTER-BASED LIBRARY  
AND INFORMATION SYSTEMS**

MACDONALD·LONDON AND  
AMERICAN ELSEVIER INC.·NEW YORK  
1972

**МАТЕМАТИЧЕСКОЕ  
ОБЕСПЕЧЕНИЕ  
ЭВМ**

---

**Дж. Хенли**

**АВТОМАТИЗИРОВАННАЯ  
БИБЛИОТЕКА  
И ИНФОРМАЦИОННЫЕ  
СИСТЕМЫ**

Перевод с английского

Д. А. Корягина

Под редакцией

В. В. Мартынюка

ИЗДАТЕЛЬСТВО

«МИР»

Москва 1974

В книге известного английского специалиста в области системного программирования Дж. Хенли рассматриваются способы использования вычислительных машин в автоматизированных библиотеках и различных поисковых системах. Автор подчеркивает, что поскольку библиотеки принадлежат к сфере обслуживания, то использование вычислительной машины должно быть организовано так, чтобы полностью учитывались запросы и нужды абонентов. В книге приводятся примеры типичных программ, обеспечивающих обработку информации, а также предлагаются критерии оценки эффективности работы библиотеки.

Книга будет полезна как системным программистам, так и библиотечным работникам. Кстати, для последних в одной из глав дается простое изложение общих принципов организации вычислительной машины.

## ИЗ ПРЕДИСЛОВИЯ К ПЕРВОМУ ИЗДАНИЮ

Эта книга посвящена рассмотрению некоторых путей использования вычислительной машины в автоматизированной библиотечной системе. При этом преследуется цель установить взаимопонимание и общность интересов между специалистами в области библиотечного дела и специалистами по вычислительной технике. Автор стремится облегчить ознакомление с вычислительной машиной читателей, занятых в сфере библиотечной службы, особенно студентов библиотечных учебных заведений, и в то же самое время помочь тем, кто интересуется вопросами машинных приложений, узнать возможности и проблемы библиотечного дела. Одна из первых глав предназначена для читателей, не знакомых с вычислительными машинами; в последующих главах проводится обсуждение логических шагов на пути от автоматизации небольшого числа тривиальных процедур библиотечного обслуживания к крупной автоматизированной информационной системе. В работе отмечаются некоторые средства машинной обработки информации, однако в интересах общности изложения материала не рассматриваются подробности, касающиеся какого-либо средства, конкретного проекта или примера. В качестве иллюстраций проведенных рассматриваний используются несколько примеров программ библиотечного обслуживания; книга заканчивается обсуждением общих основ автоматизированных информационных систем и трудностей, связанных с их разработкой.

*Дж. Хенли*

## ПРЕДИСЛОВИЕ КО ВТОРОМУ ИЗДАНИЮ

Во второе издание этой книги внесены изменения, направленные на более широкое обсуждение ограничений и возможностей современных вычислительных машин и их программного обеспечения в области информационного поиска. Включение в книгу отдельных глав, посвященных машинной аппаратуре, программному обеспечению и методам хранения файлов, преследует цель показать читателю, особенно малознакомому с вычислительной техникой, сложность доступных средств и методов, а также те различные направления, в которых следует ожидать новых разработок, применимых при решении задач информационного поиска.

*Дж. Хенли*

1971 г.

## ЗАЧЕМ НУЖНА АВТОМАТИЗИРОВАННАЯ БИБЛИОТЕЧНАЯ СИСТЕМА

---

### 1.1. ПРОБЛЕМАТИКА ОБРАБОТКИ ИНФОРМАЦИИ

И начал Хелкия, и сказал [Шафану писцу: «Книгу закона нашел я в доме Господнем...» И также донес писец царю, говоря: «Книгу дал мне Хелкия священник». И читал ее Шафан перед царем.

Вторая книга Паралипоменона, гл. 34.

Этот простой пример информационного поиска ярко отражает две характерные черты современного подхода к этой проблеме. Во-первых, исторически информационный поиск всегда лежал у истоков различных социальных преобразований, поэтому и теперь он может оказаться жизненно важным. Во-вторых, прежде не существовало систематического метода поиска; если же общество хочет эффективно использовать хранилища информации, то оно должно предусмотреть некоторую систему, обеспечивающую удобный доступ к этим хранилищам. Библиотека или информационный центр как раз и должны способствовать решению этой задачи.

В настоящее время несколько факторов обуславливают такую ситуацию, в которой становится совершенно очевидным, что ранее использовавшиеся традиционные методы обработки информации являются несовершенными. В первую очередь, это избытие информации вообще, а во-вторых, что более существенно, рост доли той информации, о которой должен быть запасен исчерпывающий комментарий, исключающий необходимость в дальнейших разъяснениях.

В связи с этим теперь сомнительным представляется часто цитируемое утверждение о том, что, хотя при обновлении информационных систем копирование в библиотеках и бывает сопряжено с серьезными трудностями [1], во многих случаях эти трудности представляют собой чисто временное явление. Дело не ограничивается только одним увеличением объема информации, поскольку одновременно с этим столь же быстро возрастает степень специализации накапливаемой информации. Недаром же современная поговорка гласит, что все больше и больше людей теперь изучают все больше и больше о все меньшем и меньшем, требуя в своей повседневной деятельности обстоятельных сведений и текущих справок на все более детальном уровне. Библиотечное или информационное обслу-

живание имеет своей целью удовлетворение указанных требований. В итоге получается, что библиотекарь, этот традиционный поверенный документальных мудростей, знает все меньше и меньше о большем и большем, поскольку возрастающий поток запросов требует от него в первую очередь реализации регистрационных актов.

В свете этих обстоятельств представляется вполне естественным, что документальный поиск и информационный поиск должны развиваться как две независимые дисциплины, которые связаны с деятельностью ученого и дилетанта, инженера и юриста, академика и практиканта. Нет ничего удивительного в том, что большинство библиотечных служащих заинтересовано в тесном сотрудничестве со специалистами по вычислительной технике, поскольку очевидно, что если машина и не является панацеей от всех библиотечных недугов, то она по крайней мере весьма тонизирующее средство.

Эта книга посвящена рассмотрению той роли, которую машина может сыграть в библиотечной или информационной службе. В наших рассуждениях обсуждаются как задачи, которые должны быть при этом решены, так и принципы машинной организации, соответствующие этим различным задачам. Предлагаемые в книге методы рассчитаны на использование в библиотечной службе таких чисто машинных преимуществ, как быстродействие, большой объем памяти и пунктуальность. Обсуждения сопровождаются примерами типичных программ. В книге предпринята попытка установить некоторый критерий для оценки выигрыша или же издержек, связанных с автоматизацией процесса обработки информации.

С электронной вычислительной машиной читатель знакомится в общих чертах; более подробно обсуждаются те особенности современных разработок, которые наилучшим образом способствуют использованию машины в библиотечном деле. Автор стремился подчеркнуть, что библиотека обеспечивает некий сервис, и поэтому если машина будет использоваться как полезная, но вместе с тем и дорогая компонента библиотечного оборудования, то при этом потребности и возможности библиотечного пользователя не должны быть забыты в сумбуре машинной терминологии и технического энтузиазма. Другими словами, машина должна обслуживать библиотеку, а не наоборот.

Основная цель, которую преследует книга, состоит в том, чтобы познакомить сотрудников библиотек с возможностями электронных вычислительных машин, а также с путями практического использования этих возможностей и вместе с тем привлечь внимание специалистов по вычислительной технике к конкретным проблемам библиотечного дела, трудностям, связанным с их решением, и обсудить подходы к преодолению этих трудностей.

## 1.2. ИЗМЕНЕНИЕ ВЗГЛЯДОВ

Здесь уместно упомянуть о том крутом повороте, который произошёл в библиотечных кругах, а также во многих других областях, где возможно использование машин, в отношении к механизации и к документалистам, пытавшимся применять научные методы в службе информации. В марте 1958 г. Бауэр в Вильсоновском библиотечном бюллетене писал [2]:

Документация сейчас в большой моде среди тех добрых душ, которые хотят расширить свою беспорядочную эрудицию, а документалисты стараются, правда, с посредственным успехом превратить свое занятие в научную профессию, изобретая жаргон, насыщенный такими абсурдными терминами, как «информационный поиск», «троп-индексация», «коррелятивная индексация», «неуправляющий индекс» и «координатная индексация».

С некоторыми оговорками, касающимися введения новой терминологии, мы можем сегодня сказать, что полуприкрытое осмеяние и явное недоверие по отношению к механизации библиотек исчезли. На смену им пришли не только сугубо профессиональные дискуссии, подобные той, что состоялась на конференции по библиотечному делу и автоматизации в 1963 г. [4], но и практические исследования, имеющие своей целью описать принципы построения и определить временные характеристики полностью автоматизированной оперативной библиотечной системы, предназначенной для широкого использования в реальных условиях. Некоторые из этих работ будут рассмотрены ниже. Если машина должна стать неотъемлемой компонентой библиотечного дела, то ее громадный потенциал, равно как и неизбежные ограничения, должны быть как следует осмыслены.

## 2

### ПРЕДВАРИТЕЛЬНЫЕ СВЕДЕНИЯ О ВЫЧИСЛИТЕЛЬНЫХ МАШИНАХ

---

#### 2.1. ОСНОВНЫЕ ПРИНЦИПЫ

Эта глава знакомит с электронной вычислительной машиной читателей, не имеющих специальной подготовки по данному вопросу, причем основное внимание здесь уделяется тем аспектам и факторам, которые касаются использования машинной системы в библиотечном деле. Машина рассматривается с точки зрения программиста, другими словами, она понимается как некий электронный черный ящик, который может, если он правильно инструктирован или запрограммирован, выполнять определенные операции. При этом не требуется знания того, каким образом эти операции реализуются с инженерной точки зрения.

В общем виде можно считать, что машинное оборудование или аппаратура состоит из четырех компонент, каждая из которых выполняет свои определенные функции. Во-первых, это оперативное запоминающее устройство (иначе говоря, основная память машины), которое хранит элементы информации до тех пор, пока они необходимы. Эта информация может быть представлена либо в форме команд, принадлежащих программе, которая должна быть выполнена, либо в форме данных, используемых этой программой. Так, например, целесообразно, чтобы при составлении машиной платежной ведомости в ее памяти одновременно хранились как список служащих, заработная плата которых должна быть зарегистрирована, так и сама программа составления платежной ведомости. После того как выполнение программы закончено, хранящаяся в памяти машины информация может быть уничтожена, если, конечно, она не потребуется в дальнейшем.

Следующая компонента аппаратуры — это устройство управления, которое интерпретирует команды программы и посылает управляющие сигналы, инициирующие выполнение требуемых операций. Третья компонента — это арифметическое устройство, или процессор, который выполняет требуемые операции над элементами информации, хранящимися в оперативной памяти машины. Таким образом, для сложения двух элементов информации необходимо чтобы соответствующая команда хранилась в оперативной памяти, была дешифрована устройством управления, а затем фактически выполнена процессором. Наконец, последняя компонента — это ряд устройств ввода-вывода, работа которых контролируется специаль-

ными устройствами управления. Устройства ввода-вывода обеспечивают передачу информации в оперативную память и обратно. Используя соответствующие команды, программист может задать процедуры чтения или записи информации на эти устройства.

Мы уже отмечали выше, что оперативная память содержит элементы информации. Программисту память представляется состоящей из ряда отдельных ячеек, каждая из которых хранит один элемент информации. Эти ячейки последовательно, начиная с нуля, нумеруются числами натурального ряда, которые принято называть адресами ячеек. Объем памяти определяется тем количеством информации, которое можно хранить в ней одновременно, и, как правило, память больших машин состоит примерно из 256 000 ячеек.

Внутри отдельно взятой ячейки памяти информация задается в виде некоторой комбинации, состоящей из нулей и единиц (двоичных цифр или так называемых битов). Каждый элемент информации, такой, как десятичная цифра, или буква алфавита, или какой-нибудь другой значащий символ, используемый в пишущей машинке, представляется в памяти некоторой фиксированной комбинацией битов, и существует несколько международных стандартов, регламентирующих кодировку символов. Во многих современных машинах память состоит из 8-битовых ячеек, называемых байтами. При этом для кодировки символов байтами можно использовать 256 различных битовых комбинаций, каждая из которых целиком помещается в одной ячейке. Некоторые из этих битовых комбинаций имеют особый смысл для аппаратуры: они представляют собой команды, которые должны быть выполнены. Поэтому если устройство управления, интерпретируя информацию, содержащуюся в ячейке, установит, что это одна из командных комбинаций, то оно инициирует выполнение соответствующей операции.

Перед началом выполнения программа вводится в оперативную память с некоторого устройства ввода в виде строки битовых комбинаций, представляющих команды, после чего управление передается на первую из этих команд. Затем аппаратура осуществляет интерпретацию и выполнение команды, а также переход к следующей команде. По функциональному назначению все команды можно разбить на четыре группы, хотя необходимо отметить, что наборы команд различных машин отличаются друг от друга. Во-первых, это арифметические команды, такие, как сложение, умножение и т. д., назначение которых очевидно. К следующей группе относятся команды символической обработки, или так называемые логические команды, которые могут использоваться для передачи информации из одной части памяти в другую или, например, для установления тождественности двух битовых комбинаций. Вообще команды, принадлежащие к двум первым группам, определяют операцию, которая должна быть выполнена, и адреса тех данных, над которыми эта операция будет выполняться. На практике большинство

современных машин используют для ускорения процесса арифметических вычислений несколько сверхбыстрых ячеек, называемых регистрами. Эти регистры также перенумерованы, т. е. имеют адреса, на которые можно ссылаться в командах.

Команды третьей группы — это команды управления процессом вычислений. Обычно команды выполняются в том порядке, в каком они оказались расположенными в оперативной памяти после ввода программы. Команды управления позволяют изменять этот естественный порядок выполнения программы, обеспечивая переход к команде, расположенной в заданной ячейке памяти. Команды передачи управления могут быть безусловными и условными. Например, за командой сравнения содержимого двух ячеек памяти может следовать команда, вызывающая передачу управления в заданную ячейку, если результат сравнения показывает, что содержимое ячеек тождественно. Используя такие команды, можно организовать, например, поиск в заданной области памяти битовой комбинации, представляющей букву «А», и переход в некоторую точку программы после того, как эта буква будет найдена.

К четвертой группе относятся команды ввода-вывода. Эти команды задают режим работы (чтение или запись) конкретного устройства, а также указывают область памяти, участвующую в процессе обмена информацией.

Таким образом, программа представляет собой длинную последовательность элементарных команд машин, каждая из которых задает выполнение только одной операции. Искусство программирования как раз и заключается в умении представить требуемый процесс вычислений в виде последовательности очень мелких логических шагов. При этом может оказаться, что даже для реализации сравнительно простых вычислений потребуется солидная (по числу использованных команд) программа.

В период эксплуатации первых машин программисты практически писали программы в некотором цифровом коде, известном как машинный код, или язык машины. В настоящее время они могут составлять программы на языке символического кодирования (символический машинный код), который дает возможность определять команды при помощи символов (например, символ «А» может обозначать операцию «сложить», «В» — «передать управление» и т. д.), а также использовать при ссылках на ячейки памяти не истинные адреса, а символические имена. Хотя программист и может писать программу на этом символическом языке, напоминающем шахматную нотацию, машина все же понимает только такие программы, команды которых представлены в формате машинного кода. Поэтому существуют специальные программы, называемые ассемблерами, которые перед выполнением программ, записанных на символическом языке, переводят их на язык машины.

Символический язык, или, как его еще называют, язык ассем-

блера, считается языком низкого уровня, поскольку, как правило, одному предложению этого языка соответствует только одна команда на языке машины. Более сложные языки, особенно те, что используются в научных исследованиях, например ФОРТРАН, относят к языкам высокого уровня. Для этих языков характерно укрупнение операций, т. е. одному предложению на языке высокого уровня соответствует несколько предложений на языке низкого уровня или машинных команд. Программа, необходимая для перевода с языка высокого уровня, конечно, сложнее и больше ассемблера, но по сути дела она реализует аналогичный процесс, а именно составление машинных команд по заданным символическим предложениям. Выигрыш заключается в том, что писать программу на языке высокого уровня проще и, следовательно, быстрее; кроме того, нет никакой необходимости знать что-либо о языке машины. В современных вычислительных системах ассемблеры и компиляторы<sup>1)</sup> входят в комплект стандартного программного обеспечения, предоставляемого пользователю разработчиком системы.

## 2.2. ВНЕШНИЕ НОСИТЕЛИ ИНФОРМАЦИИ

Таким образом, в распоряжении программиста, как правило, имеется некий язык, на котором он может написать программу, а также компилятор, переводящий такие программы на язык машины. Помимо этих программных средств, ему также необходим некоторый внешний носитель информации, с которого программа или используемые ею данные могли бы быть прочитаны в оперативную память. Для этого<sup>2)</sup> современные вычислительные системы оснащаются самыми различными устройствами ввода-вывода, в число которых входят печатающие устройства, оборудование для работы с перфокартами и перфолентами, магнитные ленты, диски и барабаны. В этом разделе мы рассмотрим принципы работы различных внешних носителей информации. Прежде всего отметим, что каждый носитель осуществляет две функции. С одной стороны, носитель обеспечивает возможность ввода в машину программы и данных каждый раз, когда такого рода необходимость возникает в процессе решения задачи. С другой стороны, носитель является запоминающим устройством, в котором можно хранить (сколь угодно долго) информацию, не используемую в текущих вычислениях. Так, например, рассматривавшаяся ранее программа составления платежной ведомости может храниться на магнитной ленте, а ее данные — на перфокартах. Когда же необходимо получать еженедельные сведения, то вся эта информация может быть считана в оперативную память с различных периферийных устройств (так еще

<sup>1)</sup> То есть программы перевода с языков высокого уровня.— *Прим. ред.*

<sup>2)</sup> А также для вывода результатов счета.— *Прим. ред.*

называют устройства ввода-вывода), после чего машина автоматически составит платежную ведомость. Поэтому внешние носители должны рассматриваться с точки зрения того, каким образом они обеспечивают ввод-вывод и хранение информации.

Перфоленды и перфокарты являются наиболее широко распространенными носителями, которые, учитывая их физическую общность, можно рассмотреть вместе. Эти носители имеют несколько достоинств. Начнем с того, что они дешевы, легко заменяемы и просты в обращении. Для подготовки перфоленд и перфокарт используются различные устройства подготовки данных, причем работа на этих устройствах — дело не более сложное, чем печатание на пишущей машинке. Столь простая подготовка данных может быть к тому же выполнена автономно, т. е. без непосредственной связи с вычислительной машиной. Кроме того, перфокарты весьма удобны для хранения такой информации, которая заносится в карточки библиотечного каталога. Наконец, оборудование для работы с перфолендами и перфокартами значительно дешевле других устройств ввода-вывода, а для надежного хранения информации на этих носителях не требуется специальных условий.

Вместе с тем перфокартам и перфолендам присуще несколько недостатков. В силу своей физической природы они недолговечны, и частое их использование приводит к неприятностям. Хотя перфокарты и перфоленды легко хранить, они значительно объемнее по сравнению с другими возможными носителями информации. Вследствие этого они не дают практически никакого выигрыша с точки зрения физического объема памяти по сравнению с хранилищем информации обычной библиотечной системы. Кроме того, перфокарты и перфоленды должны загружаться вручную в читающее устройство каждый раз, когда необходимо осуществить ввод в машину содержащейся в них информации. Более совершенные носители информации после ввода в них материала могут оставаться в состоянии оперативной связи с машиной, т. е. всегда готовы к вводу-выводу информации, не требуя при этом вмешательства человека. С точки зрения работы машины есть еще один вопрос, касающийся времени, затрачиваемого на чтение информации с перфокарт и перфоленд. Дело в том, что арифметические операции и команды передачи управления выполняются машиной чрезвычайно быстро, и по сравнению с этим время, затрачиваемое на выполнение команд ввода-вывода, просто огромно. После того как команда ввода-вывода с некоторого устройства дешифрирована устройством управления, процесс ввода-вывода информации протекает независимо от работы процессора (т. е. арифметического устройства и устройства управления), который может тем временем выполнять последующие команды программы. Однако если при этом в программе возникнет необходимость обращения к передаваемым данным, то выполнение программы должно быть приостановлено до момента завершения

процедуры ввода-вывода. Ясно, что если процесс ввода-вывода протекает медленно, то в этом случае программа может оказаться бездействующей в течение очень длительного интервала времени. И это действительно так, если учесть, что скорость работы устройства чтения с перфокарт измеряется сотнями карт в минуту, а время, затрачиваемое на выполнение обычной машинной операции, составляет миллионные доли секунды. Поэтому если фактор скорости является важным, как это и есть на самом деле в некоторых библиотечных приложениях, рассматриваемых ниже, то необходимо использовать другие, более подходящие внешние носители информации.

Помимо перфокарт и перфолент, в вычислительных системах широко используются различные виды магнитных носителей информации. Носители этого типа делятся на две группы: с одной стороны, магнитные ленты, с другой — так называемые устройства с прямым доступом, или произвольной выборкой информации, такие, как магнитные диски или барабаны. Устройства обеих групп имеют несколько общих преимуществ по сравнению с медленными носителями информации.

Поскольку хранение информации в магнитных носителях основано на магнитоэлектрических принципах, а не на механической перфорации поверхности носителя, то они имеют значительно большую удельную емкость по сравнению с перфокартами и перфолентами. В библиотечном деле, где необходимо хранить большие количества подробных сведений, это преимущество магнитных носителей выглядит особенно привлекательным. В связи с тем что ввод-вывод информации на магнитных носителях является процессом магнитной записи или считывания электрических сигналов, все эти носители работают значительно быстрее, чем устройства ввода-вывода с перфокарт и перфолент, хотя магнитная лента значительно уступает в скорости магнитным барабанам и дискам. К сожалению, эксплуатация магнитных носителей сопряжена с необходимостью поддержания в машинном зале определенных температурных и атмосферных условий. Это обстоятельство, а также цены электронных компонентов магнитных носителей обуславливают высокую стоимость такого рода устройств.

Необходимо отметить, что между магнитной лентой и устройствами с произвольной выборкой информации существуют значительные различия. Здесь имеет смысл кратко обсудить способы организации больших массивов информации, таких, например, какие используются в информационно-поисковых системах. Каждый отдельный массив информации, упорядоченный по определенному признаку, называется файлом, так, что в библиотечном контексте могут встречаться термины «авторский файл», «титовый файл» и, скажем, «главный файл», в котором содержится исчерпывающая информация о всех библиотечных документах (аналогично тому, как это делается на картах каталога). Библиотечная терминология

в данном случае весьма похожа на ту, что используется в вычислительной технике. Внутри файла каждый отдельный элемент информации называется записью. Например, в случае рассмотренных выше файлов записью является соответственно фамилия (имя и отчество) одного автора, одно название или справка об одном документе. Файл размещается на каком-нибудь внешнем носителе, и поэтому программа, обращающаяся к конкретной записи файла, должна предварительно считать этот файл <sup>1)</sup> в оперативную память, а затем определить местоположение требуемой записи в памяти. После этого текст записи может быть, например, изменен и новая версия файла переписана обратно на внешний носитель.

Именно так, как описано выше, и обстоит дело при работе с магнитной лентой. Для обращения к конкретной записи файла, расположенного на ленте, необходимо прочитать все его записи, начиная с первой, до тех пор, пока не встретится требуемая запись. При этом нет никакого механизма, который позволил бы предварительно определить точное местоположение записи на ленте и обратиться к этой записи непосредственно. В приложениях, подобных процедуре составления платежной ведомости, когда естественно начинать просмотр информации с первой записи файла и при этом обрабатываются большинство или все записи файла, использование магнитной ленты вполне приемлемо. Однако в случае информационно-поисковой системы, где, как правило, требуется найти запись, принадлежащую некоторому конкретному документу, магнитная лента оказывается явно медленным носителем, а преимущества устройств с произвольной выборкой информации проявляются особенно очевидно.

Устройства с произвольной выборкой, как подсказывает их название, позволяя программным путем точно определить местоположение требуемой записи на носителе и прочитать только эту запись. Таким образом, в данном случае нет никакой необходимости считывать в оперативную память большое число ненужных записей и затем просматривать их. Поскольку время ввода-вывода информации решающим образом сказывается на быстродействии системы в целом, то устройства с произвольной выборкой в этом плане имеют решающее преимущество по сравнению с другими внешними носителями. В некоторых наиболее сложных библиотечных приложениях, рассматриваемых в настоящей работе, использование таких устройств просто необходимо.

Для примера рассмотрим в общих чертах принципы работы устройства на магнитных дисках. Это устройство можно сравнить с проигрывателем, на вращающемся шпинделе которого зафиксированы с определенным интервалом друг от друга шесть долгоиграющих пластинок. При этом пять верхних и пять нижних торцевых поверх-

<sup>1)</sup> Как правило, файл читается по частям.— *Прим. ред.*

ностей дисков (пластинок) используются для хранения информации, а две поверхности — самая верхняя и самая нижняя — служат защитным целям. На некотором расстоянии от каждой рабочей поверхности установлен ряд читающих и записывающих магнитных головок. Рабочая поверхность диска разбивается на треки (дорожки), на которых и располагается информация, причем магнитные головки подвижны и могут быть установлены над любым треком рабочей поверхности. Процесс обращения к некоторой записи файла, расположенного на диске, состоит из этапов вычисления номера соответствующей рабочей поверхности, установки магнитных головок над треком, содержащим требуемую запись, ожидания момента появления под головками начала этой записи и, наконец, собственно обращения к записи. Этот процесс протекает чрезвычайно быстро, по сравнению с работой внешних носителей, рассмотренных ранее, хотя его труднее запрограммировать.

Заканчивая рассмотрение внешних носителей, можно сделать вывод, что для хранения больших файлов, обращение к которым происходит в произвольном порядке, но должно выполняться быстро, необходимо использовать магнитные носители с прямым доступом к информации. Кстати говоря, многие библиотечные информационные файлы относятся именно к этой категории.

### 2.3. НЕКОТОРЫЕ НАПРАВЛЕНИЯ РАЗВИТИЯ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

Итак, мы кратко обсудили, как составляется программа, а также как она выбирает и считывает с файлов, хранящихся на внешних запоминающих устройствах, необходимые ей данные в оперативную память машины. Некоторые преимущества машинизированной системы легко предвидеть. Машина может обрабатывать огромное количество информации с невероятной по обычным человеческим понятиям скоростью. При этом машина не знает усталости, она может многократно повторять различные операции без какого-либо снижения производительности и не делая ошибок. Но она требует, чтобы получаемая ею информация, а также программа, которую она выполняет, были совершенно правильными. Другими словами, производительность машинизированной системы определяется структурой данных и качеством используемой программы. Если говорить применительно к библиотечной ситуации, то это, например, означает, что работа, которая при обычной справочной системе заключается в составлении списков книг или заполнении карт каталога, должна быть в случае машинизированной системы точно выражена в виде неких этапов подготовки и контроля файлов, используемых впоследствии машиной. Причем если в первом случае незначительную ошибку можно игнорировать, то во втором она может явиться причиной катастрофических последствий для системы. Таким

образом, машина требует исключительной точности в формулировке процесса обработки информации, и различные методы достижения этого программным путем являются одной из важных тем, обсуждаемых в настоящей книге.

Мы можем закончить эту вводную главу кратким рассмотрением тех характерных особенностей современных вычислительных систем, которые могут иметь непосредственное отношение к библиотечному делу. Выше уже отмечалось, что компилятор является частью стандартного программного обеспечения машины. Это программное обеспечение представляет собой комплект программ, поставляемых разработчиком вычислительной системы для того, чтобы облегчить пользователю работу на машине. Выше рассматривалась очень простая структура программы. В действительности же с каждой машиной поставляется операционная система, которая предоставляет пользователю возможность писать программы на различных языках, а также проверять их и проводить по ним вычисления. Основой программного обеспечения является супервизор — программа, которая постоянно находится в оперативной памяти и осуществляет общий надзор за прохождением задач. Прежде чем какая-либо программа пользователя начнет выполняться, супервизор тщательно проверяет, все ли устройства, необходимые для нормальной работы программы пользователя, исправны и предоставлены ей, а также не собирается ли пользователь обращаться к тем файлам, доступ к которым ему не разрешен. Кроме этого, супервизор выполняет еще множество других проверок с целью обеспечения сохранности системы и защиты ее от любых ошибок в программе пользователя или в работе аппаратуры.

Супервизор использует ряд сложных по своей структуре подпрограмм, которые обеспечивают выполнение мониторинговых функций, т. е. постоянный контроль за операциями программы пользователя. Дело в том, что программа пользователя не может, например, сама выдать какую-либо команду обращения к устройству ввода-вывода. С этой целью она должна обратиться к супервизору, передав ему также сведения о необходимой ей информации и о том, где эта информация должна быть размещена в оперативной памяти машины. Получив управление, супервизор выдаст ряд команд, обеспечивающих ввод-вывод необходимой информации, проконтролирует правильность выполнения этих команд, а затем возвратит управление программе пользователя. Такова в общем виде картина выполнения сложной мониторинговой операции в предположении, что все устройства работают нормально.

Вероятно, наиболее значительным успехом в вычислительной технике в настоящее время является широкое использование мультипрограммных машин. Мы уже отмечали, какое значение имеет программа-супервизор, а также и тот факт, что время, затрачиваемое на выполнение операции ввода-вывода, слишком велико, по

сравнению с быстродействием оперативной памяти и арифметического устройства. Как было описано ранее, операция ввода-вывода, после того как она инициирована соответствующей командой, может выполняться независимо от работы процессора. Однако если программа, потребовавшая ввод-вывод информации, не может делать что-либо до тех пор, пока данные не поступят в оперативную память и не будет получен электрический сигнал (так называемое «прерывание»), свидетельствующий об окончании передачи данных, то она должна ждать этих событий и при этом ценное процессорное время тратится впустую.

Мультипрограммные средства как раз и предназначены для того, чтобы обеспечить эффективное использование этого времени и тем самым устранить простои процессора. Они дают возможность супервизору после того, как он инициировал выполнение операции ввода-вывода для некоторой программы пользователя, запасти сведения о текущем состоянии этой программы, а затем передать управление другой программе пользователя. Когда придет «прерывание», сигнализирующее об окончании процесса передачи данных, будут запасены сведения о состоянии этой другой программы пользователя, а управление возвращено снова первой программе. Таким образом, возможности процессора машины могут быть использованы полностью, и со стороны дело будет представляться так, что одновременно выполняются две или более программы, хотя в действительности каждая программа выполняется в течение некоторого короткого периода времени, затем останавливается и продолжается несколько позже. Это обстоятельство является еще одним доводом, подтверждающим целесообразность передачи супервизору функции управления вводом-выводом информации.

В большинстве вычислительных систем могут вместе выполняться несколько программ, и организация прохождения потока программ является функцией программного обеспечения. Для программного обеспечения программа пользователя представляется как некая «работа», которая должна быть выполнена, причем эта работа имеет больший или меньший по сравнению с другими работами приоритет и для ее выполнения требуется какой-то объем оперативной памяти и какое-то количество периферийных устройств. Каждый пользователь должен составить описание своих работ на некотором специальном языке. В этих описаниях сообщаются сведения о требуемых ресурсах и приоритете. Согласно описаниям работ, подпрограммы планирования, входящие в комплект программного обеспечения, выполняют ряд операций, позволяющих с учетом установленных пользователями приоритетов наилучшим образом использовать машину, доступные устройства ввода-вывода и уменьшить потери времени.

Так, например, пользователь автоматизированной библиотечной системы может пожелать получить программы распечатки списка

всех новых поступлений и составления карточек библиографического каталога для них. Программное обеспечение предоставляет при этом в распоряжение пользователя подпрограммы, позволяющие ему составлять на различных языках необходимые программы, переводить эти программы на язык машины и проверять их так, чтобы впоследствии выполнять, не опасаясь, что важная информация может оказаться испорченной вследствие программной ошибки. Другие же подпрограммы дают возможность наилучшим образом использовать потенциал машины при выполнении рабочих вычислений по полностью проверенным программам пользователей. Такие системы программного обеспечения такого типа как раз и предполагается при общем обсуждении вопросов организации и разработки библиотечных систем, проводимом в конце данной книги.

Следующий этап развития вычислительной техники характеризуется внедрением систем коллективного доступа, работающих в режиме реального времени. Эти термины требуют некоторых пояснений. Как уже отмечалось, программное обеспечение обычной мультипрограммной системы включает подпрограммы планирования работ. Однако для того, чтобы все переданные системе работы были выполнены в некотором определенном порядке, необходимо их описание собрать в соответствующий пакет, который будет вводиться в оперативную память в качестве входных данных для подпрограмм планирования работ. В отличие от этого система коллективного доступа, работающая в режиме реального времени, предоставляет в постоянное распоряжение пользователя несколько различных программ, к которым он может обращаться, печатая, например, запрос на выносной пишущей машинке или каком-нибудь другом терминальном устройстве. Программное обеспечение осуществляет интерпретацию запроса, или, как еще говорят, «приказа», и связывает пользователя, находящегося за терминалом, с необходимой ему программой. Один из наиболее часто употребляемых способов использования таких систем коллективного доступа заключается в составлении программы непосредственно за терминалом. При этом программа передается в машину команда за командой, а не составляется вся заранее и не вводится с перфокарт или перфоленты.

Другой важный способ использования системы заключается в установлении непосредственной связи с большими файлами. Например, пользователь системы коллективного доступа может подойти к терминалу и, напечатав соответствующий приказ, получить доступ к заданной группе файлов. Такого рода операции широко используются в системах резервирования мест на авиалиниях; очевидно, что они могут очень пригодиться и для быстрого получения информации о книгах, справочных сведений из информационных файлов и т. п.

Большая система коллективного доступа с большим числом подключенных к ней терминалов является в настоящее время все-таки дорогим удовольствием. Однако система, обеспечивающая организацию легко доступных файлов на устройствах с произвольной выборкой информации и предоставляющая некоторые возможности для работы в режиме реального времени, оказывается вполне приемлемой для практического использования. Поэтому в данной работе рассматриваются вопросы, касающиеся системы именно такого типа.

## ЧТО НУЖНО БИБЛИОТЕЧНОЙ СИСТЕМЕ

---

### 3.1. ПОДХОД К ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЕ

В предыдущих главах была обоснована необходимость использования в библиотечном деле таких средств механизации работ, как вычислительные машины, а также были кратко рассмотрены те характерные черты вычислительной машины, которые делают ее особенно привлекательным средством для указанной цели. В данной главе обсуждаются вопросы, непосредственно касающиеся такого применения машин.

Любая организация, намеревающаяся прибегнуть к услугам вычислительной машины, должна прежде всего исследовать все обстоятельства производственного процесса, при которых будет использоваться машина. Результатом таких исследований, называемых «системным анализом» (под этим термином понимается весьма широкий диапазон мероприятий, включающий как составление общих планов системы, так и детальную спецификацию используемых вместе с ней машинных программ и документов), должно явиться описание и формализация принципов функционирования системы, в которой преимущества машины будут наилучшим образом использованы для решения задач, поставленных перед этой системой. В частности, можно выделить четыре аспекта этого системного исследования, два из которых рассматриваются в конце этой главы. Во-первых, это определение целевого назначения системы, т. е. тех требований, которым должна удовлетворять всякая хорошая библиотечная система, и, во-вторых, четкое выделение тех специфических задач, для решения которых предназначается система. При обсуждении того, в какой степени вычислительная машина может способствовать успешному выполнению предъявляемых к системе требований и решению указанных выше задач, целесообразно выделить те приложения, в которых использование машины по крайней мере естественно и приемлемо, поскольку именно перечень этих приложений оказывается наилучшим тестом для выявления преимуществ или, напротив, недостатков вычислительной системы.

Два других аспекта исследования системы, которые рассматриваются подробно в конце этой главы, касаются организации взаимодействия между системой и пользователями, а также определения нормативов, позволяющих оценить преимущества или, напротив,

недостатки системы. Несколько позже будет показано, что эти вопросы отнюдь не независимы и что в действительности достоинство системы во многом определяется тем, насколько легко можно воспользоваться предоставляемыми ею услугами.

Прежде чем перейти к рассмотрению требований, предъявляемых библиотечной системой, необходимо сделать несколько общих замечаний.

Вообще говоря, чем больший объем работы может быть возложен на машину, тем более эффективным и экономичным средством механизации она становится. По этой причине в настоящей работе проводятся всесторонние системные рассмотрения библиотечной службы. Хотя в этих рассмотрениях будут обсуждаться независимые с точки зрения механизации процедуры библиотечного процесса, тем не менее следует учитывать, что библиотеки (как и другие возможные области машинных приложений) пострадали от частичного использования средств, первоначально предназначавшихся для устранения отдельных трудоемких операций; только впоследствии было понято, что эти средства могут служить основой для совершенно нового подхода к проектированию библиотечной системы. Если имеются финансовые возможности, то набор механизированных операций можно расширять, однако необходимо признать, что во многих случаях дотация, выделяемая на информационное обслуживание, может оказаться недостаточной, чтобы гарантировать достижение тех наилучших результатов, которые дает всесторонний системный подход. В настоящей работе не проводится никаких рассматриваний, касающихся весьма трудоемкой задачи переделки существующих библиотечных процедур к виду, удобному для машинной обработки, а также технических проблем преобразования файлов, к рассмотрению которых сводится большинство опубликованных работ по применению машин в библиотечном деле. Предлагаемый в книге системный подход целиком базируется на использовании вычислительной машины и имеет своей целью определить, насколько вся библиотечная деятельность может быть автоматизирована с помощью машины.

Другим фактором, требующим внимания, является совокупность обстоятельств, характерных для деятельности данной библиотеки. Любая библиотека, намеревающаяся использовать вычислительную машину, должна провести у себя тщательное системное обследование, направленное на выявление специфики своей работы. Например, специфический стоимостный анализ, проводимый Библиотекой Конгресса [5], является типичным исследованием такого рода. В книге, подобной этой, формулировки системных требований должны носить общий характер, но при этом необходимо сделать предостерегающее замечание о том, что одна и та же вычислительная система не может успешно обслуживать библиотеки различных типов. Специалисты в области информационного поиска часто попа-

дают впросак, предполагая, что метод, пригодный в одних обстоятельствах, столь же успешно будет работать в любых других.

Мы можем, например, выделить следующие три разновидности библиотечной деятельности. Информационный отдел коммерческого или промышленного предприятия будет ориентирован на хранение информации по ограниченному количеству тем, а также на ожидание специфических, часто трудных для понимания вопросов, поступающих от, вообще говоря, хорошо информированных пользователей. С другой стороны, библиотека университетского типа будет ожидать более разнообразные вопросы, относящиеся к обширному кругу тем, причем в большинстве своем эти вопросы поступают от лиц, которые имеют только некоторое представление о том, что их интересует. Наконец, третий вид библиотечной деятельности характерен для публичной библиотеки, которая должна обслуживать всех пользователей, включая даже тех, кто не знает, чего они хотят, и тех, кто не хочет ничего конкретного. Важно отметить, что в случае первого из трех рассмотренных видов библиотечной деятельности значительная часть работы по автоматизации библиотеки уже выполнена. При дальнейшем обсуждении следует иметь в виду, что различные условия библиотечного обслуживания требуют различных уровней механизации. Это обстоятельство является особенно важным при рассмотрении вопросов взаимодействия системы с библиотечными пользователями. С учетом этих замечаний мы можем перейти к рассмотрению целей библиотечной системы.

### 3.2. ЦЕЛЕВОЕ НАЗНАЧЕНИЕ СИСТЕМЫ

Библиотека, или информационное обслуживание, как следует из последнего названия, обеспечивает некий сервис для строго определенного или произвольного круга пользователей, причем если какие-то виды услуг, в которых «клиент» заинтересован, не выполняются, то коэффициент полезности обслуживания резко снижается. Поэтому ориентация на пользователя и проявление особого внимания к его нуждам должны иметь весьма высокий приоритет среди прочих факторов, связанных с функционированием библиотечной системы. В сущности, одно из главных достоинств автоматизированной библиотечной системы заключается в том, что она позволяет сравнительно просто осуществлять ориентацию обслуживания на пользователя, а также привносит некий динамический элемент, который дает возможность пользователю самому принять участие в приспособлении системы к своим собственным нуждам.

Кинг [5] так резюмирует преимущества, связанные с использованием машины:

Сосредоточение внимания библиотечного сервиса на индивидуальном пользователе с целью удовлетворения его исследовательских потребностей.

Он же утверждает, что:

В идеальном случае автоматизированная система должна предоставлять в непосредственное распоряжение пользователя все библиотечные ресурсы.

Итак, эта первостепенная задача системы может быть решена посредством использования машины и в значительной степени определяет все другие цели системы. Другими словами, целевое назначение системы определяется пожеланиями пользователя.

В данном контексте под пользователем понимается пользователь системы, но можно рассматривать две различные группы пользователей: с одной стороны, это библиотечный персонал, который управляет системой и регистрирует поступление новой информации, с другой — обычные библиотечные пользователи, которые требуют системного обслуживания. Обе эти группы могут предъявлять к системе различные требования, и желательно, чтобы система была способна выполнять различные задачи, однако в общих чертах мнения обеих групп относительно того, удачна автоматизированная библиотечная система или нет, будут определяться одинаковыми факторами.

Более того, поскольку вычислительная система мыслится как некое усовершенствование обычной библиотечной системы, то от нее вполне естественно ожидать более высоких показателей производительности.

Поэтому либо машина должна обеспечить повышенную производительность системы, либо нужно доказать, что использование машины дает возможность выполнять различные операции, которые невозможны в обычной библиотечной системе. Автор книги ставит перед собой цель показать, что дело обстоит именно так и что, хотя на этом пути пока и трудно ожидать превышения показателей многих из существующих средств библиотечного обслуживания, тем не менее механизация системы обеспечивает возможность получения широкого набора услуг, а также придает системе логичность и точность, столь желательные в условиях, когда хранится и обрабатывается большая масса информации.

Можно указать несколько положительных особенностей, свойственных автоматизированной библиотечной системе:

1. Устранение значительной части примитивной или чисто канцелярской работы.

Использование машины предполагает уменьшение, а не увеличение объема различных сортировочных работ. Действительно, машина обладает высоким быстродействием и идеально приспособлена для выполнения повторяющихся операций, входящих, например, в такие стандартные процедуры, как составление списков или карт файлов для каталогов.

2. Устранение ошибок, попадающих в систему.

Машина не устает и, как правило, не делает ошибок. Обнару-

жение ошибок является решающим моментом в работе системы; так, например, если книга будет ошибочно отнесена в другую рубрику, то впоследствии будет очень трудно найти информацию, которая в ней содержится.

3. Постоянство информации внутри системы.

При обработке большой массы информации ручным способом обязательно сказывается различное понимание вопросов исполнителями. Это приводит к тому, что один и тот же документ может быть по-разному классифицирован различными людьми. Машина же всегда пунктуальна и помнит все подробности, касающиеся данного вопроса.

4. Быстрота ответов на вопросы.

Машина может значительно быстрее, чем человек, просматривать файлы и выдавать новейшую информацию о последних поступлениях в библиотеку.

Итак, при обсуждении различных машинно-ориентированных средств мы можем в дальнейшем руководствоваться следующими критериями. Действительно ли машина облегчает работу библиотекаря или она просто несколько меняет характер этой работы? Может ли машина успешно справиться с ошибками, содержащимися в получаемой ею информации? Будет ли способна машина устранить противоречия, возникающие в системе? И наконец, сможет ли машина обеспечить удобный сервис при достаточно быстрой реакции системы на поступающие ей запросы? Будем надеяться, что на все эти вопросы можно дать благоприятный ответ.

### 3.3. ТРЕБОВАНИЯ БИБЛИОТЕЧНОГО ПЕРСОНАЛА

Теперь перейдем к детальному обсуждению отдельных задач, связанных с работой библиотечной системы, и вопросов приспособления машины для решения этих задач. Начнем с рассмотрения обязанностей библиотечного персонала.

Следует отметить, что первоначально возможности машин были осознаны и стали практически использоваться именно в тех сферах библиотечной деятельности, которые связаны с повседневными рутинными библиотечными операциями. Библиотека должна вести учет всех приходящих в нее новых документов, регистрировать их поступление, составлять различные указатели и аннотировать непоступившие материалы, например периодические издания. Часто с помощью вычислительных систем пытаются каким-то образом механизировать именно рутинную процедуру обработки постоянно растущего потока периодических изданий, которые нередко трудно классифицировать и индексировать средствами обычной библиотечной системы.

Машина позволяет ускорить процесс регистрации библиотечных поступлений, а также сравнительно просто обеспечивает использование различных списковых удобств. Действительно, программа, осуществляющая сортировку массива информации различными способами и составляющая затем соответствующие списки, является довольно-таки простым средством.

Имеющиеся успехи в этом направлении, связанные с ускорением процесса обслуживания и упрощением обязанностей персонала, выглядят весьма скромно, однако при творческом подходе к рассматриваемой проблеме может быть достигнут значительный выигрыш.

Распространенным приемом является использование так называемого индекса КСК (ключевого слова в контексте), с помощью которого машина составляет список названий, содержащих данное ключевое слово. В случае технических статей этот прием оказывает весьма ценную помощь при поиске информации, особенно если одно и то же название могло оказаться проиндексированным по различным ключевым словам, т. е. отнесенным в различные части индексного указателя.

Типичный пакет программ, предназначенный для составления списков, информирующих о периодических изданиях, может обеспечивать следующее:

- а) поддержание списка полных заголовков для серийных изданий,
- б) список текущих поступлений,
- в) различные указатели названий и статей,
- г) списки необходимых периодических изданий,
- д) списки неполученных изданий,
- е) сроки окончания подписных листов.

Несколько обстоятельств определяют привлекательность программного пакета, предоставляющего описанные выше возможности.

В первую очередь следует иметь в виду, что при условии разумного регистрирования сведений о поступающих документах все перечисленные выше списки или любые желаемые их варианты могут быть получены за однократное прохождение пакета программ на машине в конце каждого календарного периода времени, допустим дня или недели. После того как документы зарегистрированы, определенная часть библиотечного персонала освобождается от дальнейшей работы, связанной с составлением различных указателей и списков. Кроме экономии времени и сокращения объема ручных работ, есть и другие сопутствующие преимущества. С помощью программ из пакета может быть получена и размножена для непосредственной демонстрации на стендах или для распределения среди пользователей более полная текущая информация. Далее, можно организовать автоматическое оповещение библиотечного персонала, например, о том, что какие-то периодические издания не поступили

вовремя или не выписаны, а также о невозвращенных в срок документах; при этом машина будет сама следить за соответствующими календарными сроками. Уведомления такого рода, оформленные наподобие счетов за электроэнергию, могут выпускаться без участия библиотечного персонала.

К сожалению, использование машины связано также и с некоторыми неудобствами. Одно из них, которое не будет рассматриваться далее в данной работе, касается проблемы вывода информации. Вопросам получения печатных документов на машине уделяется большое внимание со стороны специалистов по вычислительной технике. Существенное значение они имеют и для сотрудников библиотек, поскольку форматы списков или выдаваемых машинной документов важны всегда, а особенно если эти документы предназначены для широкой аудитории.

С задачами, сформулированными в предыдущем разделе, тесно связана проблема ввода информации. На этом этапе не следует уделять слишком много внимания предупреждению ошибок, поскольку проверка вводимых перфокарт может быть осуществлена самой машиной, а последующий контроль и внесение исправлений в составленные машиной списки выполняются таким же образом, как если бы эти списки были получены обычными средствами. Любая неправильная информация после соответствующих исправлений просто вводится снова. Однако объем работы, связанной с подготовкой необходимой информации в удобной для машинной обработки форме, весьма значителен. Поскольку с точки зрения машины вся информация выглядит просто как строка символов, некоторые специальные символы, такие, как \* или &, должны быть зарезервированы в качестве ограничителей, используемых при вводе информации для отделения, например, поля названия документа от поля автора. Часто при подготовке информации применяют числовое кодирование, что связано с дополнительной работой и необходимостью запоминания каких-то новых правил. В таких случаях вероятность внесения ошибок значительно возрастает, а обнаружение их становится более трудным. Такого рода предварительное редактирование может оказаться сущим ужасом. Проект автоматизированной библиотеки, разработанный Калифорнийским университетом (Сан-Диего) [10], включает 13 специальных правил для служащих о том, как печатать ограничители, используя 10 специальных символов клавиатуры устройства подготовки данных. По-видимому, такого рода система взваливает на библиотечный персонал невыносимое бремя.

Правда, информация для формирования списков должна быть подготовлена только один раз, и если такую же информацию можно использовать для составления карт каталога или файлов с целью их пополнения, то дополнительная работа может иметь смысл. Однако следует отметить, что аналогичного результата можно более

успешно достигнуть, используя машину. Различные способы такого подхода будут рассмотрены несколько позже.

В связи с каталогом возникает следующий вопрос. Вместо того чтобы использовать машину для составления карт файлов, что является всего лишь механизацией некоторых операций существующей системы, можно, сохранив за автоматизированной системой обязанности по формированию необходимых списков и печатных документов, организовать в ней также каталог, представленный в виде машинно-ориентированного файла. При этом все материалы о поступлениях вводятся в вычислительную систему и хранятся на одном из внешних запоминающих устройств, рассмотренных в гл. 2. Решение использовать машину для хранения информации, а не только для механизации канцелярских процедур является первым из серии логических шагов, направленных на передачу машине все большего числа традиционных функций библиотечного персонала.

Такой логический подход к вычислительной системе будет подробно рассматриваться в гл. 7. Здесь же достаточно сказать, что обнаружение ошибок в вводимой информации теперь становится чрезвычайно важной задачей, поскольку в данном случае дело обстоит не так просто, как при вводе информации с перфокарт, когда любую испорченную карту можно было переписать и отперфорировать заново. Кроме того, сам факт хранения информации в машине, естественно, предполагает наличие эффективного и легкого способа доступа к этой информации. Такого рода задача информационного поиска представляет собой эффективный тест для проверки приспособляемости вычислительной машины к библиотечному делу.

Наконец, вероятно, следует упомянуть о совершенно другом возможном использовании вычислительных машин. Если несколько библиотек используют вычислительные системы, то связь между различными такими библиотеками может быть достигнута при помощи такого средства, как сеть вычислительных машин. Этот вид общественных услуг уже сейчас используется при проведении вычислительных работ и позволяет добиться сокращения времени выполнения заказа, что очень важно с точки зрения эффективного обслуживания библиотечных пользователей. При использовании сети библиотеки могут не только общаться друг с другом, обмениваясь новостями и планами, но и местонахождение любого неизвестного или отсутствующего в данной библиотеке документа, требуемого пользователем, может быть немедленно установлено путем автоматического запроса, а затем соответствующий заказ можно передать через машинную связь. Это означает, что неприятная пауза, пока идет поиск требуемого документа и выполнение связанного с ним заказа, может быть сокращена до минимума, и поэтому нет необходимости хранить в каждой библиотеке огромное количе-

ство редко используемого материала только для того, чтобы удовлетворять практически любой запрос пользователя.

#### 3.4. ТРЕБОВАНИЯ БИБЛИОТЕЧНЫХ ПОЛЬЗОВАТЕЛЕЙ

Рассмотрев несколько типичных задач, которые система, очевидно, может выполнить под наблюдением персонала, вернемся к библиотечному пользователю. Как уже отмечалось выше, прежде всего в данном случае необходимо учитывать различия, существующие между разными типами библиотек и классами пользователей. Профессор университета, конечно, рассчитывает на другой вид обслуживания, нежели старая миссис Джонс. Тем не менее некоторые обобщения все же целесообразны.

Оставляя в стороне решение вопроса о выдаче библиотечных книг на дом, можно обсудить следующие три независимых требования к системе (или виды обязательных услуг), связанных с тремя различными типами библиотечных пользователей:

##### 1. Беспорядочный перебор.

Пользователь не имеет в виду какую-то определенную информацию или документ, а просто хочет, как говорят, оглядеться вокруг в поисках чего-нибудь интересного.

##### 2. Информационный поиск.

Пользователь периодически или случайно нуждается в определенной информации или документации того или иного рода.

##### 3. Регулярное оповещение.

Пользователь рассчитывает, что он будет информирован о всех новых публикациях, касающихся его конкретной области деятельности.

Каждая из этих услуг должна быть обсуждена с точки зрения механизации библиотечных процедур.

Из рассмотренных трех услуг беспорядочный перебор менее всего пригоден для машинной обработки. Действительно, несколько искусственной представляется такая процедура общения с системой, когда книги почти наугад выбираются из воспроизводимых машиной списков, после чего случайным образом просматриваются с помощью машинных средств отображения те или иные страницы. Самое приятное при беспорядочном переборе — это возможность, натолкнувшись случайно на что-нибудь интересное, получить на руки соответствующую книгу. Нет основания позволить автоматизированной системе вмешиваться в такую процедуру, если это не оговорено специальным указанием не подпускать публику к книгам и статьям, а при поиске нужных материалов использовать только устройства считывания микрофильмов и вычислительную машину. В последнем случае только после того, как обнаружено, что материал слишком велик или слишком серьезен, чтобы его можно было

усвоить не сходя с места, приводится в движение механическое устройство, доставляющее материал из подземных хранилищ библиотеки. По-видимому, такая несколько «мрачная» система может быть внедрена только в условиях технического информационного центра или в другом аналогичном месте, где требования беспорядочного перебора сводятся к минимуму. Ведь нет основания считать, что в обычной библиотечной ситуации доступ к книгам может оказаться затруднительным. Особенность автоматизированной системы поиска и обработки файлов заключается в том, что она обеспечивает хранение информации о документах, а не самих документов.

В связи с этим возникает вопрос об организации информационного поиска. Пользователи требуют, чтобы необходимая им информация была легкодоступной, а также чтобы библиотека предоставляла средства, упрощающие поиск этой информации. Выполнение этих требований связано с двумя характеристиками библиотечной системы. Во-первых, должен существовать некоторый порядок классификации и расположения материалов в системе. В действительности это является служебной обязанностью библиотечного персонала, о которой мы ничего не говорили выше только потому, что она очень тесно связана с другой рассматриваемой характеристикой, а именно способом сопоставления информации, запрашиваемой пользователями, с информацией, имеющейся в библиотеке. Из рассмотрения этих характеристик следует вопрос: каким образом должна быть организована классификация поступающих в систему материалов, чтобы гарантировать пользователям возможность нахождения этих материалов в тех случаях, когда они необходимы? Этот вопрос отражает основную трудность решения проблемы информационного поиска; различные методы ответа на него, основанные на использовании вычислительной машины, обсуждаются в гл. 7.

Третий вид системных услуг обеспечивает предоставление заинтересованному пользователю полной информации о всех материалах, относящихся к его дисциплине. Как уже отмечалось выше, полные текущие отчеты о всех библиотечных материалах могут быть легко получены в автоматизированной системе, и если удастся найти некоторый способ сопоставления разделов этих отчетов со списками, характеризующими интересы пользователей, то можно добиться этого весьма привлекательного вида обслуживания без особых издержек. Система такого типа называется избирательным распространителем информации (ИРИ). Вопросы, касающиеся реализации такой системы, будут рассматриваться ниже. Весьма интересным обстоятельством в данном случае является то, что любой метод или аппаратура, с помощью которых машина подбирает информацию, соответствующую заказам пользователей, могут быть также использованы в ИРИ с целью снабжения пользователей этой информацией на основе некоторых постоянных критериев. Так, например, любая машина, которая может найти для данного поль-

зователя определенный документ по ядерной физике, может также «запомнить», что этот пользователь интересовался такой рубрикой, и информировать его о новых поступлениях.

В заключение сделаем краткий обзор преимуществ автоматизированной системы. Многие из возложенных на систему задач могут быть выполнены заодно как некий побочный продукт общей процедуры доступа, которая должна быть реализована в любом случае независимо от специфики системы. Во-вторых, обеспечивается определенный контроль над работой системы, причем человеку легко формировать и передавать системе сообщения о необходимых действиях. Применение машины обеспечивает логичность и единообразие в организации системы, а также практически исключает возможность потери информации после того, как она поступила в систему. Платой за это является то, что к библиотечному персоналу предъявляются повышенные требования на этапе ввода информации в систему; и необходимо обеспечить удобный метод поиска и средство сопряжения, позволяющее устанавливать связь с машиной, которая теперь является единственным источником библиографической информации как для библиотечного персонала, так и для пользователей.

## 4

### АППАРАТНОЕ ОБОРУДОВАНИЕ ВЫЧИСЛИТЕЛЬНОЙ МАШИНЫ

---

#### 4.1. ОБЩИЕ СООБРАЖЕНИЯ

В предыдущих главах были определены требования, предъявляемые к автоматизированной библиотечной системе, и показано, что для разработки такой системы необходим довольно-таки сложный вычислительный комплекс, включающий вычислительную аппаратуру, стандартное программное обеспечение и пакеты прикладных программ. В последующих главах более подробно описываются запросы библиотечной системы, предъявляемые к вычислительной машине, и обсуждается, насколько современная вычислительная техника способна удовлетворить эти требования.

Основное рассмотрение касается материального оборудования или аппаратуры, необходимой для поддержания больших файлов информации, обработки этих файлов и организации удобного доступа к этой информации для библиотечного пользователя. Мы уже отмечали, что, помимо объема файлов, который при выборе аппаратуры для библиотечной системы является одним из главных критериев, требуется также учитывать необходимость достаточно быстрого доступа к любой конкретной записи в этих файлах с целью ее модернизации в случае обращения к системе библиотечного персонала или же с целью чтения этой записи в случае обращения к системе библиотечного пользователя, запрашивающего какую-то информацию. Как указывалось в гл. 2, в такой ситуации особенно удобным оказывается оборудование, обеспечивающее оперативную связь носителей информации с машиной. Используя инженерную терминологию, основные требования можно сформулировать так: аппаратура должна обеспечивать возможность хранения больших файлов и простой доступ к ним; режим случайного доступа к информации является обязательным; если это возможно, то система, работающая в режиме реального времени, должна разрешать проводить модернизацию записей файлов и их чтение пользователям, находящимся на обычных рабочих местах.

#### 4.2. ТЕРМИНАЛЫ И КЛАВИАТУРЫ

Если мы начнем изложение, так сказать, с пользовательского конца системы, то прежде всего должны отметить некоторое оборудование, предназначенное для передачи в машину заказов на доступ

к файлам и информации, а также обеспечивающее возможность получения системой необходимых уточнений. Ранее отмечалось, что обычные вычислительные системы получают большую часть такого рода входных данных с перфокарт или перфолент. Однако такой способ ввода информации неприемлем для рассматриваемого здесь режима работы, поскольку нам необходимы средства непосредственной связи с машиной, которая может использоваться человеком, совсем или почти не знакомым с системой.

Обычно для организации работ, выполняемых в режиме разделения времени, применяют какие-нибудь терминалы, непосредственно подключенные к центральному процессору машины. В настоящее время промышленность выпускает много разных терминалов, чтобы обеспечить комплектацию машин всех типов, тем не менее основные характерные черты этих устройств сходны. Во-первых, они имеют клавиатуру, подобную клавиатуре пишущей машинки, с которой информация может быть введена в машину и которая может быть также использована машиной для передачи ответных сообщений непосредственно пользователю, находящемуся за терминалом. Клавиатура терминала, как правило, кроме клавиш обычных символов пишущей машинки, содержит несколько специальных клавиш, нажатие которых инициирует определенные процессы. Так, например, есть клавиша, сигнализирующая о начале работы за терминалом или о подключении пользователя к машине, другая клавиша предназначена для подачи сигнала о вычеркивании только что напечатанной строки информации и т. д.

В зависимости от типа информации, запрашиваемой пользователем, машина выдаст ответ либо на выносной пульт с пишущей машинкой, либо на какое-нибудь другое устройство отображения информации. Так, если пользователь запрашивает список всех книг по данной дисциплине, то он не должен рассчитывать, что машина отпечатает этот список на его выносном пульте <sup>1)</sup>. В данном случае информация будет выведена на стандартное печатающее устройство, выполняющее печать материала построчно. С другой стороны, если пользователь хочет узнать названия и каталожные номера всех книг интересующего его автора, то такого рода сведения могут быть легко воспроизведены на его терминале. Стандартное программное обеспечение должно предоставлять пользователю средства, позволяющие ему указывать, где он предпочитает получить нужный ему результат.

Более совершенные (не приходится и говорить, что более дорогие) терминалы, помимо клавиатуры, снабжаются также электронно-лучевыми трубками или дисплеями (устройствами с дискретным способом построения изображения на экране). При использовании та-

---

<sup>1)</sup> Это потребовало бы значительных излишних затрат машинного времени.— *Прим. ред.*

ких терминалов выводимую информацию вместо того, чтобы печатать, можно воспроизводить на экране. Отметить какую-нибудь точку и произвести запись на экране можно с помощью специального устройства, называемого световым пером. Преимущество в данном случае состоит в уменьшении времени ответа, а также в отсутствии движущихся частей в оборудовании, которые склонны к ошибкам. Компоненты стандартного программного обеспечения и программы, необходимые для обслуживания такого рода терминалов, будут обсуждаться в следующей главе.

С помощью терминала каждый сотрудник библиотеки или библиотечный пользователь может выполнять свою конкретную задачу, хотя в это же время машина обслуживает ряд других пользователей. Поскольку организация коллективного доступа обеспечивается разными терминалами, а также благодаря мультипрограммным возможностям аппаратуры и программного обеспечения машины, в результате у каждого пользователя создается впечатление будто все время, пока он находится за терминалом, машина обслуживает только его одного.

Пользователь может легко установить с машиной связь, где бы он ни находился, если там установлены терминалы. Вопросы, касающиеся внешнего оформления терминалов, относятся к проблеме научной организации труда, и здесь полезно будет отметить, что весьма неприятно, когда приходится, для того чтобы сделать отметку на экране, перегибаться через пульт терминала, ударяясь коленями о стол, или целый день стоять на ногах перед электронно-лучевой трубкой. Однако, если еще на этапе конструирования терминала тщательно продумано все, чтобы сделать устройство удобным, комфортабельным, такого рода вопросы могут быть успешно решены.

Не менее важен вопрос о размещении терминалов в здании. Если сотрудник научно-исследовательской организации не может получить какую-либо справку, находясь на рабочем месте, то он несомненно обратится за ней к библиотечной системе. Терминалы, установленные в рабочих кабинетах, как это предлагается в проекте системы коллективного доступа Массачусетского технологического института, являются пока что слишком дорогим удовольствием для большинства современных библиотечных систем, однако желательно, чтобы проектно-конструкторские работы велись именно в этом направлении, чтобы полностью исполняется преимущества прямого контакта с системой, который обеспечивается терминалами. Опрос мнений, проведенный недавно среди научных работников, показал, что многие из них предпочитают пользоваться услугами удобного и находящегося под рукой источника информации, даже если ожидаемые информационные возможности этого источника окажутся меньше, чем у другого, менее удобного в обращении средства. Если стандартное программное обеспечение сможет гарантировать

достаточно быстрое получение ответов на запросы, поступаемые с терминалов, то система коллективного доступа, по-видимому, будет наиболее привлекательной для возможных библиотечных пользователей

#### 4.3. ОСНОВНАЯ ПАМЯТЬ

Основная память вычислительной машины была кратко описана в гл. 2. Там же было отмечено, что подавляющее превосходство в быстродействии основной памяти по сравнению с устройствами ввода-вывода данных явилось большим стимулом для разработки эффективных мультипрограммных средств, позволяющих полнее использовать возможности основной памяти. Даже с учетом того, что огромное количество команд должно быть выполнено для обработки и интерпретации данных, поступающих через терминалы, а также для расшифровки запросов пользователей и для перебрасывания записей информации с внешних запоминающих устройств в основную память и обратно, быстродействие основной памяти или центрального процессора вряд ли окажется лимитирующим фактором при рассмотрении вопроса о времени, необходимом для получения ответа на запрос, поступивший с терминала. Следует отметить, что в вычислительной технике наблюдается тенденция к повышению быстродействия центральных процессоров, и рабочие характеристики вычислительных машин нового поколения как раз подтверждают это обстоятельство. Так, передача восьми символов информации в арифметическое устройство (эта операция необходима для выполнения машинной команды) будет занимать только 160 наносекунд (т. е. десятимиллионные доли секунды). Вычислительные машины нового поколения в два, а то и в четыре раза превосходят по быстродействию выпущенные ранее модели машин. Поскольку в то же самое время снижается стоимость основной памяти по сравнению с другими компонентами оборудования, то становится все более и более экономически целесообразным выпуск машин с большой оперативной памятью, что позволяет обеспечить эффективное взаимодействие с терминалами и обработку больших массивов оперативных данных.

#### 4.4. ВСПОМОГАТЕЛЬНЫЕ ЗАПОМИНАЮЩИЕ УСТРОЙСТВА

Разработки, направленные на увеличение быстродействия центрального процессора, должны сопровождаться повышенным вниманием к емкости и скорости вспомогательных запоминающих устройств. Действительно, весьма незначительным будет эффект от использования сверхбыстрого центрального процессора, если он будет вынужден простаивать вследствие того, что обрабатываемые

им данные не могут быть достаточно быстро переданы из вспомогательной памяти. Многие пользователи, работающие на крупных вычислительных машинах, требуют, чтобы большие файлы были постоянно доступны для машины. Мы уже отмечали, что эксплуатируемые в настоящее время запоминающие устройства с произвольным доступом не отвечают требованиям с точки зрения как быстродействия, так и объема памяти, предъявляемым к таким файлам в сфере широкого информационного обслуживания, и поэтому разработчики вычислительной техники делают все, чтобы исправить создавшееся положение. В этой связи опять же следует сказать, что модели нового поколения машин имеют весьма внушительные преимущества. Так, например, максимальная емкость запоминающего устройства на магнитных дисках достигнет 800 млн. восьмиразрядных байтов, а скорость передачи данных в основную память будет равна 800 тыс. байтов в секунду. Время, затрачиваемое на перемещение магнитных головок диска к требуемой записи, сокращается вдвое. Одно крупное коммерческое предприятие объявило о своем намерении хранить около 2400 млн. байтов информации на дисковом запоминающем устройстве, используемом для информационной системы и банка данных. Растущая способность машин к хранению и обработке больших банков данных означает, что различные приложения, связанные с использованием больших файлов, теперь могут быть продуманы на уровне, который до последнего времени представлялся недостижимым.

#### 4.5. ДРУГОЕ ОБОРУДОВАНИЕ

В настоящее время целый ряд компаний специализируется на выпуске вспомогательного оборудования для вычислительных машин, причем число различных типов устройств, относящихся к этому оборудованию, постоянно растет. Здесь мы отметим только те разработки, которые могут оказаться важными при использовании машин в библиотечных системах.

Так, например, можно подключить терминал к устройству считывания микрофильмов. Если терминал используется для передачи запроса на некоторый документ из фонда библиотеки, то документ, хранящийся в виде микрофильма, может быть автоматически вызван для изображения на экране фильмоскопа. Хотя такой способ хранения информации представляется несколько расточительным, тем не менее какая-то форма микрохранилища необходима для решения проблемы вместимости, стоящей перед большими библиотеками. Магнитные карты представляют собой более современную форму микрохранилища, которая выдержала испытание практикой. В большинстве библиотек до сих пор предпочитают иметь свободный доступ к книгам, стоящим на полках хранилища, при условии, что информация об этих книгах может быть сообщена пользователю,

находящемуся за терминалом. Однако устройства считывания микрофильмов имеют то преимущество, что в случае необходимости они могут быть использованы для получения фотокопий.

Другим приятным достижением будет усовершенствование оборудования для оптического чтения символов информации. При использовании такого оборудования все документы могут быть введены в машину, причем они автоматически читаются без предварительной обработки их на устройстве подготовки данных. Продаваемое в настоящее время оптическое оборудование пока что более или менее удачно осуществляет только распознавание символов вводимой информации; нет сомнений в том, что усовершенствования существующего оборудования могли бы помочь устранить многие затраты, связанные с хранением и поиском информации. Пэйн [27] рассмотрел некоторые практические проблемы, касающиеся установки и использования современного оборудования для оптического чтения информации.

В этой главе был проведен обзор эксплуатируемой в настоящее время аппаратуры и ее связи с автоматизированной библиотечной системой. Однако для того, чтобы машина могла решать задачи и выдавать желаемые результаты, она должна быть снабжена стандартным программным обеспечением и пакетами прикладных программ, что и будет предметом нашего рассмотрения в следующей главе.

## ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ И ПРОГРАММИРОВАНИЕ

---

### 5.1. СТАНДАРТНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Основная идея операционной системы была кратко рассмотрена в гл. 2, где объяснялось, что для обеспечения эффективного прохождения потока задач и надежности вычислительной системы большое число функций обслуживающего характера выполняется операционной системой. К этим функциям относятся организация обмена данными между основной памятью и устройствами ввода-вывода, планирование и управление прохождением через машину программ, составленных пользователями, и перевод на соответствующий машинный код программ, написанных на различных алгоритмических языках, значительно облегчающих процесс программирования задач. Эта глава посвящена некоторым особенностям, касающимся использования стандартного программного обеспечения, а именно средств обработки файлов. При этом предполагается, что все разработанное для данной машины программное обеспечение доступно любому ее пользователю. Речь пойдет о том, какого типа программы необходимо написать или выбрать из имеющихся пакетов прикладных программ, чтобы реализовать библиотечную систему, модель которой была рассмотрена выше. Эти программы будут разбиты на ряд групп.

### 5.2. УПРАВЛЕНИЕ ТЕРМИНАЛАМИ

Модель системы, предложенная выше при описывании аппаратуры вычислительной машины, предусматривает наличие нескольких терминалов, каждый из которых непосредственно связан с центральным процессором. Независимо от того, какая информация фактически передается по линиям связи от этих терминалов, обязательно должна быть некая весьма сложная программа, которая осуществляет контроль над всеми «активными», т. е. взаимодействующими с системой, терминалами и обрабатывает сообщения, поступающие или отправляемые по соответствующим линиям связи. Такого рода средство обслуживания, называемое «опросом» теле-тайпов, как правило, уже имеется в операционной системе, входящей в стандартное программное обеспечение, однако целый ряд аналогичных программ может быть получен независимо от изготовителя машины или даже составлен самими пользователями, в рас-

поряжение которых предоставлены средства специального программного обслуживания. В связи с тем что к машине может быть подключено большое число устройств и в процессе работы системы будет использоваться различное число терминалов, целесообразно, чтобы управление терминалами осуществлялось некоторой самостоятельной компонентой или, как теперь принято говорить, модулем программного обеспечения. Этот модуль управления терминалами будет последовательно один за другим «опрашивать» терминалы, определяя, какие из них находятся в активном состоянии, и осуществлять посимвольный прием и запоминание каждого сообщения, поступающего от активного терминала, до тех пор, пока пользователь, находящийся за данным терминалом, не укажет, что его сообщение закончено. Далее сообщение будет передано другому модулю программного обеспечения, который, проанализировав его, выполнит соответствующие действия. После того как ответ на сообщение, поступившее с терминала, будет готов, модуль управления терминалами определит, какой из терминалов ожидает данный ответ, а затем опять-таки посимвольно передаст весь ответ пользователю. Поскольку значительную часть времени управляющий терминалами модуль будет проводить в ожидании передачи сообщений по линиям связи, можно совместить его работу с выполнением пакетов каких-нибудь других программ.

### 5.3. КОМАНДНЫЕ ЯЗЫКИ

После того как сообщение, поступившее с терминала, полностью собрано в основной памяти, его необходимо проверить, а затем интерпретировать так, чтобы выполнить требуемые пользователем действия. Так, например, сотрудник библиотеки, подойдя к терминалу, может выяснить, разрешено ли ему работать с системой, а затем сообщить, что он хочет изменить некоторый конкретный файл, скажем записать новые поступления в основной файл. Такого рода общение с системой можно осуществить с помощью так называемого командного языка. Последний представляет собой определенный набор слов или команд, которые пользователь может печатать на своем терминале, иницируя тем самым работу конкретных компонент системы. Например, слово «алло» может быть стандартным сообщением о том, что пользователь хочет воспользоваться услугами системы. После того как интерпретатор командного языка получит сообщение «алло», он проверит, входит ли данное слово в список допустимых терминов языка, а затем передаст управление подпрограмме, которая зафиксирует переход терминала в активное состояние. Возможно, что на терминал пользователя будет послано ответное сообщение, в котором ему будет предложено указать свое имя. Когда пользователь в ответ на это сообщит свое имя, оно также будет проверено по списку имен тех лиц, которым разрешено работать

с системой. Если имя пользователя есть в указанном списке, то он может напечатать команду, допустим, такого вида: «КОРРЕКТИРУЕМЫЙ ФАЙЛ = ОСНОВНОЙ». Когда командный интерпретатор получит это сообщение, он передаст управление подпрограммам обработки файлов, сообщив им также сведения о том, какой файл хочет изменить пользователь. На каждом этапе получение очередного сообщения пользователя может быть подтверждено соответствующей программой, так что он знает, когда можно печатать следующую команду.

Вопрос о том, каким должен быть командный язык или какие команды в него включать, связан с использованием языка для поиска данных и будет обсуждаться несколько позже, но, как бы вопрос ни был решен, язык будет основываться на рассмотренных только что принципах. При этом возможно, что либо будут заранее определены все допустимые в данном языке команды, либо, как это делается в развитых системах, пользователю будет разрешено самому определять собственные команды и включать их в язык общения с системой. Однако в том другом случае работа за терминалом будет происходить в форме некоего диалога между пользователем за терминалом и машиной, в котором каждая команда, поступающая от пользователя, проверяется и интерпретируется как запрос на определенные услуги, выполняемые конкретной программой или группой программ внутри системы. Отметим также, что модуль управления терминалами, командный интерпретатор и подпрограммы, вызываемые различными командами, все вместе могут выполняться в мультипрограммном режиме на одной машине.

#### 5.4. ПОИСКОВЫЙ ЯЗЫК

Итак, мы познакомились с двумя компонентами программ, которые необходимы для нормальной работы библиотечной системы, использующей терминалы: интерпретатором командного языка и набором подпрограмм, обеспечивающих доступ к файлам и их обработку. Подпрограммы обработки могут быть самыми различными в зависимости от тех информационных услуг, которые будут предоставляться системой. К подпрограммам, связанным со средствами обработки файлов, мы вернемся снова в следующей главе. Сейчас же основное внимание уделим вопросу взаимодействия между системой и пользователем, особенно с неосведомленным пользователем; другими словами, обсудим командный язык такого типа, который соответствует обстоятельствам, характерным для информационного поиска.

Казалось бы, в качестве такого языка можно использовать какой-нибудь язык низкого уровня, который сродни машинному языку, непосредственно воспринимаемому машиной. Операционная

система, как правило, пишется на одном из языков низкого уровня. Для наших же целей языки низкого уровня не годятся, поскольку они требуют значительных знаний и навыков программирования. Несомненно, что пользователями библиотечной системы будут не только программисты.

Поэтому мы должны попытаться использовать какой-нибудь язык высокого уровня, с которым работать значительно легче, но, правда, составленные на нем тексты должны быть переведены или интерпретированы операционной системой, прежде чем они станут иметь какой-то смысл для центрального процессора. Выбранный язык должен быть ориентирован на работу с большими файлами, содержащими информацию, представленную в символьной форме, например названия работ или имена авторов.

Стандартные научные и коммерческие языки, подобные таким наиболее широко используемым языкам, как ФОРТРАН, АЛГОЛ или КОБОЛ, не пригодны для использования в той области, где важна символьная обработка. Кроме того, эти языки являются процедурными, т. е. предназначенными для описания различных вычислительных процессов, а не командными языками. То же самое можно сказать о двух проблемно-ориентированных языках: ЛИСП (язык для обработки списков) и КОМИТ. Эти языки критиковались Сэлтоном [18], являющимся пионером в области информационного поиска, за то, что они не приспособлены для работы с большими файлами, способ организации которых наиболее удобен для информационного поиска, а также за то, что на них неудобно описывать небольшие процессы математических вычислений, которые могут потребоваться в течение поиска. Выводы Сэлтона сводятся к тому, что существующее программное обеспечение фактически не соответствует задачам информационного поиска и что совершенно необходим какой-то новый язык высокого уровня с обязательным учетом специфических потребностей библиотечных пользователей.

Несомненно, если мы намереваемся разрешить обычному пользователю обращаться к машине через терминал, то нам необходим такой командный язык, которым можно было бы легко овладеть, причем лексика этого языка должна представляться естественной пользователю. Любой язык, который требует овладения справочником команд и знания приемов программирования, снижает выгоды от непосредственного доступа, обеспечиваемого системой. Даже если вблизи от пользователя будет находиться специалист, готовый оказать ему помощь, все равно необходимо некоторое легко доступное средство общения между пользователем и системой. Тем не менее желательно сохранить возможность использования какого-нибудь стандартного языка высокого уровня для пользователей-специалистов. Одно дело рассчитывать на то, что научный сотрудник согласится овладеть языком ФОРТРАН, с помощью которого он сможет, например, обработать в течение дня результаты исследований,

полученные за год, и совсем другое дело заставить его изучать КОМИТ для того, чтобы он мог прочитать двухстраничную статью, которую он может без помощи машины за пять минут найти в журнале. В руководстве по языку КОМИТ [19] утверждается, что этот язык «легок для изучения», но весьма спорно, так ли это на самом деле даже для опытного программиста, не говоря уже о неквалифицированном пользователе.

С точки зрения пользователя, необходимо такое средство общения с системой, которое он считает естественным и простым. Представляется очевидным включение в это средство некоторых форм обычного английского<sup>1)</sup> языка. При этом не надо думать, что пользователю разрешается употреблять все слова, какие он пожелает, и в любом порядке, но ведь любому человеку понятны стилизованные идиоматические обороты типа «нагрей до цвета побежалости», и подобные обороты используются в довольно сложных системах обозначений, например при описании структур узоров, шахматных комбинаций и музыкальных партитур. Кроме того, большая часть информации в библиотечной системе будет, вероятно, представлена на естественном языке (названия книг, предметные рубрики), и следует по возможности использовать это обстоятельство.

### 5.5. ЕСТЕСТВЕННЫЙ ЯЗЫК

Следует особо подчеркнуть, что успех или неудача поиска во многом зависит от точности формулировки запроса на поиск и от точности задания первоначальной операции запоминания информации. Это соображение учитывается в дальнейших рассмотренных. Однако верно и то, что пользователь, как правило, пунктуален, работая на привычном для него языке, и если поисковый язык подобен языку, на котором написано большинство документов в библиотечной системе, то в теоретическом аспекте вероятность успешного проведения информационного поиска значительно возрастает. На фоне этого обстоятельства необходимо оценить трудности, встречающиеся при автоматическом анализе естественных языков.

При решении задачи автоматического анализа любого естественного языка возникает ряд проблем, а если говорить об английском языке, то их особенно много, поскольку в нем мало способов образования форм слов с помощью окончаний и исключительно много двусмысленностей. Первой существенной проблемой является синтаксический анализ. Стремление осуществить автоматический перевод с языка на язык с помощью машины побудило специалистов в областях вычислительной техники и лингвистики взяться за решение очень трудных задач, и поэтому в настоящее время уже есть методы, позволяющие вполне успешно выполнять синтаксический

<sup>1)</sup> Или русского.— *Прим. ред.*

анализ. Введенное предложение может быть разобрано в виде некой структуры отдельных элементов, о каждом из которых установлено, что он играет самостоятельную роль в предложении. Возможные в языке многочисленные отклонения и двусмысленности вызывают трудности при разработке программ грамматического разбора такого рода, тем не менее на этом пути могут быть получены вполне хорошие результаты, особенно в тех случаях, когда критерии ориентированы на конкретные области применения языка.

Другой подход к проблеме синтаксического анализа, который можно назвать методом подгонки, подразумевает программирование для каждого встречающегося в тексте слова по своим правилам с учетом исключительных случаев. При этом предполагается, что машина «учится», накапливая опыт. Такой подход, называемый эвристическим программированием, пока еще находится в стадии эксперимента. Производились поиски более систематичного метода составления точного логичного описания естественного языка, поскольку уже доказано, что наличие формального описания позволяет разработать высокоэффективные методы грамматического разбора для языков высокого уровня, подобных АЛГОЛу. Однако естественный язык, по-видимому, не поддается анализу такого рода, и история математической лингвистики, которая потерпела неудачу в своей попытке представить естественный язык как некое исчисление, дает основания предполагать, что этот подход обречен на окончательную неудачу.

Еще более существенной проблемой является анализ семантики, т. е. значения слов и оборотов, естественного языка. Система, обслуживающая терминалы, должна вести разговор с пользователем таким образом, чтобы на каждом этапе этого разговора у пользователя создавалось впечатление, будто он понят машиной. Конечно, в действительности дело обстоит так, что в результате анализа определенных входных данных, поступивших с терминала, машина инициирует выполнение соответствующего набора процедур и операций, а также печатает на пульте терминала определенные ответные сообщения. В настоящее время есть несколько интересных экспериментальных систем, с которыми пользователь может общаться на естественном языке в рассмотренном выше режиме «вопрос — ответ». Одной из лучших является система BASEBALL, отвечающая на конкретные вопросы о бейсбольных командах, их турнирном положении и т. д. [20]. Другим впечатляющим примером системы разговорного типа является система ELIZA [21], которая ведет кажущуюся человеческой разумную беседу при помощи простой программы «вопрос — ответ». В процессе разговора эта программа повторяет значительную часть вопросов и комментариев пользователя, заменяя в них личные местоимения «я» на «вы» и вставляя перед ними короткие стандартные реплики типа: «Сообщите» еще что-нибудь о ...», «Поясните...». Этот пример наглядно показывает, что

машине нет необходимости «обладать разумом», чтобы поддерживать хороший разговор с пользователем.

По-видимому, необходимое нам средство общения является неким компромиссом с одной стороны между совершенно искусственным языком, который может навязать пользователю совершенно неподходящий для него способ общения либо потому, что этот язык просто непосилен для пользователя, либо из-за отсутствия в языке достаточно удобных средств общения, и с другой стороны между сложным естественным языком, который только с трудом анализируется автоматически, причем для этого нужна очень большая и сложная программа, на выполнение которой может потребоваться много времени. Если удастся придумать соответствующий выразительный набор команд общения, то многие другие вопросы могут быть решены путем предоставления машине права управлять разговором и диктовать каждый его очередной шаг.

## 5.6. АББРЕВИАТУРА

Прежде чем перейти к обсуждению файлов и способов их организации при информационном поиске, следует отметить два сопутствующих обстоятельства. Во-первых, когда в системе нужно хранить много информации, представленной в обычной символической форме (названия, фамилии авторов), файлы могут оказаться очень дорогим средством с точки зрения расхода памяти; это касается как дисков, на которых размещаются файлы, так и основной памяти, в которую файлы считываются в процессе обработки. В связи с этим заслуживают внимания методы сокращения или кодирования информации. Во-вторых, особенно полезно, если записи можно закодировать таким образом, что они окажутся одинаковой длины, тогда программе обработки файлов не надо будет вычислять длину каждой записи перед тем, как считывать ее в оперативную память.

В настоящее время предложено несколько методов, позволяющих использовать резервы избыточности естественного языка и практически исключающих при этом возможность возникновения в информации двусмысленности из-за одинаковой кодировки двух различных элементов. Основной интерес представляют коды, которые основаны на сокращении или беспорядочном перемешивании фактической информации, подлежащей хранению, а не на полном преобразовании элементов или присвоении значений случайно выбранных кодов. Детальный обзор, сделанный Боурном и Фордом [23], охватывает ряд методов, в числе которых авторы отличают такие, как выборочное опускание букв, основанное на учете позиции или частотности или же производимое случайным образом; простое сокращение записей; перетасовку букв и способы кодирования, подобные методу усечения звуков, согласно которому гласные звуки игнорируются, а согласным присваиваются цифровые значения. Все

эти методы примерно одинаково эффективны, и поэтому нельзя рекомендовать какой-нибудь из них по сравнению с другими. Однако есть одно дополнительное соображение, которое необходимо учитывать.

### 5.7. ИСКЛЮЧЕНИЕ ОШИБОК

Как уже отмечалось выше, очень важно предпринять меры, чтобы избежать неправильной классификации вводимой информации или неудачного поиска по запросу вследствие элементарных ошибок, допущенных во входных данных, например, орфографических ошибок. Рассмотрим случай, когда студент, записав на лекции фамилию заинтересовавшего его автора, пытается найти работу этого автора в библиотеке. Если он допустил в фамилии незначительную орфографическую ошибку, то вероятно, что в условиях обычной библиотечной системы он обнаружит эту ошибку, просматривая другие книги, находящиеся на той же полке, где находится нужная ему работа, или другие карты каталога из того же ящика. Если же студент хочет воспользоваться услугами автоматизированной библиотечной системы, то, работая за терминалом, он должен совершенно точно сформулировать свой запрос, иначе он потерпит неудачу. Еще хуже, если орфографическую ошибку, печатая сведения о вновь поступившей книге, допустит сотрудник библиотеки, занимающийся первичной обработкой материалов. Фактически после этого работа окажется недоступной читателю.

Положение может быть улучшено путем использования некоторых видов аббревиатур или кодирования. Если при кодировании очень похожим элементам информации присваиваются значения одинаковых кодовых комбинаций, то вероятно, что одна орфографическая ошибка не сможет вызвать потерю документа. Так, если один и тот же способ кодирования используется как во время запоминания информации, так и на этапе поиска, то искаженный элемент может быть найден с таким же успехом, как и любые правильно написанные элементы. Платой за это является то, что пользователь может получить в качестве ответа несколько похожих по написанию элементов наряду с тем, который ему нужен, но это, наверное, все же предпочтительнее, чем потеря информации совсем. Подобным аббревиатурным кодом, используемым в информационном поиске, является SPECOL [24]. Пример использования метода усечения звуков приведен на стр. 84.

При обработке определенных групп данных можно попробовать произвести автоматическое исправление орфографических ошибок на этапе ввода элементов информации. Решение такого рода задачи главным образом сводится к хранению в памяти машины некоторого словаря стандартных слов и проверке каждого вводимого слова по этому словарю. Более интересные эксперименты связаны с анализом

---

некоторых групп букв, например пар или троек, обозначающих один звук. Проводя исследования в этом направлении, Карлсон [22] использовал английские имена, взятые из церковной книги записей. Опубликованные им результаты свидетельствуют о значительном успехе на пути исправления ошибок, возникающих вследствие искажения вводимых символов оборудованием оптического чтения данных. Как выбор поискового языка, так и способ хранения информации в системе в некоторой степени определяются структурой файлов, используемых для размещения информации. Эту тему мы обсудим в следующей главе.

## 6

### СТРУКТУРА ФАЙЛОВ

---

#### 6.1. ВНЕШНЯЯ СТРУКТУРА ФАЙЛОВ

В этой главе будут рассмотрены различные способы организации данных в больших файлах, предназначенных для использования в информационно-поисковых системах.

При этом основное внимание будет уделено следующим трем вопросам: каким образом лучше всего располагать файлы на внешних запоминающих устройствах (внешних носителях), как можно организовать чтение этих файлов в оперативную память, как может быть выполнен быстрый и экономный просмотр файлов после чтения их в оперативную память? Обсуждение этих вопросов будет проведено с учетом того, что обращения к файлам, связанные как с чтением содержащихся в них данных, так и с изменением или дополнением файлов новой информацией, будут происходить часто и через произвольные интервалы времени.

В силу изложенных ранее причин в информационно-поисковой системе следует применять внешние запоминающие устройства с произвольной выборкой данных. Используя в качестве примера носитель на магнитных дисках, рассмотрим в общих чертах способ хранения файлов и записей на таких устройствах. Отметим, что операционная система, поставляемая изготовителем совместно с аппаратурой, предоставляет пользователю средства, помогающие ему создавать и эксплуатировать файлы, а также обеспечивает защиту файлов от неправомерных обращений.

Местоположение каждой записи на диске определяется посредством адреса, состоящего из номера дорожки диска, на которой будет найдена данная запись, и номера записи в пределах дорожки. Таким образом, чтобы обратиться к некоторой отдельной записи, операционная система должна знать как номер дорожки на диске, так и номер самой искомой записи. Тем не менее пользователю нет необходимости знать адрес записи или файла, к которому он хочет обратиться. Как правило, дело обстоит так, что пользователь, желающий образовать новый файл, сообщает операционной системе имя файла, а также указывает, какой объем памяти (количество дорожек) необходим для хранения файла и, возможно, на каких из доступных ему дисках он хочет этот файл разместить. С этого времени все возможные обращения к файлу осуществляются посредством указания имени, которое использовалось при образовании

файла. Для этого операционная система о каждом образованном файле хранит такие сведения, как имя файла, указатель конкретного диска, на котором этот файл хранится, а также номер дорожки и номер той записи на этой дорожке, с которой начинается файл. Поэтому если некий пользователь сообщит системе, что он хочет прочитать файл, называемый АВТОРЫ, то система выяснит, какие дорожки были выделены для размещения файла в момент его образования, а затем установит магнитные головки диска так, чтобы можно было прочесть первую запись этого файла.

Если пользователь далее пожелает найти некоторую конкретную запись, то он, вероятно, будет ссылаться на какой-нибудь «ключ» или заглавную часть файла. Например, в случае файла, содержащего фамилии авторов, в качестве ключа можно взять фамилию автора, причем при образовании файла его можно упорядочить в алфавитном порядке. Пользователь может найти необходимую ему информацию несколькими способами: просматривая весь файл до тех пор, пока не встретит нужную запись, или сообщая операционной системе информацию, позволяющую определить местоположение той записи, которая содержит нужные ему сведения. Эти способы будут рассмотрены в следующем разделе.

Сейчас же особо подчеркнем, что образование файлов и хранение сведений о них обеспечивается операционной системой. Кроме того, система осуществляет «привязку» к файлу согласно запросу, т. е. устанавливает магнитные головки диска так, чтобы пользователь мог читать или писать информацию, даже не имея каких-либо знаний о точном местоположении файла.

## 6.2. МЕТОДЫ ДОСТУПА

На самом деле операционная система, как правило, не ограничивается простой «привязкой» к началу файла. В составе программного обеспечения предусмотрены подпрограммы, осуществляющие чтение и запись по указанию пользователя отдельных записей файла и передачу этих записей программе пользователя для обработки. Эти подпрограммы обычно известны как средства доступа. Отчасти они необходимы для предохранения программ пользователей от записи информации в запрещенные области диска, т. е. в области, которые не отведены для хранения файлов, используемых этими программами.

Есть два основных типа средств доступа, которые основаны на способе организации данных и на способе обращения к данным. Первый называется последовательным методом доступа. При использовании подпрограмм этого метода любой запрос на чтение из файла вызывает передачу программе пользователя следующей по очереди записи файла. Поскольку пользователь начинает работу с файлом после того, как магнитные головки диска установлены

у первой записи, то в случае последовательного метода доступа пользовательской программе будет передана первая запись, затем вторая и т. д. Если программа пользователя предназначена для печати списка всех авторов, упомянутых в файле АВТОРЫ, то такой метод доступа является идеальным. Записи появляются в оперативной памяти в том же порядке, в каком они были первоначально занесены в файл, и программа пользователя может поочередно обработать каждую запись.

С другой стороны, если пользователь хочет найти одну запись с конкретным ключом, то ясно, что было бы чрезвычайно расточительным читать подряд каждую запись до тех пор, пока не встретится искомая. Пользователь должен учитывать, что время, затрачиваемое на выполнение операции ввода-вывода, даже если мы имеем дело с диском, слишком велико по сравнению со временем выполнения операций процессором. Возможность обращения к отдельным записям обеспечивается индексно-последовательным методом доступа. Для этого операционная система ведет таблицы, или указатели первой записи на каждом диске (если файл размещен на нескольких дисках) и первой записи на каждой дорожке диска. В таблицах для каждой первой записи сообщаются ключ и адрес. Поэтому магнитные головки диска всегда могут быть автоматически установлены именно на той дорожке, которая содержит требуемую запись.

Однако подпрограммы, используемые для выполнения индексного поиска, сложны, и этот метод доступа может привести к замедлению процесса поиска при некоторых характеристиках основной памяти (для малых файлов более экономно использовать последовательный метод доступа, читая и проверяя записи по очереди).

Кроме того, возникают проблемы, связанные с внесением изменений в индексно-последовательный файл, поскольку на каждой дорожке должно быть зарезервировано много места для любых дополнительных записей. Если к файлу добавляется слишком много записей, то они «переполняют» ту дорожку, где должны были бы храниться, и заносятся в некую дополнительную область. В связи с этим возникает необходимость в использовании дополнительных указателей и таблиц для этой области «переполнений», что приводит к значительному замедлению поиска.

Другой подход заключается в том, чтобы не определять местоположение отдельных записей на диске, а считывать в основную память целую порцию файла, содержащую ряд связанных записей, и затем искать среди них требуемую запись, поскольку такой процесс будет быстрее, чем поиск на диске. На какие порции (или «черпаки», как их иногда называют) будет делиться файл, зависит от его специфики. Возможно, что для каждой буквы алфавита (если говорить о файле АВТОРЫ) будет выделена одна порция. В примерах из гл. 8 рассматривается малый файл, который мог бы соответствовать одной порции. Другими словами, весь файл можно было

бы прочитать в оперативную память, как будто он является одной записью, а затем просматривать его в поисках нужной записи. В этом случае легкость, с какой может быть выполнен просмотр, в значительной степени определяется внутренней организацией записей на файле.

Логическим развитием подхода такого типа является идея «базы данных», которая в настоящее время вызывает большой интерес. Вместо того чтобы, как обычно, образовывать несколько файлов, каждый из которых предназначен для определенной информации, образуют просто один «большой» файл, содержащий всю информацию. При этом мы избавились бы от необходимости иметь один файл для хранения фамилий авторов, один для названий или рубрик, один для статистики и один для новых библиотечных поступлений. Сторонники «базы данных» утверждают, что наличие нескольких файлов является расточительством, поскольку весьма много информации дублируется на различных файлах. Более экономно хранить информацию в одном «экземпляре» и попытаться обеспечить возможность доступа к этой информации несколькими способами, используя, например, для этого систему ссылок между элементами файла. По мере развития этой идеи, а также с увеличением быстродействия и объема основной памяти все более необходимым становится детальный анализ внутренней структуры файлов.

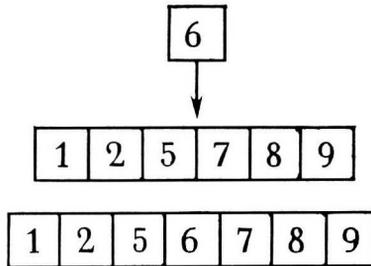
### 6.3. ВНУТРЕННЯЯ СТРУКТУРА ФАЙЛОВ

Если будет применяться метод просмотра записей в основной памяти, то необходимо предпринять меры, обеспечивающие такое расположение записей в файле, чтобы облегчить выполнение всех операций, требующихся при работе с файлами. Нас интересуют такие операции, как чтение элемента из файла, изменение элемента, включение дополнительной записи в соответствующее место уже имеющейся последовательности записей файла, исключение ненужной или устаревшей записи. Обычно, чем легче добавить новую запись или изменить файл, тем дольше приходится просматривать файл для того, чтобы определить местоположение данной записи. Ниже мы рассмотрим три возможные структуры файлов.

#### 6.3.1. ПОСЛЕДОВАТЕЛЬНЫЕ ФАЙЛЫ

В таких файлах элементы располагаются один за другим в определенном логическом порядке. В качестве примера опять-таки можно привести файл АВТОРЫ, записи которого упорядочены по алфавиту. Для того чтобы проиллюстрировать описываемые структуры файлов, допустим, что есть некоторый файл, предназначенный для хранения цифр от 1 до 9, и что первоначально при образовании этого файла в него заносятся цифры 1, 2, 5, 7, 8 и 9. Используя

способ двоичного поиска, можно довольно легко определить местоположение конкретного элемента в последовательном файле, организованном в соответствии с некоторым известным порядком. Этот способ состоит в сравнении ключа требуемого элемента с ключом элемента, находящегося в середине файла. Если значение ключа требуемого элемента окажется меньше, то этот элемент находится в первой половине файла; если же больше, то во второй. Далее описанная процедура повторяется на половине файла, содержащей требуемую запись, и это происходит до тех пор, пока запись не



Р и с 1. Изменение последовательного файла.

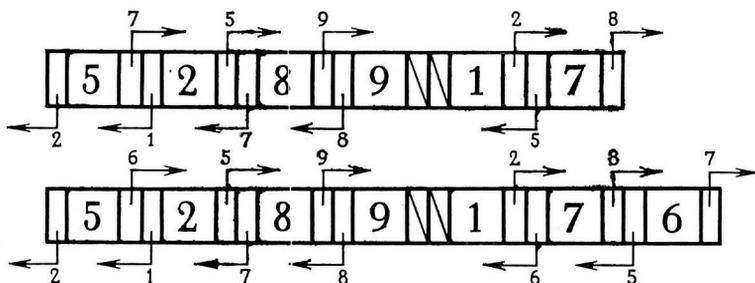
будет найдена. Таким образом, в процессе двоичного поиска после каждой проверки исключается половина возможных элементов файла.

Теперь допустим, что требуется, как это показано на рис. 1, добавить к файлу цифру 6. Ясно, что поскольку элементы файла расположены в памяти последовательно один за другим, то включение в эту последовательность еще одной записи может быть выполнено только путем физического сдвига целой порции файла в памяти. В результате такого сдвига будет освобождено в соответствующей позиции последовательности место для нового элемента файла. В случае больших файлов находящаяся в оперативной памяти порция может оказаться только лишь частью всего файла. В таком случае сразу возникает вопрос, не придется ли перестраивать другие порции файла после того, как измененная порция будет записана обратно на диск.

### 6.3.2. СПИСКОВЫЕ ФАЙЛЫ

Файл со списковой структурой в принципе отличается от последовательного файла. В таком файле физический порядок следования записей не связан с логическим порядком информационных элементов файла. Элементы располагаются в файле в том порядке, в каком они добавлялись к файлу, однако каждая запись содержит,

помимо самого элемента, еще два указателя, первый из которых ссылается на элемент, логически предшествующий данному, а второй — на элемент, логически следующий за данным в файле. Процедура изменения файла теперь оказывается более простой, чем в случае последовательного файла. Любой новый элемент физически заносится в очередную запись в конце файла. Указатели логически предшествующего элемента и элемента, следующего за добавленным к файлу, соответствующим образом корректируются, а в указатели нового элемента заносятся ссылки на его логических соседей. Таким образом, по мере того как все новые и новые элементы добавляются в область в конце файла, единственное, что необходимо делать,



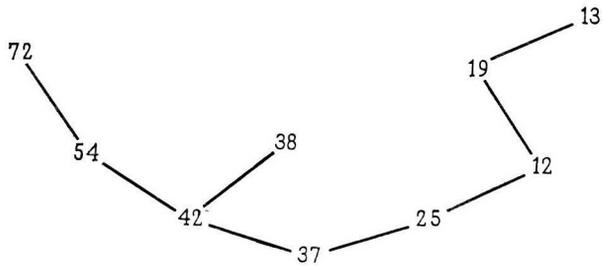
Р и с. 2. Списковый файл до и после изменения.

так это соответствующим образом изменять значение логических указателей в различных записях. Трудность возникает тогда, когда приходится просматривать файл для того, чтобы получить конкретную запись. В этом случае надо начинать поиск с логически первого элемента файла и, следуя далее по ссылкам указателей, сравнивать ключ требуемого элемента с ключом каждой просматриваемой записи. В отличие от просмотра последовательного файла в данном случае необходимо по очереди исследовать каждую запись до тех пор, пока не встретится запись с требуемым ключом.

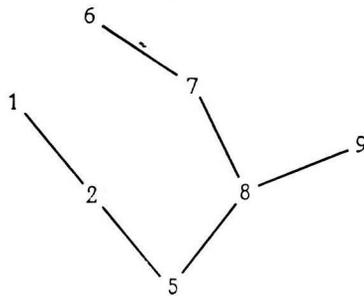
На рис. 2 изображен списковый файл, соответствующий файлу на рис. 1.

### 6.3.3. ДРЕВОВИДНЫЕ ФАЙЛЫ

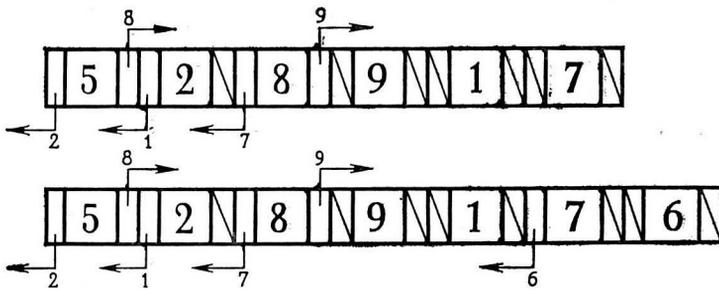
Компромиссом между рассмотренными выше двумя типами структур файлов является древовидная структура файла. Как и в случае спискового файла, в древовидном файле элементы могут размещаться в памяти в произвольном порядке, а также сохраняются указатели на логических соседей. Однако схема или структура файла выбирается так, что он похож на дерево. Типичная древовидная структура для несколько иного примера показана на рис. 3, а на рис. 4 и 5 иллюстрируется древовидная структура и соответствующий



Р и с. 3. Типичная древовидная структура.



Р и с. 4. Пример (из предыдущего раздела), представленный древовидной структурой.



Р и с. 5. Древовидный файл до и после изменения.

файл, применительно к тому же самому примеру, что был рассмотрен в предыдущем разделе. Просмотр файла в данном случае начинается с элемента, называемого «корнем» дерева. Если ключ требуемого элемента окажется меньше ключа, указанного в очередной записи, то следующей необходимо просматривать запись, определяемую указателем логически предшествующего элемента; если же больше, то — запись, содержащую элемент файла, логически следующий за только что просмотренным. Файл такого типа позволяет сочетать способ двоичного поиска и возможность связывать элементы без их физического переупорядочивания. Модифицированная версия связанного древовидного файла используется в примерах программ, рассматриваемых в гл. 8.

7.1. ОСНОВНЫЕ КРИТЕРИИ  
ИНФОРМАЦИОННОГО ПОИСКА

В ходе обсуждений, проведенных в гл. 3, мы рассмотрели два начальных логических шага на пути к полностью автоматизированной библиотечной системе. Первый из этих шагов предусматривает использование машины для составления стандартных библиотечных средств (например, таких, как указатели и карты каталога, облегчающие обработку) и документирования большого количества информации. Второй шаг, предусматривающий хранение каталожных сведений на машинных файлах, напротив, исключает некоторые из этих стандартных средств из библиотечного обихода и тем самым упрощает работу библиотечного персонала. Такого рода машинные приложения не были рассмотрены нами сколько-нибудь подробно, и заинтересованный читатель может получить сведения о различных проектах, связанных с реализацией этих приложений, из целого ряда источников. Эти проекты свидетельствуют о значительном прогрессе на пути к полностью автоматизированной системе, но рано или поздно они сталкиваются с проблемой информационного поиска.

Как отмечает Кимбер [6], «каталог следом за самой коллекцией книг лежит в сердце библиотечных операций». После того как принято решение хранить библиографические данные о собрании библиотечных материалов в машинных файлах, нужно использовать или разработать точный метод, обеспечивающий поиск этой информации. При таком методе человеку вместо того, чтобы обращаться к файлу карт каталога или справляться по какому-нибудь списку, нужно только передать машине определенный заказ, и соответствующая информация будет выбрана непосредственно из файлов, хранящихся в машине.

В этой главе рассматриваются пути решения такой задачи, а также те трудности, с которыми приходится сталкиваться.

Мы будем упоминать и рассматривать отдельные методы и проекты, но, как и раньше, главный акцент будет сделан на общей философии используемых методов, на необходимых ограничениях, а также критериях, позволяющих оценить преимущества и недостатки любого конкретного метода. Важно, чтобы всякий эксперимент, выполняемый в этой области машинных приложений, велся в правильном направлении и чтобы результаты таких экспериментов надлежащим образом и научно были оценены. Только таким путем

можно доказать пригодность машины для решения конкретных библиотечных задач.

Упомянутые в этой главе методы могут быть оценены с точки зрения следующих вопросов: как решается в конкретных случаях проблема больших файлов; где есть возможность для проникновения в информацию ошибки и предусмотрена ли соответствующая защита; с помощью какого средства связи организуется взаимодействие между пользователем и машиной и насколько оно удачно; увеличивается ли рабочая нагрузка библиотечного персонала вследствие использования данного метода, и если это так, то на кого возлагаются дополнительные обязанности; возможна ли экономия за счет однородности методов, т. е. можно ли одинаковую процедуру применять для каталогизации, поиска и распространения сведений, относящихся к одному и тому же фрагменту информации?

## 7.2. ЛОГИЧЕСКИЕ ПРИНЦИПЫ ЗАПОМИНАНИЯ И ПОИСКА ИНФОРМАЦИИ

При разработке поисковой системы приходится рассматривать ряд проблем, связанных с организацией фонда материалов таким способом, который позволил бы лучше осуществлять поиск информации. При этом весьма важно отличать те проблемы, которые являются общими для всех поисковых систем, от тех специфических, которые обусловлены использованием вычислительной машины. Многие критические статьи, посвященные машинному поиску, в равной степени относятся к любой системе и сводятся просто к изложению задач, стоящих перед библиотечными сотрудниками. В сущности, в них не формулируется никаких целей машинного поиска, за исключением того, что предлагается использовать машину для решения тех проблем, которые иначе не могут быть преодолены. Некоторые из трудностей такого типа будут здесь отмечены.

Традиционный способ организации фонда документов представляет собой некую разновидность классификационной системы, как правило, основанной на анализе предметных заголовков и часто имеющей иерархическую структуру. Хотя нет определенного логического довода, почему сами документы и средства поиска, такие, например, как карты каталога, следует хранить согласно одному и тому же принципу, книги в секциях хранилища часто размещаются с учетом их тематики и заголовков. С другой стороны, журналы и отчеты трудно хранить по такому принципу, и они обычно хранятся отдельно, так как содержащиеся в них статьи могут охватывать ряд различных дисциплин. Отдельные статьи трудно классифицировать, и поэтому их поиск может оказаться затруднительным. Десятичная система Дьюи является типичным примером такой классификационной системы и отражает много общих преимуществ и недостатков.

Эта система представляется простой для понимания, легкой в использовании и достаточно всеобъемлющей. Часто будет достаточно поверхностного просмотра книги, чтобы определить, к какому разделу системы она относится; таким образом, первичная обработка выполняется даже до того, как книга будет передана в систему, и является минутным делом. Тем не менее, как часто отмечается, использование такой классификационной системы связано с разнообразными трудностями, которые способны перечеркнуть любое удобство на стадии первичной обработки материалов. Эти трудности будут рассмотрены ниже.

Вместо таких систем часто применяются на практике некоторые формы координатного индексирования. Эта методика обходит основные трудности, связанные с классификацией, путем частичного отказа от классификации вообще. Идея заключается в том, что вместо того, чтобы относить документы к predetermined библотекарным категориям, элементы документации при поступлении изучаются и с ними связываются некие определенные термины, первоначально именованные общими терминами, а теперь обычно называемые ключевыми словами. Эти термины предназначаются для составления достаточно точного описания документа, используя которое можно будет найти документ путем поиска любых элементов документации с данным термином или с комбинацией терминов. Другими словами, указанные термины обеспечивают однозначное опознание документа при выполнении обычного поиска. Итак, документы, которые не могут быть очевидным образом отнесены к определенному классу, могут быть столь же легко индексированы (т. е. им приписываются соответствующие ключевые слова), как и обычные документы. Отметим здесь два интересных обстоятельства, а затем перейдем к сравнению тех характерных процессов в обеих возможных системах, для которых мы обсудим проблемы автоматизации. Во-первых, при координатном индексировании можно хранение самих документов осуществлять согласно принципам, отличным от того, что принято при хранении материалов, ссылающихся на эти документы. Ссылочные материалы и документы логически разделены друг от друга, что обеспечивает дополнительную гибкость методики, а средство поиска становится при этом настоящим средством связи с информацией. Другое обстоятельство, не приносящее ничего хорошего, заключается в том, что приходится выполнять значительный объем предварительной редакторской работы даже для самой простой книги, прежде чем удастся приписать ей соответствующие ключевые слова.

Интересно, что первые шаги в механизации информационного поиска оказались возможными благодаря координатному индексированию. Дело в том, что можно использовать перфокарты с пробивками в определенных колонках, представляющими ключевые слова или описатели. Каждая карта символизирует некоторый документ,

и документы отбираются посредством введения специальных штоков, или осей, через пробивки, представляющие требуемые описатели. Любая карта, символизирующая документ, которому не приписано конкретное ключевое слово, имеет в соответствующей этому слову колонке не пробивку, а открытый паз. Таким образом, всякий раз, когда штоки поднимаются, вместе с ними поднимаются те карты, которые символизируют документы со всеми заданными описателями. Другие карты, позволяющие штокам пройти через пазы, остаются в колоде карт.

Такого вида система работает только с ограниченным объемом материалов, и остается выяснить, существует ли какая-то процедура координатного индексирования, вполне удобная для приспособления к вычислительной системе. Чтобы подключить к делу вычислительную систему, необходимо рассматриваемую проблему разбить на ряд логических шагов. Осуществляя затем поэтапный анализ, мы сможем наглядно сравнивать полную классификацию с координатным индексированием, выявляя те области, где, вероятно, встретятся особые трудности, и оценивать обе методики с точки зрения сформулированных выше принципов.

В практике информационного поиска выделяются три самостоятельные процедуры. Одна процедура формулирует систему правил, с помощью которой будет происходить индексирование элементов информации, другая рассматривает отдельные элементы и индексирует их согласно принятой системе, а третья отыскивает соответствующие элементы через систему в ответ на запрос. Здесь мы будем рассматривать общие запросы на любые документы по данной теме или группе тем, поскольку они, очевидно, создают большую нагрузку для системы, чем запросы на специальные документы.

Каким же образом выполняет эти три процедуры система, классифицирующая документы по признаку предметности? На этапе формулирования классовой структуры и на этапе распределения документов по классам на основе их предметности могут встретиться два вида трудностей; одни из них объясняются существом дела, а другие — связью этих процедур с охватом человеческих знаний. При выделении классов мы встречаемся с проблемой учреждения некой иерархии того, что на первый взгляд кажется не допускающим организации. Действительно, совершенно не очевидно, почему человеческие знания могут быть четко подогнаны под различные категории. При попытке сделать это могут возникнуть альтернативные классовые структуры, и некоторые категории человеческих знаний не могут быть легко подогнаны к какой-то заданной структуре. Более того, однажды определенная система классификации оказывается жесткой, и ее нельзя легко расширить, сохранив при этом достигнутую четкость классов, для обслуживания новых областей знаний или уже учтенных областей, но с более высокой степенью детализации. Наглядным примером этого являет-

ся десятичная система Дьюи. Итак, мы сталкиваемся (1) с трудно-обрабатываемой природой предметности, которая означает, что любая попытка систематизировать или классифицировать, вероятно, вносит неестественность в классовую структуру; (2) с произвольностью нашего случайного выбора классов, которая приводит к только что рассмотренной жесткости структуры.

Аналогичные проблемы появляются на уровне принятия решения о том, к какому из классов нашей системы отнести конкретный документ. (1) Документ может не соответствовать никакому из классов или может содержать информацию, относящуюся к двум или более различным классам. В каталоге для таких документов могут храниться несколько одинаковых карт, по одной для каждого класса рубрик, но если сами документы размещаются на полках хранилища по классам, то необходимо либо выбрать некоторый класс, либо продублировать документы. (2) Возможно, что вследствие нашей недостаточной осведомленности в данном предмете мы примем ошибочное решение о том, к какому классу относится документ, и тогда практически документ окажется частично или полностью недоступным.

Вследствие того что ее классы раз и навсегда predeterminedены, рассмотренная система не ориентирована на пользователя, и требуется значительная реорганизация, если нужно приспособить ее классы к интересам пользователей. В системе могут возникать двусмысленные ситуации, при которых велика вероятность ошибки. Такие несообразности объясняются тем, что разные сотрудники библиотеки будут принимать различные решения при классификации одних и тех же документов. Что же касается третьей процедуры, то она относительно проста. Для получения необходимой информации пользователь обращается за справкой к классификационной системе и формулирует свой запрос в терминах предусмотренных предметных заголовков. Далее отыскиваются все документы, индексированные по данному заголовку, причем весьма вероятно, что многие из них окажутся не относящимися к делу. Однако ответственность за это лежит непосредственно на процедуре распределения документов по классам. Если какой-нибудь документ из-за невнимательности отнесен к несоответствующему классу и поиск осуществляется путем выявления всех документов, относящихся к указанному в запросе классу, то документ оказывается недоступным для практического использования. Поэтому успех системы зависит до некоторой степени от того, как сформулирована классовая структура, поскольку удачная структура оставляет мало места для двусмысленностей, и в очень большой степени от тех индивидуальных решений, которые были приняты сотрудниками библиотеки при распределении документов по классам. Это означает, что, хотя и легко будет классифицировать некоторые книги, все же в отдельных случаях может понадобиться заключение эксперта, позволяю

ще уменьшить вероятность ошибочной классификации. Таким образом, эта система будет успешно работать применительно к неспециальной литературе. Наибольший недостаток системы заключается в том, что обычно пользователь не оказывает влияния на процедуру поиска, поскольку он полагается на то, что документы первым делом правильно классифицированы.

Систему координатного индексирования можно сравнить с классификационной схемой по тем же самым трем этапам обработки информации. Здесь также возникает несколько проблем, мало чем отличающихся от рассмотренных ранее. Так, при определении правильных ключевых слов для документов по-прежнему присутствует субъективный элемент произвольности в выборе этих слов: один может решить, что данный набор терминов соответствующим образом описывает документ, а кто-нибудь выберет другую совокупность ключевых слов. Другими словами, процедура выбора ключевых слов представляется почти такой же важной, как и определение названий классов. В данном случае риск ошибочного описания документа совершенно не такой, поскольку, вероятно, описание должно быть сделано со значительным перекрытием, но, как будет показано, один или два термина могут придать описанию совсем другой смысл. Кроме того, теперь требуется особая тщательность даже для каталогизации довольно простого документа, если мы хотим точно выделить те термины, которые отличают этот документ от других работ на аналогичную тему. При координатном индексировании очень часто может потребоваться консультация эксперта, так что рабочая нагрузка на библиотечный персонал возрастает. Даже если самому автору предоставляется возможность составить описание, наилучшим образом характеризующее работу, сотрудник библиотеки все равно должен нести определенную ответственность за процедуру каталогизации, поскольку у автора, вероятно, будет к работе более субъективный подход, чем у профессионального специалиста по индексированию.

Если при координатном индексировании слегка уменьшается вероятность появления непоправимых ошибок на втором этапе, то первый и третий этапы оказываются соответственно более трудными. Этап поиска в данном случае уже не является процедурой простого сравнения, поскольку предполагаемый пользователь теперь может выбрать для определения своего запроса ряд терминов. Эти термины должны в последующем сопоставляться с терминами, являющимися ключевыми словами для фонда документов. Следовательно, можно выделить два различных обстоятельства, связанных с системой, индексирующей документы посредством ключевых слов: во-первых, «абонент» должен более внимательно относиться к составлению своего запроса на документ, поскольку термины, которые он при этом использует, будут служить инструментом при определении того, какие из документов фонда выбирать, а какие пропускать;

и, во-вторых, процедура поиска больше не представляется в виде простого сравнения одного термина (название класса) с другим, а при ее выполнении необходимо принять решение о том, сколько из выбранных для данного документа ключевых слов должно совпасть с терминами, указанными в запросе, чтобы считать этот документ относящимся к делу, и какие ключевые слова, используемые для выражения запроса, должны обязательно содержаться в документе, чтобы имело смысл предлагать его пользователю.

Таким образом, мы имеем дело с системой, которая значительно менее жестка, чем классификационная система, но в то же время предъявляет гораздо более строгие требования к формулировке запроса, если желательно воспользоваться преимуществом, обусловливаемым повышенной гибкостью этой системы.

Есть еще одна дополнительная проблема. Речь идет о том, что необходимо как-то выбирать термины, относящиеся к документам фонда. Эти термины могут быть конкретными и общими, их может быть много или мало. Поэтому для составителя указателя должны быть установлены некоторые руководящие принципы, и тем не менее очевидно, что большая книга, охватывающая целый ряд тем, требует больше ключевых слов, чем небольшая работа, посвященная одному конкретному предмету.

Необходимо, чтобы как составителю указателя, так и пользователю был дан некоторый критерий, позволяющий оценить выбранный набор терминов. Если при индексировании книги сотрудник библиотеки выберет, например, термин «дорога», а абонент, с другой стороны, сделает запрос на все книги, содержащие ключевое слово «улица», то очевидно, что при поиске не будет установлено нужного соответствия. Поэтому для того, чтобы устранить такое разногласие, обусловленное произвольностью выбора терминов как составителем указателя, так и абонентом, необходимо принять одну из двух стратегий: либо должен быть составлен список разрешенных терминов, либо нужно использовать тезаурус некоторого вида.

Если список разрешенных терминов составлен, то он показывает-ся как составителю указателя (а возможно, самому автору документа), так и пользователю. Затем они оба составляют описания документов, выраженные в разрешенных терминах, после чего пользователь может просто указать, как это отмечалось выше, сведения, которые, с его точки зрения, обеспечивают успех в поиске необходимой работы. Однако такой подход снижает гибкость системы (ради которой, собственно говоря, и разрабатывается система координатного индексирования) из-за ограничений, испытываемых составителем указателя. Как следствие этого опять появляется возможность ошибочного определения документа в тех случаях, которые кажутся не соответствующими предложенному списку. Конечно, можно просто расширить список разрешенных слов,

однако человек все равно ограничен теми терминами, которые разрешены в системе в момент установления с ней контакта.

Другой подход, как уже отмечалось, заключается в использовании некоторого тезауруса. Шарп [1] высказывает важное соображение о том, что тезаурус может использоваться для одной из двух совершенно различных целей. Одна может состоять в расширении форм запроса путем введения возможности последовательного поиска терминов, относящихся к исходному запросу, затем терминов, связанных с этими терминами, и т. д. Например, запрос всех документов по ключевому слову «улица» приведет к тому, что сначала будут найдены все документы, индексированные термином «улица», а затем на основе сведений, полученных из тезауруса, будут искаться все документы, индексированные терминами «дорога», «автострада», «колея» и т. д. Таким образом может быть исключена ситуация несовпадения терминов, указанных в запросе, с ключевыми словами.

Шарп считает неправильным такое использование тезауруса. С его точки зрения, на самом деле разумно использовать тезаурус для ограничения запроса путем отнесения терминов, предложенных пользователем в запросе, к какой-нибудь категории некоторого заданного набора терминов, которые затем используются исключительно на этапе сопоставления. Такой подход аналогичен фактически ограничению пользователя заданным списком терминов, исключая то обстоятельство, что выбор терминов делается не пользователем, справляющимся по списку, а сотрудником, ответственным за этап сопоставления, который находит термины, установленные составителем указателя и пользователем документов, в предварительно подготовленном тезаурусе.

Если используется один из двух способов, основанных на тезаурусе, то необходимо выполнить определенную работу по составлению тезауруса. Это оказывается очень важным, поскольку нельзя игнорировать ни один термин, который может быть выбран пользователем, если тезаурус предназначается для того, чтобы сократить принятие индивидуальных решений о соответствии терминов, или для выполнения процедуры сопоставления.

В общем, по-видимому, чем шире возможности, предоставляемые составителю указателя или абоненту при выборе своего собственного описания документа, тем большего объема предварительную редакторскую работу необходимо выполнять и тем труднее процедуры сопоставления выбранных описаний. Для того чтобы система работала нужным образом, описания должны быть настолько правильными и точными, что это очень часто на практике трудно достижимо и даже превышает требования классификационной системы. Тем не менее, выбирая достаточно терминов и делая это внимательно, пользователь может обеспечить большую вероятность отыскания нужных документов.

### 7.3. АВТОМАТИЧЕСКОЕ СОПОСТАВЛЕНИЕ

За исключением случая небольшой технической библиотеки, обычно реализация метода координатного индексирования без использования вычислительной машины сопряжена с проблемой обработки слишком большого количества документации. Изобилие повторений, возникающих при индексировании одного и того же документа по различным ключевым словам, приводит к тому, что некоторая форма автоматизации становится почти обязательной. Выше уже рассматривался элементарный метод, основанный на использовании пробивок в картах, однако с появлением машин стал возможным более совершенный метод, поскольку даже большие файлы могут быть просмотрены машиной весьма быстро при условии, что эти файлы размещаются на устройствах с произвольной выборкой данных. Здесь мы рассмотрим, как реализуется процедура сопоставления запроса и описания документа с помощью машины, т. е. как автоматизируется третий из выделенных выше трех этапов в ситуации информационного поиска. Для этого организуется машинный файл, состоящий из записей для каждого используемого при индексировании информации названия класса или ключевого слова, причем за каждой такой записью следует список, в котором перечислены все относящиеся к ней документы. Так, например, в файле может быть запись для ключевого слова ДОРОГА, сопровождаемая списком регистрационных номеров документов. Всякий раз, когда поступает на рассмотрение запрос на поиск документа, этот файл просматривается и из него выбираются номера документов, упомянутых в списке, относящемся к указанному в запросе названию класса, ключевому слову или комбинации ключевых слов. Затем будет исследован главный файл, содержащий сведения о всех доступных документах библиотеки, которые индексированы в данном файле номерами, присвоенными им при поступлении в библиотеку. Эти номера сопоставляются с номерами, выбранными из указанного выше списка, и все сведения о документах, соответствующих сделанному запросу, печатаются или воспроизводятся на видеозкране для пользователя.

Таким образом, машина должна просто выполнить две последовательности сравнений, чтобы найти документы, интересующие пользователя. Соответствующая программа оперирует с предварительно составленными описаниями (ключевыми словами или их комбинациями) документов из библиотеки и информацией, указанной в запросе. Классификационная система очень прямолинейна, система же ключевых слов более гибка, и с тех пор, как оказалось возможным возложить на машину дополнительную работу, связанную с использованием этой системы, интерес к ней возрос. Используя ключевые слова, можно более точно сформулировать запрос и, следовательно, исключить выборку массы не относящихся к делу

документов, обеспечив отбор небольшого числа необходимых. Очевидно, что такая система требует особой тщательности при составлении описаний документов и запросов, причем эта обязанность ложится на плечи сотрудников библиотеки.

Вся предварительная работа по определению того, какие описания или названия будут использоваться, и распределению документов по соответствующим классам (ключевым словам) выполняется сотрудником библиотеки, который использует машину для хранения файлов каталога и файлов индексов. Кроме файлов ключевых слов, возможно, будут составляться из новой поступающей информации файлы авторов, названий книг, издательств, так что при поиске документа в зависимости от того, насколько подробно составлен запрос, могут использоваться несколько различных файлов. Любой из этих файлов может быть просмотрен с целью составления подробных сведений о документах, номера которых были получены из главного файла.

Следовательно, рабочая нагрузка на персонал оказывается высокой, но процесс поиска документов автоматизируется. Такая система может быть весьма просто расширена для обеспечения избирательного распространения информации (служба ИРИ). Подобно тому как специфические запросы содержат списки ключевых слов, определяющих интересы пользователей, так и в машине могут храниться на некотором постоянном носителе данных файлы, в которых указываются области научных интересов ряда привилегированных пользователей, требующих подробные сведения о всяком новом поступлении, относящемся к этим областям. Это может быть реализовано в форме фоновой работы, заключающейся в том, что все новые поступления исследуются и сопоставляются с «пользовательскими научными профилями», но полученные результаты не воспроизводятся тотчас же на терминале или устройстве отображения данных, а сохраняются до конца некоторого установленного в данной системе периода (например, недели, месяца), после чего передаются заинтересованным группам пользователей. Эта работа может выполняться все время и прерываться более срочными работами, протекающими в режиме реального времени, например регистрацией новых поступлений или поиском документов по специальному запросу. Такой подход обеспечивает не только дополнительный сервис без увеличения нагрузки на библиотечный персонал, выдача ответов на некоторые возможные вопросы, прежде чем они были заданы, уменьшает количество дополнительных запросов на информацию. Более того, поскольку опять-таки речь идет о простом сопоставлении одной последовательности терминов с другой, одна и та же программа может использоваться для выполнения непосредственного поиска, а также просмотров, связанных со службой ИРИ, и таким образом достигается экономия программных средств и совместимость системы.

Поскольку сотрудник библиотеки не в состоянии помнить интересы каждого пользователя, то даже ограниченная по своим возможностям служба ИРИ является усовершенствованием неавтоматизированной системы, и информация при этом может передаваться быстро, регулярно и достоверно. Еще одно преимущество заключается в том, что есть готовый канал для установления обратной связи с пользователем. Это обеспечивает возможность контролировать и усовершенствовать систему. Пользователи могут делать замечания относительно материалов, полученных ими, и это будет весьма полезно при модификации тех методик, которые использованы при выборе и распределении ключевых слов. Система службы ИРИ была впервые предложена Лахном [11] и разработана в Техническом информационном центре фирмы IBM. При этом автоматически был подготовлен и затем разослан пользователям ряд анкет с просьбой заполнить их и вернуть. Сведения, указанные в анкетах, после возвращения были введены в вычислительную систему, которая исследовала их и вычислила статистические характеристики для системы.

Как при избирательном распространении информации, так и при обычном поиске могут использоваться одни и те же машинные методы обработки данных, поскольку в обоих случаях речь идет о выполнении некой процедуры сопоставления. В большинстве реализованных систем такого рода применяется метод, основанный на булевой алгебре. Он предусматривает возможность соединения нескольких терминов посредством логических операций И, ИЛИ, И НЕ. Как правило, при этом разрабатывается простой поисковый язык, такой, что при составлении запроса фактически можно напечатать что-нибудь подобное следующей записи:

**НАЙТИ ДОКУМЕНТЫ СОДЕРЖАЩИЕ «ДОРОГА» И «ТРАНСПОРТ» ИЛИ «ПЕШЕХОД» И НЕ «ВЕЛОСИПЕД».**

После обработки подобного рода запроса был бы напечатан список тех документов, которые в системе индексируются терминами «дорога» и либо «транспорт», либо «пешеход», но не «велосипед». Закрывая термины в круглые скобки, можно использовать эти простые выражения для составления весьма сложных запросов, причем научиться этому не так уж трудно.

В процессе обработки запроса будет обнаружено слово НАЙТИ, которое понимается как команда обращения к программе поиска. Последняя в свою очередь исследует строку входных данных, представляющую собой некое булево выражение, и выделит слова, использованные между операциями. Затем программа поиска просмотрит файл с целью получения информации, соответствующей указанной в запросе комбинации терминов.

Вместе с тем возможно, что даже если описывающие термины подобраны тщательно и в большом количестве, пользователь подобной системы все равно получит слишком много документов,

поскольку он не может составить свой запрос настолько специфично, чтобы исключить ненужные ему материалы, не рискуя при этом недополучить сведения о некоторых действительно заслуживающих внимания документах. С целью решения такой проблемы в некоторые приложения рассматриваемого метода внесено дополнительное усовершенствование. Оно заключается в том, что для каждого документа еще на этапе индексирования устанавливается некоторый весовой коэффициент, указывающий, насколько удачно данное ключевое слово (или комбинация ключевых слов) описывает этот документ. Такой коэффициент может быть, например, числом из диапазона от 1 до 10. Так, например, работа, посвященная вопросам транспортной перегруженности дорог и ссылающаяся также на статистику улично-дорожных происшествий с пешеходами, может быть индексирована ТРАНСПОРТ 9, МАШИНЫ 6, ПЕШЕХОДЫ 3, и таким образом указывается относительная значимость ключевых слов с точки зрения составителя указателя.

Аналогичные весовые коэффициенты могут использоваться для ключевых слов при составлении запроса. Если при этом, например, запрос содержит термин «транспорт», дополненный высоким коэффициентом, то весьма вероятно, что рассмотренный выше документ будет найден, а если в запросе указан только термин «пешеход» с небольшим весовым коэффициентом, то документ, вероятно, будет оставлен без внимания.

Другие модификации, вносящие элемент избирательности в процедуру выборки документов при поиске, предусматривают вычисление отношения числа совпавших терминов к общему числу терминов, указанных в запросе или сводке, характеризующей интересы пользователя, или же вычисление аналогичного отношения между числом совпавших терминов и общим количеством ключевых слов в описании данного документа (последний критерий лучше отражает соответствие документа запросу). Однако оба этих метода менее удачны, чем использование весовых коэффициентов.

Хотя метод, основанный на применении весовых коэффициентов, и обеспечивает исключение большего числа не относящихся к делу документов, чем это достигается при использовании комбинаций, состоящих только из ключевых слов и булевых операций, тем не менее он сопряжен с несколькими затруднениями. Одно из них заключается в том, что как файл ключевых слов, так и запросы должны быть составлены с исключительной тщательностью. Далее, дополнительная работа по анализу документов (с целью установления соответствующих весовых коэффициентов) недопустима в условиях большой библиотеки. Кроме того, пользователь должен еще указывать некоторые целочисленные оценки своей заинтересованности в документах, причем система отбирает те документы, для которых эти оценки достигаются. Конечно, вероятность того, что в результате ошибочного определения одного ключевого слова будет

утрачена возможность отыскания документа, мала, однако у пользователя должен быть значительный опыт работы с системой, для того чтобы он мог достаточно точно указывать весовые коэффициенты в запросе.

Следующий вопрос, связанный с использованием такого рода системы: каким образом можно оценить ее преимущества и недостатки? При проведении испытаний можно установить, насколько правильно работает система. При этом, например, могут быть собраны две группы статистических данных. Одну составляют относительные коэффициенты точности, где

коэффициент точности = (общее число соответствующих документов)/(общее число фактически найденных документов).

В другую группу входят коэффициенты отказа, где коэффициент отказа = (число ненайденных соответствующих документов)/[(число ненайденных соответствующих документов) + (общее число найденных документов)].

Такие статистические данные могут быть полезны при оценке производительности некой моделирующей системы, однако в случае реально работающего варианта библиотечной системы такие показатели не могут быть получены, поскольку нет способа регистрации ненайденных соответствующих документов.

В заключение следует сказать, что метод автоматического сопоставления обеспечивает (правда, за счет значительной дополнительной работы, возлагаемой на библиотечный персонал) быстрое действие, совместимость, привлекательность системы, а также дополнительный сервис для пользователя. При этом основное значение имеет принятый в данной системе способ индексирования документов, который может быть классификационной схемой, но более вероятно, что он будет основан на координатном индексировании, поскольку в этом случае можно более полно воспользоваться преимуществами машинной обработки данных. Производительность системы связана с аккуратностью и осведомленностью, проявленными при индексировании документов и составлении запросов, и ее трудно оценить. Это объясняется тем, что машина сопоставляет две независимо составленные последовательности терминов, причем она никак не контролирует процесс составления какой-либо из этих последовательностей. В следующем разделе мы рассмотрим, каким образом можно исправить этот недостаток.

#### 7.4. АВТОМАТИЧЕСКАЯ КАТАЛОГИЗАЦИЯ

До сих пор мы обсуждали вопросы автоматизации последнего из трех различных этапов обработки информации, которые имеют место при информационном поиске в условиях библиотеки. Ставя вопрос об автоматической каталогизации, мы тем самым вплотную

подходим к вопросам автоматизации второго этапа, включающего исследование вновь поступивших документов и индексирование их согласно принятой системе правил. Если удастся установить критерии, которые используются при распределении конкретных документов по определенным в данной системе классам или при индексировании конкретных документов определенными ключевыми словами, то, вероятно, можно будет возложить выполнение второго этапа тоже на машину.

При этом машина будет определять, какое описание документа из доступного ей набора описаний должно быть отнесено к данному конкретному документу. Общая структура системы классификации документов все же выбирается и задается сотрудником библиотеки, тем не менее машина подгоняет отдельные документы под эту структуру. Такой шаг, если он сделан, знаменует переход через исключительно важную границу. Действительно, до сих пор машина имела дело с описаниями документов, составленными после анализа их человеком. Если же анализ будет выполняться самой машиной, то документ или некоторая соответствующая часть его должны быть доступны машине.

Один из возможных подходов состоит в использовании названия документа в качестве базы для автоматического индексирования. Однако было бы слишком опрометчиво считать, что автор исчерпывающим образом охарактеризует свою работу в названии или что из названия в любом случае может быть получена информация, достаточная для того, чтобы установить различия в содержании документов. Другое предположение предусматривает использование при автоматическом индексировании некоего вида резюме работы, подобного тому, что часто требуется при публикации работы в периодическом издании. Такое резюме может быть подготовлено автором наряду с работой или же, например, может состоять из первого и последнего параграфов документа. В последнем случае выбранные из работы фрагменты и, стало быть, само резюме могут оказаться слишком произвольными; что же касается составляемых авторами резюме, то они уже используются в некоторых экспериментальных системах.

Все же очевидно, что наиболее успешное решение обсуждаемого вопроса может быть достигнуто путем ввода в машину всего поступившего в библиотеку документа. В большом масштабе это практически не осуществимо, пока либо не появится надежное оборудование (которое потребует введения единого формата и печатного шрифта) для оптического чтения символьной информации, либо издатели не получат дешевую и простую возможность для представления опубликованных работ в форме, непосредственно пригодной для машинного чтения.

Если временно допустить, что весь документ или часть его, представленная в виде резюме, могут быть считаны в машину, то

остается рассмотреть критерий, который можно использовать при подборе классов и ключевых слов. Обычный способ такого подбора предусматривает исследование словарного состава самого документа или названий его разделов, когда речь идет о большой книге. Если в данной библиотечной системе установлена некоторая иерархия классов, то машине сообщается список этих классов вместе с типичными примерами слов, связанных с тем или иным классом. Машинная программа будет просматривать документ, выделяя в нем слова, встречающиеся в указанном выше списке, и подсчитывая общее количество таких вхождений в документ для каждого слова. Необходимо отметить, что в исходном списке не должны встречаться слова, которые не могут служить средством для распознавания классов. После завершения просмотра документа программа, используя полученные статистические данные, вычисляет некие значения, являющиеся псевдоиндексами документа или вероятностями того, что данный документ принадлежит тому или иному классу иерархии. Затем после анализа этих вероятностей рассмотренный документ приписывается к тому классу, в котором его псевдоиндекс имеет наибольшее значение. Поиск выполняется, как и прежде, просто путем отыскания всех документов, приписанных к данному классу; при этом может использоваться также список документов так называемой второй очереди, т. е. документов, для которых данный класс по значениям псевдоиндексов был определен как следующий наиболее вероятный за выбранным. Экспериментальные исследования в этом направлении, проведенные Мароном [25, 26], принесли различные результаты, свидетельствующие об успехе методики в 50—80% случаев, причем результаты оказались лучше тогда, когда учитывалась не только частота слов в документе, но и их расположение.

За исключением того, что процедура определения ключевых слов документа оказывается несколько более сложной, чем рассмотренная выше, в остальном при использовании схемы координатного индексирования вопрос автоматизации второго этапа решается аналогичным образом. Необходимо отметить, что трудно привести веские основания, которые оправдывали бы применение классификационной схемы в автоматизированной библиотечной системе, не считая того, что классы используются при выполнении информационного поиска. Поскольку ключевые слова обеспечивают более точные средства и для проведения поиска, то схема координатного индексирования, вероятно, все же более привлекательна для автоматизированной библиотечной системы, чем классификационная схема. Для того чтобы определить ключевые слова, машине нужен не только сам документ, но также и список терминов, которые она может использовать в качестве описателей.

Поскольку нет никакой гарантии, что автор ограничится только теми терминами, что указаны в этом списке, должен также быть

предусмотрен тезаурус, который машина может просматривать с тем, чтобы связать термины, которые, как устанавливается при исследовании документа, используются часто, и термины из принятого списка описателей. Подобное ограничение относится и к запросу пользователя на поиск документа. При этом либо пользователь должен придерживаться списка стандартных терминов при составлении запроса, либо он может быть свободен в выборе терминов, но тогда машина должна реализовывать какую-то процедуру словарного поиска. Таким образом, слова, используемые как в документе, так и в формулярах запросов, «декодируются» в термины, употребляемые внутри данной автоматизированной библиотечной системы. Процедура поиска, как и прежде, сводится к просмотру и сопоставлению комбинаций ключевых слов, однако в данном случае машина выполняет значительную часть работы, ранее возложенной на библиотечный персонал. Действительно, теперь библиотечный персонал должен составить словарь или тезаурус, а после этого единственное, что от него требуется, так это изменять словарь или тезаурус по мере накопления опыта работы с системой. Осложнение, связанное с частым использованием автором какого-нибудь особенно непонятного термина (возможно, впервые предложенного им самим), может быть частично разрешено, так же как и многие другие возможные затруднения в автоматизированной библиотечной системе, путем применения разговорной системы, работающей в режиме реального времени. При использовании такого аппарата библиотечная система, оказавшись в тупиковой ситуации при рассмотрении конкретного документа, может попросить помощи, напечатав сообщение сотруднику библиотеки. Если случится такая неприятность, то сотрудник библиотеки может отменить выполнение машинной процедуры и сам приписать документу соответствующие описатели. Таким образом, обработка документа системой не будет прекращаться только из-за того, что некое частное решение не было заранее запрограммировано в системе. Такого рода возможность является наиболее важной характерной чертой системы, работающей в режиме реального времени. При условии что в первоначальной системной программе предусмотрены подпрограммы обслуживания каналов, через которые осуществляется связь с библиотечным персоналом, отмеченная выше возможность реализуется обычными программными средствами.

Еще одна проблема связана с тем, что при описании документа может потребоваться как раз редко используемое слово. И если в системе предусмотрен словарь только часто употребляемых слов, то документ будет описан ошибочно. Этот недостаток предлагаемого критерия автоматического индексирования обсуждается в заключительной главе. Сейчас же мы можем перейти к рассмотрению следующего логического звена в цепочке процедур информационного поиска.

### 7.5. АВТОМАТИЧЕСКАЯ КЛАССИФИКАЦИЯ

Уже было отмечено, что даже если в библиотечной системе осуществляется автоматическая индексация документов с помощью машины, то все равно библиотечный персонал несет еще большое бремя предварительной редакторской работы. Эта работа начинается с этапа выбора классовой структуры или индексной структуры, в пределах которой можно автоматизировать выполнение процедур информационного поиска, и продолжается в виде составления и изменения списков терминов и тезауруса с целью повышения эффективности, точности и гибкости системы. При надлежащей автоматической классификации эта работа исключается из обязанностей библиотечного персонала и выполняется машиной.

Еще до появления вычислительных машин был предложен так называемый метод аспектного анализа, согласно которому фонд документов исследуется с целью определения соответствующего набора ключевых слов, так же как это делается в случае обычного координатного индексирования. Затем выявленные ключевые слова распределяются по группам согласно тематическим аспектам или родственным связям между ними. Сформированные таким образом группы становятся базой для классификационной системы. Причем в этой системе нахождение любого элемента из фонда ведет к отысканию всех других элементов, связанных с той же самой группой ключевых слов.

Этот метод ликвидирует разрыв между классификационным и координатным способом индексирования документов и дает возможность составить некий вариант динамической классификационной системы посредством формирования классов из групп терминов, взятых непосредственно из документов. Таким образом, эта система является эмпирической, а не теоретической. Приписка любого документа происходит по отношению не к какому-то набору заранее определенных классов, а к набору классов, формируемому согласно некоторым документам, уже имеющимся в системе. Итак, необходимо только иметь классы для материалов, содержащихся в фонде в данный момент, и классовую структуру можно пересматривать с учетом новых поступлений в библиотеку.

Если попытаться реализовать этот метод вручную, то потребуется выполнить огромную работу, связанную с анализом документов, однако с помощью машины в этом направлении можно достигнуть хороших результатов. Таким образом, следующим логическим шагом является автоматизация первого этапа информационного поиска, т. е. этапа формулирования индексирующей структуры системы. Это достигается путем автоматического составления групп документов на основе результатов машинного анализа содержания этих документов.

Несколько попыток, связанных с автоматическим распределе-

нием документов по группам, сделали в Англии Нидхэм и Паркер-Родс [31], а в США Борко и Берник [27, 28]. Для анализа словарного состава документов могут быть выбраны различные математические и статистические способы, однако подробности этих способов относятся к области численной систематики. Эти способы используются для объединения в группы тематически одинаковых документов, а последующие библиотечные поступления уже распределяются по имеющимся в данный момент группам с учетом терминов, содержащихся в них. Как и при аспектном анализе, в данном случае поиск по ключевому слову приводит к нахождению всей группы документов с одинаковой тематикой.

Весьма похожий метод состоит в вычислении коэффициентов связи между документами. При использовании этого метода, строго говоря, документы на самом деле не классифицируются. Вместо этого на основе анализа сочетания появившихся терминов вычисляется связь между всяким новым документом и имеющимися в библиотеке документами. Таким образом, каждый элемент информации оказывается связанным с рядом других элементов посредством коэффициента связи, что позволяет организовать поиск. При этом все документы, отношения между которыми характеризуются высокими коэффициентами связи, как правило, будут отыскиваться вместе. Из терминов, которые часто встречаются в сочетании с другими терминами, может быть составлен вариант внутреннего тезауруса, и он будет учитываться при отборе документов. Другими словами, процедурой анализа составляется некий вариант плана библиотеки, в котором зафиксированы все документы и связи между ними. Этот план хранится внутри системы в виде машинного файла.

В любом из этих методов все же присутствует элемент человеческой деятельности на уровне принятия решения о том, сколько терминов, например, следует использовать при составлении индекса или определении класса. Однако можно составить программу, которая вводила бы эти показатели в качестве параметров системы, а затем с помощью ряда экспериментов можно установить наилучшие значения параметров. Естественно, что при программировании крупномасштабной автоматизированной библиотечной системы потребуются весьма и весьма значительные усилия для успешного решения такой проблемы.

Результаты экспериментов, связанных с распределением документов по группам, различны в зависимости от содержания самих документов и тех конкретных способов, которые при этом использовались. Тем не менее Дойял [29] предпринял попытку разработать систему контрольных тестов, которые можно было бы применять для определения эффективности любого из этих методов. Он предлагает при проведении испытаний какого-нибудь из рассмотренных методов использовать в качестве различных элементов фонда фрагменты одного и того же документа, чтобы убедиться, действительно

ли эти «элементы» группируются вместе. Он также использует при определении объективной ценности рассматриваемого метода в качестве тестовых данных работы одного и того же автора и издания хронологического характера, подобные календарям. Полученные им результаты показывают, что чем больше и больше терминов используется при объединении документов в группы, тем успешнее выполняется классификация; при этом почти или совсем не было тенденции к сходимости и элементы передвигались из класса в класс, по-видимому, случайно. По мере того как изменялось количество используемых терминов, претерпевала изменения и классификационная схема. При эксперименте такого рода, по-видимому, можно установить пределы эффективности разработанных в настоящее время методов объединения документов в группы, если при этом основываться исключительно на ситуации информационного поиска.

Подобные методы характеризуются следующими основными моментами. Весь документ или по крайней мере резюме документа используется системой, и таким образом устраняется необходимость проведения человеком обширного контекстного анализа. За качество классификации отдельных документов и за применяемую в системе классификационную схему отвечают машина и ее программы. Классификационная схема по своей природе оказывается эмпирической и составляется с учетом содержания имеющихся в фонде документов, а не на основании предварительно сложившихся мнений о том, какая классовая иерархия более выгодна. Специальные методы применяются также для того, чтобы «определить» степень связности отдельных документов друг с другом. Это обеспечивается либо путем статистической оценки связности, либо путем составления квазипространственного аналогового плана связей между документами [30].

Можно высказать несколько соображений, касающихся расширения критерия, используемого при группировании и классификации документов с помощью машины. Во-первых, в частном случае, когда дело касается научных публикаций, для объединения элементов в группы можно воспользоваться библиографическими ссылками. Результаты проведенных Айви [32] экспериментов свидетельствуют, что в 64—90% случаев поиск элементов, сгруппированных согласно ссылкам, выполнялся успешно, причем хуже всего дело обстоит тогда, когда рассматриваемый документ содержит менее трех ссылок.

Другое соображение связано с использованием избирательного распространителя информации. Дело в том, что при реализации процедуры поиска в ней можно учесть те мнения и замечания относительно документов, которые могут быть высказаны в результате выборочного опроса среди пользователей, получающих документы через службу ИРИ. Такой подход предпочтительнее, чем расшифровка заметок, сделанных каракулями на полях документа, и при этом

заключение эксперта, а также накопленный опыт могут быть в полной мере использованы на благо системы и обычного пользователя.

Очевидно, что полная оперативная библиотечная система должна быть достаточно гибкой, чтобы допускать использование ряда различных методов и средств. Можно представить себе такую ситуацию, когда в библиотеке установлен ряд терминалов и экранных устройств, часть из которых используется библиотечным персоналом, а другие доступны обычным абонентам. При этом на терминалах будут предусмотрены клавиатуры для установления связи с системой, а ведущая системная программа может спрашивать у пользователя, что ему необходимо, передавая затем управление для обработки запроса соответствующей подпрограмме. Вполне возможен такой случай, что сотрудник библиотеки дополняет файлы, работая за одним выносным пультом, а в то же самое время пользователь передает запрос системе с другого терминала.

В файлах системы для каждого пользователя можно хранить подробные сведения о всех документах, к которым он обращался за справкой, и тогда система способна удовлетворять запросы, сформулированные даже таким образом, как: «книга, которая была у меня в прошлый четверг» или «большая красная книга, которую я смотрел на прошлой неделе». Если сформулирован запрос общего характера, то система потребует сообщить ей набор терминов, отражающих интересы пользователя, или же, если речь идет об обслуживании постоянного абонента, она может обратиться за справкой к ранее составленному файлу, содержащему информацию об интересах пользователей. Если система не осведомлена об интересах данного пользователя, то она может задать ему более конкретный вопрос или воспроизвести для него на устройстве отображения некоторый фрагмент тезауруса. Так, например, если пользователь обратился к системе с термином «шар» в качестве ключевого слова, то ему может быть показана некая «карта», содержащая термины «стереометрия», «астрономия» и «бильярд», с просьбой выбрать один из них.

После того как интересы пользователя будут в достаточной степени определены, управляющая программа системы обратится к подпрограмме, осуществляющей поиск в файлах всех уместных в данном случае документов.

В зависимости от принятого для доступа к документам метода в результате поиска могут быть найдены либо все документы, содержащие в достаточном количестве заданные ключевые слова, либо наборы документов, коэффициенты связности которых друг с другом достаточно высоки, либо все документы, принадлежащие к какой-то конкретной группе. Развитая автоматизированная библиотечная система может, по-видимому, выполнять поиск тремя различными наборами процедур, предоставляя тем самым три различных комплекта документов, найденных с помощью разных

методов. Причем все документы, указанные в этих трех списках, конечно, будут именно теми, что следует просмотреть в первую очередь. Разработанная в США система SMART основывается на том принципе, что использование нескольких методов, вероятно, более эффективно, чем какого-то одного [33, 34].

Изучив список (или списки) предложенных ему документов, пользователь может указать те названия, которые его интересуют, а система после этого либо сообщит ему все подробности, касающиеся этих документов, и справки о том, где они находятся, либо воспроизведет на экранном устройстве резюме документа или выдержку из него. Ознакомившись с такого рода информацией, пользователь может убедиться, действительно ли это те документы, которые ему нужны. При возникновении каких-то сложных обстоятельств пользователю автоматически могут быть переданы копия документа или микрофильм. В тех случаях, когда пользователь столкнется с трудностями, ему окажут помощь либо специальные системные программы, либо сотрудники библиотеки, хорошо знакомые с системой.

Некоторый критический разбор философских принципов рассмотренных методов проводится в последней главе данной работы наряду с обсуждением последнего возможного логического шага на пути автоматизации процесса библиотечной деятельности, а именно замены материальных документов информацией, хранящейся непосредственно в машине.

Тем не менее отметим, что общая картина вполне работоспособной библиотечной системы, работающей в режиме реального времени, может быть составлена с учетом использованных нами ранее терминов и с учетом машины, которая помогает, насколько это возможно, пользователю в поиске, спрашивает его, когда необходимо прояснить ситуацию, и консультирует сотрудника библиотеки в случаях возникновения различных затруднений и двусмысленностей. Подобно тому как разделы тезауруса или плана связей документов воспроизводятся на экране при выполнении специфических запросов на поиск, точно так же и при беспорядочном переборе документов из фонда они могут оказаться весьма полезными для абонента, однако, как уже отмечалось выше, беспорядочный перебор, выполняемый с удаленного терминала, фактически не является заменой обычного просмотра документов на полках.

Насколько такая система будет эффективна и удобна для пользователя, зависит от «разумности» и гибкости ее программ. Если при общении с системой пользователь вынужден преодолевать невидимый барьер, воздвигаемый «необычно глупой» машиной, то такая система никогда не завоеует доверия. В следующей главе обсуждаются некоторые специальные предложения и примеры, касающиеся программирования задачи общения человека с машиной.

## 8

### НЕКОТОРЫЕ ПРИМЕРЫ ПРОГРАММНОЙ РЕАЛИЗАЦИИ

---

#### 8.1. ВВЕДЕНИЕ

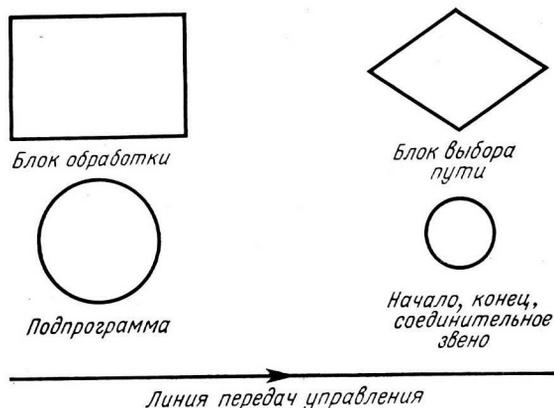
В этой главе приведено несколько примеров программ, иллюстрирующих способы решения некоторых проблем, выявленных при обсуждении принципов автоматизированной библиотечной системы. В частности, эти примеры показывают, как можно устранить ту очень большую дополнительную работу, которую приходится выполнять библиотечному персоналу при подготовке регистрационных материалов, а также как учитывать ошибки, двусмысленности и неточности, появляющиеся в информации или запросах, передаваемых системе. Программы не претендуют ни на какое конкретное назначение, а служат лишь иллюстрацией рассмотренного ранее материала и написаны для выполнения на малой машине, не обладающей возможностями произвольного доступа к информации и работы в реальном режиме времени. Допускаются только небольшие примеры файлов документов, и это сделано с той целью, чтобы при знакомстве с этими программами можно было весьма подробно рассмотреть многие из затруднений, связанных с обработкой файлов, и их возможные решения. Конечно, большие файлы привносят свои специфические проблемы, но следует иметь в виду, что большинство разработчиков вычислительных систем, как правило, включают в комплект программного обеспечения пакет программ управления файлами, в том числе и большими файлами.

В связи с этим в данной главе описываются простые наглядные примеры, и наша цель состоит в том, чтобы отработать такой подход, который пригоден для разработки крупномасштабной системы. После того как процедура запрограммирована, ее можно использовать 50 000 раз так же легко, как и пять раз. Приводимые здесь программы иллюстрируют, как может быть реализована элементарная разговорная система, осуществляющая хранение библиотечного каталога в машинных файлах и нахождение информации по запросам пользователей.

Программы описываются с помощью блок-схем, т. е. стандартных диаграммных изображений потока управления в программе. Каждый самостоятельный этап программы изображается на диаграмме в виде отдельного блока (геометрической фигуры), а пути выполнения программы указываются стрелками, соединяющими

блоки. В тех случаях, когда в программе должно быть принято решение о том, по какому из нескольких возможных путей она будет выполняться дальше, это решение изображается на блок-схеме в виде ромба, связанного с несколькими соединительными звеньями. Условные обозначения, принятые в блок-схемах, изображены на рис. 6.

Обсуждаемые здесь программы написаны для вычислительной машины марки IBM 1620, в оперативной памяти которой можно разместить 20 000 десятичных цифр или 10 000 буквенно-цифровых



Р и с. 6. Обозначения, принятые в блок-схемах.

символов, поскольку каждый символ занимает столько же места, как две цифры. Единственными средствами ввода-вывода являются пишущая машинка и оборудование для пробивки и чтения информации с бумажной перфоленды. При обращении к устройству чтения возможен ввод только с одной перфоленды, и какие бы то ни было операции с файлом, размещенным на перфоленде, разрешаются только после завершения процесса ввода всего этого файла. Поэтому файл ограничен в объеме. Таким образом, программы, к рассмотрению которых мы переходим, никоим образом не являются неким желаемым вариантом автоматизированной библиотечной системы. Тем не менее они моделируют функционирование системы, работающей в режиме реального времени, и служат иллюстрацией методов, с помощью которых решаются некоторые проблемы, обусловленные взаимодействием между пользователем, который «всего лишь» человек и часто ошибается, и машиной, «особой», весьма пунктуальной и требующей пунктуальности со стороны пользователя.

## 8.2. ПРОГРАММА РЕДАКТИРОВАНИЕ

При написании программ преследовалась цель создания небольшой вспомогательной системы, которая снабдила бы пользователя средствами обслуживания и управления при работе с библиотечной системой. Небольшая, постоянно находящаяся в оперативной памяти машины управляющая программа, называемая монитором, спрашивает пользователя, какие средства обслуживания ему необходимы. В системе предусмотрены следующие программы обслуживания: программа РЕДАКТИРОВАНИЕ, которая принимает и заносит в файл новые дополнения к фонду документов; программа СОРТИРОВКА, обеспечивающая сортировку информации, содержащейся во всех библиотечных файлах; и, наконец, третья программа, которая обновляет файл, содержащий регистрационные номера документов и упорядоченный по фамилиям авторов, а также отыскивает в этом файле элементы информации согласно запросам пользователя.

Файл документов был произвольным образом выбран как набор статей из каких-то пяти философских журналов. Задача обработки последовательности объектов, подобных этим статьям, может быть выполнена машиной, однако на практике, как уже отмечалось выше, дело осложняется необходимостью значительного предварительного редактирования, связанного с использованием специальных символов в качестве разделителей полей. Таким образом, возникает вопрос, как возложить на машину редакторскую работу, не увеличив при этом степень риска, связанного с появлением ошибок в основном каталожном файле.

Есть два пути преодоления такого рода трудности. Первый связан с использованием оборудования, работающего в режиме реального времени, и предусматривает управление всем процессом каталогизации документов с помощью специальной программы, которая печатает или воспроизводит на экранном устройстве, например, сначала слово АВТОР и ожидает ответа. Получив ответ, она печатает слово НАЗВАНИЕ, снова ждет ответ и т. д. Прежде чем будет нажата клавиша «ИСПОЛНЕНИЕ», пользователь, находящийся за пультом, может исправить любую ошибку в составленном им сообщении. После нажатия клавиши «ИСПОЛНЕНИЕ» рассматриваемая программа обрабатывает любой вводимый материал как ответ на сделанный ею информационный запрос, и при этом не возникает никаких трудностей в определении отдельных полей информации. Однако этот способ редактирования может оказаться утомительным, поскольку каждый раз допускается ввод только одного небольшого кванта информации, и, кроме того, он требует использования терминалов или экранных устройств.

Другой метод, иллюстрируемый ниже, дает возможность библиотечному персоналу вводить информацию обыкновенным способом,

но при этом уже машина определяет, где одно поле информации начинается и другое кончается. Это можно обеспечить и путем включения в информацию ограничителей, однако рассматриваемый здесь метод позволяет использовать более свободный формат представления информации, требуя только, чтобы элементы вводились в фиксированном порядке. Поскольку больше ничто не указывает программе, где начинается название работы и кончается фамилия автора, в нее должны быть включены разнообразные средства контроля для предотвращения ошибок. Преимущество данного метода заключается в том, что информация сводится в форме, хорошо знакомой библиотечному персоналу и свободной от докучливых ограничений, причем не представляет особого труда включение любых дополнительных контрольных признаков, требуемых пользователем.

Приводимый ниже формат был принят процедурой редактирования для ввода информации.

При составлении сообщения используются следующие основные элементы: цифра буква пробел, Ж и 18 19

@ используется в качестве терминального символа (признак конца записи).

Формат определяется так:

```

<строка> ::= буква | <строка> буква
<дата> ::= цифра цифра | 18 цифра цифра | 19 цифра цифра
<месяц> ::= <строка>
<журнал> ::= Ж цифра
<инициал> ::= буква
<автор> ::= <строка>, пробел <инициал> | <автор> пробел <инициал> |
          <автор> пробел и пробел <автор>
<название> ::= <строка> | <название> пробел <строка>
<элемент ввода> ::= <название> пробел <автор> пробел <журнал>
                  пробел <месяц> <дата> @

```

Рис. 7. Описание в бэкусовской нормальной форме принятого формата входных данных.

Для работы программ предоставляются перфоленты, содержащие следующую информацию в указанном порядке: название работы, автор, название журнала, месяц, год. С целью сокращения длины сообщения название журнала задавалось двухбуквенной аббревиатурой из диапазона от Ж1 до Ж5. Информация выбиралась произвольно только для иллюстрации и могла бы быть легко изменена. Дату можно вводить в виде, наиболее знакомом для оператора, при этом записи «Март 1968» и «Мар 68» одинаково допустимы. На рис. 7 приведено описание в бэкусовской нормальной форме используемого формата сообщений. В приложении А настоящей книги объясняется формальный метод описания форматов ввода, называемый бэкусовской нормальной формой. Для подготовки входных данных установлен определенный порядок печатания элементов ввода и требуется специальная отметка конца сообщения (КС).

Программа РЕДАКТИРОВАНИЕ просматривает входную строку, определяет и подробно исследует различные поля, присваивает

поступившей в библиотеку статье регистрационный номер и формирует на входной перфоленте основной файл новых поступлений, расположенных в порядке их регистрационных номеров. В крупномасштабной автоматизированной системе при выполнении таких операций, естественно, использовались бы устройства с произвольной выборкой информации.

В программе предусмотрены три уровня исправления ошибок:

1. Программа учитывает отклонения в формате: дополнительные пробелы, пунктуацию и т. д. Она также может корректировать ошибки в поле даты в случае неправильного написания месяца. Такие ошибки и отклонения автоматически исключаются из окончательной версии элемента ввода.

2. Программа может обнаружить наличие ошибки в элементе ввода, но не исправляет ее сразу. В этих случаях программа обращается за справкой к оператору, сообщая ему о возможной ошибке и печатая правильную версию поля. Если ошибка очень серьезная или не ограничивается одним полем, программа печатает оператору сообщение НЕДИАГНОСТИРУЕМАЯ ОШИБКА, после чего элемент ввода перепечатывается оператором.

3. Любые ошибки, которые не были обнаружены процедурой редактирования, могут быть устранены на заключительном этапе. Всякий раз, когда отредактированный элемент готов для вывода на основной файл, он печатается в полном виде на пульте оператора, и оператор имеет возможность исправить любую до сих пор необнаруженную ошибку в отредактированной версии. Затем элемент ввода будет перепечатан, и если новая версия удовлетворит оператора, то она будет выведена на основной файл.

Программа начинает просмотр входной строки с конца, определяя и представляя в расширенном виде сначала год, затем месяц, журнал и т. д. Самые различные проверки могут быть предусмотрены в такой процедуре редактирования. Например, в случае когда в поле даты указано число, меньшее чем 40 или большее чем текущий год, программа обратится с вопросом к оператору, однако полностью записанная дата 1890 будет принята без вопроса. Анализ поля заканчивается при появлении пробела, предшествующего этому полю. Ошибки в названии месяца исправляются автоматически, за исключением двусмысленных случаев, подобных записи *Март*, которая может означать *Март* или *Май*. Поле автора устанавливается по пробелам, разделяющим инициалы, и запятой, следующей за фамилией (например, *Смит, Д К*). Если программа подозревает, что в поле допущена ошибка, она может задать вопрос, например, *ЗАПЯТАЯ ПРОПУЩЕНА* и, получив ответ *ДА*, автоматически вставит запятую. При редактировании записи вида «*Смит, Д К Мл*» программа передает оператору сообщение-вопрос, поскольку предполагается, что каждый инициал задается одной буквой. Однако после того, как оператор укажет, что данная исход-

ная строка правильная, она будет принята программой. Если в названии статьи были пропущены или неправильно записаны слова, то такого рода ошибки могут быть исправлены на заключительном этапе, когда отредактированная версия входного элемента будет передана оператору. Ниже приводится пример работы с системой, типичный для случая обработки новых библиотечных поступлений, а в конце главы приведены блок-схемы обсуждаемых здесь программ.

На следующих страницах представлена запись «беседы» оператора с системой, иллюстрирующая работу программ, описанных в данной главе. Ответы системе, печатаемые на выносном пульте оператора, заканчиваются отметкой КС, позволяющей отличать их от сообщений, передаваемых системой. При этом иллюстрируются следующие характерные детали:

1. Программа МОНИТОР, находящаяся постоянно в оперативной памяти машины и выбирающая необходимые для данного пользователя средства обслуживания.

2. Поступающие в машину и отредактированные системой элементы библиотечной документации.

3. Поиск заполненных элементов документации в авторском файле и соответствующие сообщения о них пользователю. (Там, где это необходимо, запись «беседы» дополнена нашими комментариями.)

#### 1. Прием документов и редактирование.

В приведенных ниже примерах показана процедура приема нескольких входных элементов библиотечной документации, причем все эти элементы, кроме первого, содержат различные ошибки. Прежде чем перейти к фактической записи «беседы», следует ознакомиться со списком неправильных элементов, дополненным комментариями допущенных ошибок.

Следующие элементы содержат ошибки:

ОБ ОДНОМ НЕУМЕСТНОМ ОТОЖДЕСТВЛЕНИИ ЛУКАС Д Р Ж З  
ОКТ65@

(Пропущена запятая после фамилии.)

НЕ ВСЛЕДСТВИЕ, А ВОПРЕКИ ЛУКАС, Д Р Ж9 ОКТ66@

(Аббревиатура Ж9 недопустима, правильная ссылка на журнал Ж4.)

О ЛИБЕРАЛИЗМЕ МИЛЛА МАК-КЛОСКИ, Г Д Ж4 ЯНВ66@

(Неправильно указана дата, надо — янв. 63.)

ЕЩЕ РАЗ О ЛИБЕРАЛИЗМЕ МИЛЛА МАК-КЛОСКИ, Г Д Ж4 ЯНВ66@

(Пропущен пробел между инициалом и ссылкой на журнал.)

ИСТОЧНИК ОГОРЧЕНИЙ ТЕЙЛОР, ДМ ЯНВ69@

(Правильная дата — янв 65; пропущена ссылка на журнал; пропущен пробел между инициалами.)

О ПРАВАХ МАК-КЛОСКИ, Х Д Ж4 АПГ1965@

(Неправильно записана аббревиатура названия месяца — надо АПР; эта ошибка будет исправлена автоматически.)

ФИЛОСОФИЯ И ТЕОЛОГИЯ ФИЛИПС, Д З ЖЗ ОКТ63@

(Допущена ошибка в названии статьи.)

Ниже приводится запись фактического диалога между пользователем и системой.

(Перед началом диалога в машину вводится программа МОНИТОР, после чего пользователь, находящийся за выносным пультом, нажимает клавишу НАЧАЛО.)

ЧТО ВАМ НУЖНО: ПРИЕМ, СОРТИРОВКА, ДОБАВЛЕНИЕ ИЛИ ПОИСК?  
ПРИЕМ КС.

ПОДГОТОВЬТЕ ПЕРФОЛЕНТУ

(Оператор проверяет, установлена ли выводная перфоленка, на которой будет получен основной файл новых поступлений.)

ВВЕДИТЕ ПРОГРАММУ РЕДАКТИРОВАНИЕ, НАЖМИТЕ КЛАВИШУ ПУСК

(С перфоленты считывается программа РЕДАКТИРОВАНИЕ, а затем вводятся подготовленные на этой же перфоленке исходные данные о документах.)

001. ЛЮДИ И ДУХИ/КОВАЛ, С/ФИЛ. К./ЯНВАРЬ/1963

(Информация о первом документе подготовлена на перфоленке без ошибок. Этому документу присваивается регистрационный номер, а сведения о нем выдаются на перфоленку.)

ХОТИТЕ ЛИ ВЫ ИСПРАВИТЬ ЧТО-НИБУДЬ?

НЕТ КС.

(Перед тем как будет отперфорирована заполненная системой версия сведений о документе, система предоставляет возможность внести окончательные исправления.)

ОБ ОДНОМ НЕУМЕСТНОМ ОТОЖДЕСТВЛЕНИИ ЛУКАС Д Р  
ЗАПЯТАЯ ПРОПУЩЕНА?

ДА КС.

002. ОБ ОДНОМ НЕУМЕСТНОМ ОТОЖДЕСТВЛЕНИИ/ЛУКАС,  
Д Р/ФИЛОСОФИЯ/ОКТЯБРЬ/1965

ХОТИТЕ ЛИ ВЫ ИСПРАВИТЬ ЧТО-НИБУДЬ?

НЕТ КС.

НЕ ВСЛЕДСТВИЕ, А ВОПРЕКИ ЛУКАС, Д Р Ж9

ПЕРЕПЕЧАТАЙТЕ ССЫЛКУ НА ЖУРНАЛ

Ж4 КС.

003. НЕ ВСЛЕДСТВИЕ, А ВОПРЕКИ/ЛУКАС, Д Р/ФИЛ. К./ОКТЯБРЬ/  
1966

ХОТИТЕ ЛИ ВЫ ИСПРАВИТЬ ЧТО-НИБУДЬ?

НЕТ КС.

004. О ЛИБЕРАЛИЗМЕ МИЛЛА/МАК-КЛОСКИ, Г Д/ФИЛ. К./ЯНВАРЬ/  
1966

ХОТИТЕ ЛИ ВЫ ИСПРАВИТЬ ЧТО-НИБУДЬ?

ДА КС.

ЧТО ВЫ ХОТИТЕ ИСПРАВИТЬ?

ДАТУ КС.

ПРОДОЛЖАЙТЕ

1963@ КС.

004. О ЛИБЕРАЛИЗМЕ МИЛЛА/МАК-КЛОСКИ, Г Д/ФИЛ. К./ЯНВАРЬ/  
1963

ХОТИТЕ ЛИ ВЫ ИСПРАВИТЬ ЧТО-НИБУДЬ?

НЕТ КС.

(Если исправления содержали ошибки, то их можно устранить в момент получения указанного выше вопроса от системы.)

ЕЩЕ РАЗ О ЛИБЕРАЛИЗМЕ МИЛЛА МАК-КЛОСКИ, Г ДЖ4

ПРОБЕЛ ПРОПУЩЕН?

ДА КС.

005. ЕЩЕ РАЗ О ЛИБЕРАЛИЗМЕ МИЛЛА/МАК-КЛОСКИ, Г Д/ФИЛ.  
К./ЯНВАРЬ/1966.

ХОТИТЕ ЛИ ВЫ ИСПРАВИТЬ ЧТО-НИБУДЬ?

НЕТ КС.

ИСТОЧНИК ОГОРЧЕНИЙ ТЕЙЛОР, ДМ

ВЕРНА ЛИ ДАТА—69?

(Этот диалог происходил до 1969 года, и поэтому дата 1969 была ошибочной.)

НЕТ КС.

НАПЕЧАТАЙТЕ ПРАВИЛЬНУЮ ДАТУ

65 КС.

ИНИЦИАЛЫ ПРАВИЛЬНЫЕ?

НЕТ КС.

ЗАПЯТАЯ ПРОПУЩЕНА?

НЕТ КС.

НЕДИАГНОСТИРУЕМАЯ ОШИБКА ПОЖАЛУЙСТА ПЕРЕПЕЧАТАЙТЕ  
СВЕДЕНИЯ ПОЛНОСТЬЮ

(Программа РЕДАКТИРОВАНИЕ останавливается, и сведения о документе  
перепечатываются на выносном пульте.)

ИСТОЧНИК ОГОРЧЕНИЙ ТЕЙЛОР, Д М Ж4 ЯНВ65@КС.

006. ИСТОЧНИК ОГОРЧЕНИЙ/ТЕЙЛОР, Д М/ФИЛ. К./ЯНВАРЬ/1965

ХОТИТЕ ЛИ ВЫ ИСПРАВИТЬ ЧТО-НИБУДЬ?

НЕТ КС.

007. О ПРАВАХ/МАК-КЛОСКИ, Х Д/ФИЛ. К./АПРЕЛЬ/1965

ХОТИТЕ ЛИ ВЫ ИСПРАВИТЬ ЧТО-НИБУДЬ?

НЕТ КС.

(Ошибка в названии месяца устранена системой автоматически.)

008. ФИЛОСОФИЯ И ТЕОЛОГИЯ/ФИЛИПС, Д З/ФИЛ. К./ОКТЯБРЬ/1963

ХОТИТЕ ЛИ ВЫ ИСПРАВИТЬ ЧТО-НИБУДЬ?

ДА КС.

ЧТО ВЫ ХОТИТЕ ИСПРАВИТЬ?

НАЗВАНИЕ КС.

ПРОДОЛЖАЙТЕ

ФИЛОСОФИЯ И ТЕОЛОГИЯ КС.

008. ФИЛОСОФИЯ И ТЕОЛОГИЯ/ФИЛИПС, Д З/ФИЛ. К./ОКТЯБРЬ/1963

ХОТИТЕ ЛИ ВЫ ИСПРАВИТЬ ЧТО-НИБУДЬ?

НЕТ КС.

## 2. Поиск.

Поскольку объем памяти машины не позволяет хранить весь основной файл в виде инвертного древовидного авторского файла, то программа ПОИСК в ответ на запросы печатает не сводки о документах, а их регистрационные номера. Приводимая ниже запись беседы с системой иллюстрирует эффективность метода усечения звуков при поиске неправильно записанных имен на примере некоторых особенно трудных случаев. В беседе принимал участие немец, которому было предложено указать в запросах имена авторов, услышанные им до этого только один раз. Такой способ проверки метода представляется наиболее жестким. В каждом из указанных ниже шести имен испытываемый допустил орфографическую ошибку, иногда серьезную, но в пяти случаях имена были успешно отождествлены с именами, указанными в авторском файле. Итоги проведения более массовых проверок будут показаны на рис. 9. Правильно имена записываются следующим образом: MASCKIE, DILMAN, HINTIKKA, SRZEDNICKI, MCCLOSKEY и POLANYI.

Запись беседы выглядит так:

(Как и прежде, сначала в машину вводится программа МОНИТОР.)

ЧТО ВАМ НУЖНО: ПРИЕМ, СОРТИРОВКА, ДОБАВЛЕНИЕ ИЛИ ПОИСК?

ПОИСК КС.

ВВЕДИТЕ ПРОГРАММУ ПОИСК НАЖМИТЕ КЛАВИШУ ПУСК

ВВЕДИТЕ ФАЙЛ АВТОРОВ НАЖМИТЕ КЛАВИШУ ПУСК

(Вводится текущий файл авторов.)

СООБЩИТЕ ЗАПРОС НА ПОИСК

MASCKEY КС.

СЛЕДУЮЩИЕ ДОКУМЕНТЫ СООТВЕТСТВУЮТ ВАШЕМУ ЗАПРОСУ

015  
ЕСТЬ ДОПОЛНИТЕЛЬНЫЙ ЗАПРОС?  
ДА КС.  
СООБЩИТЕ ЗАПРОС НА ПОИСК  
DEALLMAN КС.  
СЛЕДУЮЩИЕ ДОКУМЕНТЫ СООТВЕТСТВУЮТ ВАШЕМУ ЗАПРОСУ  
014  
ЕСТЬ ДОПОЛНИТЕЛЬНЫЙ ЗАПРОС?  
ДА КС.  
СООБЩИТЕ ЗАПРОС НА ПОИСК  
HINTEKER КС.  
СЛЕДУЮЩИЕ ДОКУМЕНТЫ СООТВЕТСТВУЮТ ВАШЕМУ ЗАПРОСУ  
020  
ЕСТЬ ДОПОЛНИТЕЛЬНЫЙ ЗАПРОС?  
ДА КС.  
СООБЩИТЕ ЗАПРОС НА ПОИСК  
SCHREDNISKY КС.  
ТРЕБУЕМОГО ИМЕНИ НЕТ В ФАЙЛЕ АВТОРОВ  
ЕСТЬ ДОПОЛНИТЕЛЬНЫЙ ЗАПРОС?  
ДА КС.  
СООБЩИТЕ ЗАПРОС НА ПОИСК  
SHREDNISKY КС.  
ТРЕБУЕМОГО ИМЕНИ НЕТ В ФАЙЛЕ АВТОРОВ  
(Выполнению удачного поиска препятствует отсутствие в запросе буквы Z,  
которая есть в оригинале имени.)  
ЕСТЬ ДОПОЛНИТЕЛЬНЫЙ ЗАПРОС?  
ДА КС.  
СООБЩИТЕ ЗАПРОС НА ПОИСК  
MSCLUSCY КС.  
СЛЕДУЮЩИЕ ДОКУМЕНТЫ СООТВЕТСТВУЮТ ВАШЕМУ ЗАПРОСУ  
005  
006  
007  
008  
(В системе, работающей в режиме реального времени и снабженной видео-  
экранами, пользователь может световым пером указать на экране интересую-  
щий его документ.)  
ЕСТЬ ДОПОЛНИТЕЛЬНЫЙ ЗАПРОС?  
ДА КС.  
СООБЩИТЕ ЗАПРОС НА ПОИСК  
PELLANY КС.  
СЛЕДУЮЩИЕ ДОКУМЕНТЫ СООТВЕТСТВУЮТ ВАШЕМУ ЗАПРОСУ  
013  
ЕСТЬ ДОПОЛНИТЕЛЬНЫЙ ЗАПРОС?  
НЕТ КС.

Таким образом, как показывают примеры, даже на машине с ограниченными возможностями может функционировать система, предоставляющая достаточно развитые средства редактирования и исправления ошибок в библиотечной документации. Эти средства снижают до минимума необходимую канцелярскую работу, причем их использование не требует каких-либо серьезных навыков работы на машине и осуществляется в форме, удобной для библиотечного персонала. Другие средства, например, такие, как контроль недопустимых буквенных сочетаний в словах, используемых в назва-

ниях работ, также желательны и легко могут быть включены в систему в виде самостоятельных подпрограмм.

Совершенно необходимо обеспечить естественный для пользователя формат ввода, не ухудшая при этом систему с точки зрения количества пропускаемых ошибок. Пожалуй, желательнее разрешить оператору частичное исправление любого элемента с ошибкой (при условии, конечно, что такие элементы могут быть автоматически обнаружены), чем проверять вручную все элементы, многие из которых будут правильными.

Программа СОРТИРОВКА здесь не описывается. Она принимает в качестве исходного материала основной файл новых поступлений и осуществляет сортировку в алфавитном порядке полей информации или каких-нибудь частей этих полей. Конечно, в случае поля даты сортировка выполняется не согласно алфавиту, а по годам и месяцам. Программа СОРТИРОВКА состоит из ряда подпрограмм, которые считывают информацию с файла, выбирают соответствующие поля и запоминают их в таблице, выполняют сортировку по заданному количеству символов и печатают или перфорируют на ленте список упорядоченных элементов или же делают и то и другое.

В конце этой главы приведены блок-схемы программы СОРТИРОВКА, которая главным образом используется для составления так называемого инвертного файла, т. е. файла, содержащего фамилии и инициалы авторов статей вместе с регистрационными номерами документов и упорядоченного по фамилиям авторов согласно алфавиту. Этот авторский файл используется как исходный материал программой ПОИСК.

### 8.3. ПРОГРАММА ПОИСК

Эта программа выполняет две функции. Во-первых, она модифицирует древовидный файл авторов и регистрационных номеров документов, используя при этом ленту, подготовленную программой СОРТИРОВКА, и добавляя к имеющемуся файлу сведения о вновь поступивших работах. Кроме того, эта программа принимает вопросы и дает на них ответы в виде списков документов, хранящихся в файле. При этом дело обстоит так, что пользователь печатает для системы фамилии интересующих его авторов, а ему в ответ печатается список регистрационных номеров документов. Недостаточный объем памяти машины не позволил реализовать в данной программе еще одну процедуру, а именно сопоставление найденных регистрационных номеров с полными записями основного файла и сообщение пользователю подробных сведений (а не только номеров) о нужных ему документах. Однако вполне очевидно, что это должно быть сделано в реальной оперативной библиотечной системе.

При разработке поисковой программы данного типа особого внимания требуют следующие два фактора. Во-первых, время, затрачиваемое на просмотр большого файла, должно быть минимизировано, поскольку даже при использовании устройств с произвольной выборкой время, необходимое для определения местоположения нужной порции файла и чтения ее в оперативную память для проведения соответствующих операций, может оказаться значительным для пользователя, ожидающего ответа от системы на выносном пульте. К числу обстоятельств, препятствующих быстрому выполнению просмотра, относятся плохая организация файла, которая влечет за собой массу просмотров информации, а также записи с неопределенной длиной, которую необходимо вычислить прежде, чем производить сравнение записи с запросом. Другой важный фактор заключается в том, что при составлении запроса пользователь может ошибиться в написании фамилии автора из-за невнимательности или же потому, что он не знает, как точно пишется фамилия. Например, студент, прослушавший лекцию, может и не знать со всей определенностью, написана ли рекомендованная ему работа Миллером, Мюллером или же Мойллером. Нужно, чтобы система по возможности учитывала двусмысленности и ошибки в запросе. Еще одно соображение касается того, что объем работ по программированию системы значительно уменьшится, если удастся использовать одну и ту же подпрограмму как для запоминания элементов информации, так и для поиска их при обработке запроса.

Описываемая программа дает возможность быстро и эффективно просматривать файл и, как показывает накопленный опыт, может отвечать на большинство запросов, содержащих ошибки в именах. Программа экономна с точки зрения объема, что достигнуто использованием одного и того же алгоритма для запоминания и выборки элементов информации. Исключение записей не запрограммировано, но эту процедуру легко подключить.

На первом этапе работы программы происходит чтение элемента информации, который должен быть запомнен или согласно которому будет выполнен поиск, и этот элемент преобразуется с помощью упоминавшегося выше метода усечения звуков в четырехбуквенную закодированную запись. При этом первая буква сохраняется такой, какой она была, а затем остальная часть имени автора кодируется до тех пор, пока не будет сформирован элемент, содержащий четыре символа, или не окажется исчерпаной запись имени. Если имя слишком длинное, то оно сокращается за счет того, что правый конец его не учитывается. Если же имя будет слишком коротким, то закодированный элемент дополняется нулями.

Имя кодируется путем исключения из рассмотрения всех гласных букв, а также букв *h* и *w* и присваиванием каждой согласной букве, содержащейся в строке имени, числового значения из диапа-

зона от 0 до 6. Двойные буквы обрабатываются при кодировании, как одиночные согласные. Очень непохожим между собой буквам, которые, вероятно, нельзя спутать, присваиваются одинаковые числа, так же как очень похожим согласным, например *m* и *n*. Идея рассматриваемого способа кодирования заключается в том, что одна и та же кодовая комбинация может быть присвоена именам, которые либо отличаются друг от друга только гласными буквами, либо содержат очень похожие согласные буквы. Это означает, что закодированные таким способом имена, даже если в них допущены незначительные орфографические ошибки, могут быть успешно использованы при выполнении поиска. При этом, конечно, может быть найден ряд дополнительных документов, не относящихся к рассматриваемому запросу, однако если пользователь точно не знает имя автора или допустил ошибку из-за невнимательности, то лучше предоставить ему слишком много документов, чем слишком мало. После же того, как через соответствующее внешнее устройство будут сообщены все подробности о найденных согласно данному запросу документах, пользователю, как правило, очень просто установить, какие из документов ему подходят. Наконец, поскольку для всех элементов файла авторов установлена фиксированная длина (четыре цифры), то поиск может быть организован без использования специальной отметки, указывающей окончание записи.

Файл имен авторов хранится в виде древовидной структуры, в которой для каждой буквы алфавита предусмотрена одна запись (так называемая запись буквы), расположенная на самом верхнем уровне дерева. Это сделано в связи с тем, что первая буква каждого имени, как уже отмечалось выше, в процессе кодирования сохраняется. Помимо самой буквы, запись буквы содержит указатель на первую кодовую цифру первого из запомненных в данном файле имен, начинающихся с данной буквы. Все записи кодовых цифр содержат три элемента, а именно саму кодовую цифру и два указателя, один из которых ссылается на следующую кодовую цифру в данном имени, а другой — на альтернативную кодовую цифру, входящую в другое имя. Другими словами, эти указатели соответственно ссылаются на очередную запись, расположенную на следующем уровне древовидной структуры, и на альтернативную запись, относящуюся к тому же уровню, что и рассматриваемая. Все это очень хорошо можно пояснить примером, в котором для ясности не будут использованы кодовые формы слов. Допустим, что необходимо запомнить в древовидном файле слова «heat» и «head». Такой файл будет выглядеть следующим образом:

запись для *h* — указатель на *e*  
запись для *e* — указатель на *a*  
запись для *a* — указатель на *t*

запись для  $t$  — (признак конца) — указатель на  $d$

запись для  $d$  — (признак конца)

Если к файлу будет добавлен элемент «herd», то получится следующая структура:

запись для  $h$  — указатель на  $e$

запись для  $e$  — указатель на  $a$

запись для  $a$  — указатель на  $t$  — указатель на  $r$

запись для  $t$  — (признак конца) — указатель на  $d$

запись для  $d$  — (признак конца)

запись для  $r$  — указатель на  $d$

запись для  $d$  — (признак конца)

По рис. 8 можно проследить графическую структуру файла. При включении нового элемента в файле запоминаются только те буквы, которые отличаются от встречавшихся ранее, и сам файл

Имя	Код	Документы
Tomkin	T525	25
Tomlinson	T545	26
Timberly	T516	32,43

Если запомнить три приведенных выше элемента в древовидном файле в указанном порядке, то этот файл выглядел бы следующим образом:

Адрес	Элемент	Указатель 1	Указатель 2
1000	T	1010	—
1010	5	1020	—
1020	2	1030	1040
1030	5	25	—
1040	4	1050	1060
1050	5	26	—
1060	1	1070	—
1070	6	32	1080
1080	—	43	—

Р и с. 8. Древовидный файл для закодированных имен.

может расширяться по мере необходимости. В случае использования больших файлов это обстоятельство обеспечивает довольно значительную экономию памяти и возможность простого просмотра информации в файле. Если два элемента имеют полностью одинаковые кодовые имена, то регистрационный номер первого содержится в поле первого указателя, принадлежащего записи последней кодовой цифры, а второй указатель этой записи определяет запись, в которой (опять-таки в поле первого указателя) находится регистрационный номер другого элемента. Очень наглядно этот случай поясняется примером, приведенным на рис. 8. Если, кроме самой фамилии, в качестве вспомогательной характеристики при поиске используются и инициалы автора, то абонент, аккуратно указав-

ший фамилию и инициалы, получит от системы такой список рекомендуемых документов, который содержит очень мало работ, не относящихся к делу. И напротив, невнимательный пользователь в этом случае вынужден будет рассматривать согласно предложенному системой списку большое количество работ, отбирая из них необходимые ему. Такая библиотечная процедура несколько напоминает процесс наведения справок по телефонной книге.

Просмотр файла начинается с записи первой буквы имени автора и продолжается далее согласно ссылкам, содержащимся в указателях записей. Если в какой-нибудь момент просмотра будет обнаружено, что обрабатываемый входной элемент не совпадает с имеющимися в файле элементами и при этом на данном уровне дерева нет никаких альтернативных связей, то просмотр считается

Имена были выбраны из телефонной книги случайным образом и предложены различным участникам эксперимента; приводимая ниже таблица показывает, сколько из 20 неправильно названных каждым участником имен были успешно найдены системой в файле.

Группа из 20 неправильно записанных имен	Количество успешно найденных имен
A	15
B	17
C	17
D	16
E	19
F	18
G	15
H	20
I	18

Эксперименты показали, что приведенные выше показатели могут быть значительно улучшены, если дополнить программу процедурами анализа некоторых специфических созвучий в языке.

#### Р и с. 9. Поиск ошибочно записанных имен.

завершенным. После этого, если выполняется операция запоминания новой информации, формируются соответствующие указатели и добавляются новые записи. Если же просмотр файла выполнялся с целью поиска документов, то система констатирует, что требуемого элемента в файле нет. С другой стороны, если в процессе просмотра будет зафиксировано совпадение входного элемента с одним из элементов, находящихся в файле, то либо в файл заносится запись, содержащая регистрационный номер нового документа, либо печатаются регистрационные номера элементов, совпавших с указанным в запросе. Следовательно, значительная часть процедуры кодирования, применяемой при выполнении просмотра древовидного

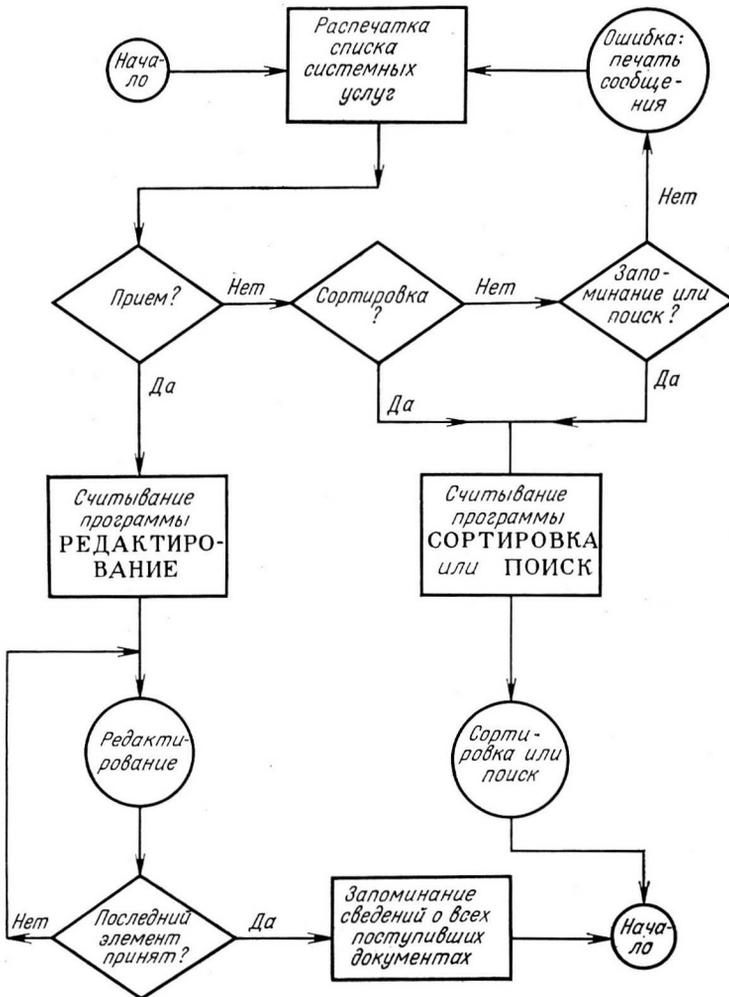
файла, оказывается общей как для операции запоминания информации, так и для операции поиска.

Программа работала весьма успешно при поиске элементов, которые были в запросе написаны с ошибками. При проведении одного из экспериментов «беседу» с системой вел немец, печатавший на клавиатуре терминала в запросах те имена, которые он только слышал, но не видел как они записываются на английском языке. Запись этой «беседы» с системой была приведена на стр. 84—85. Другой эксперимент, проведенный над большим количеством имен, дал результаты, анализируемые на рис. 9.

Рассмотренные примеры программ, не претендуя ни на что, кроме общего описания проблем, связанных с автоматизацией библиотечной деятельности, тем не менее показывают, что даже с ограниченными программными средствами можно создать работающую библиотечную систему. В этой системе предусматривается решение некоторых специальных проблем, возникающих, когда люди, не имеющие опыта работы в области вычислительной техники, получают доступ к терминалу и машине. Разумеется, недопустимо, чтобы неосведомленность пользователя вредила системе. Проблемы взаимодействия с пользователем, касающиеся обработки возможных неточностей и ошибок, возникающих в процессе общения, могут быть успешно решены без ухудшения эксплуатационных характеристик системы и без значительного увеличения объема выполняемой ею работы.

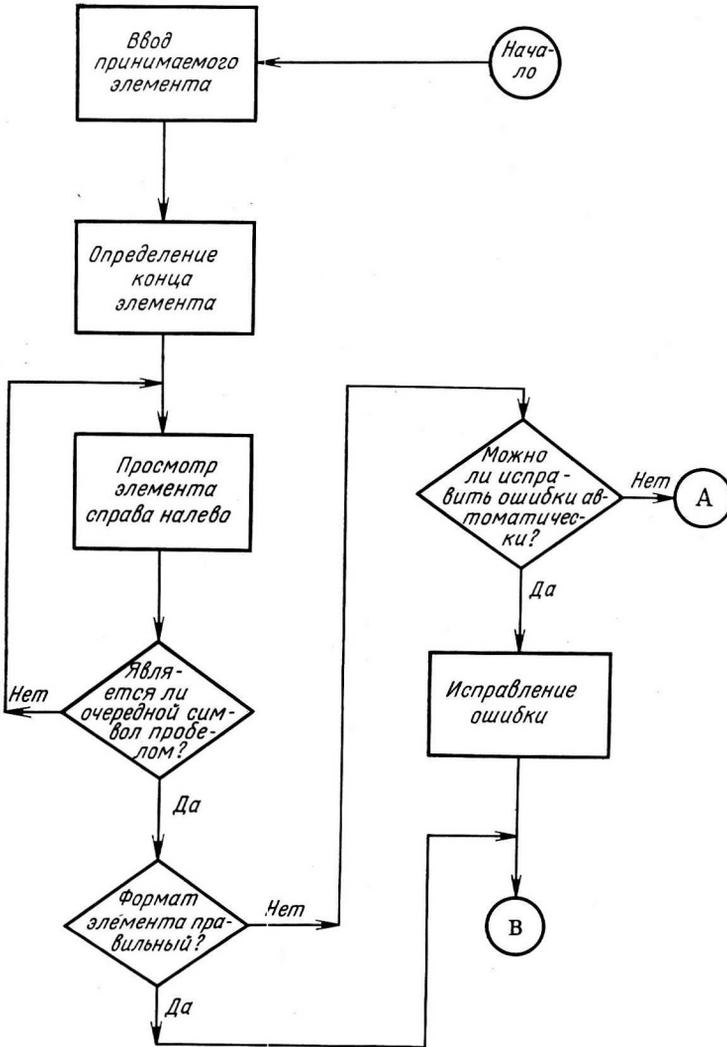
## БЛОК-СХЕМЫ

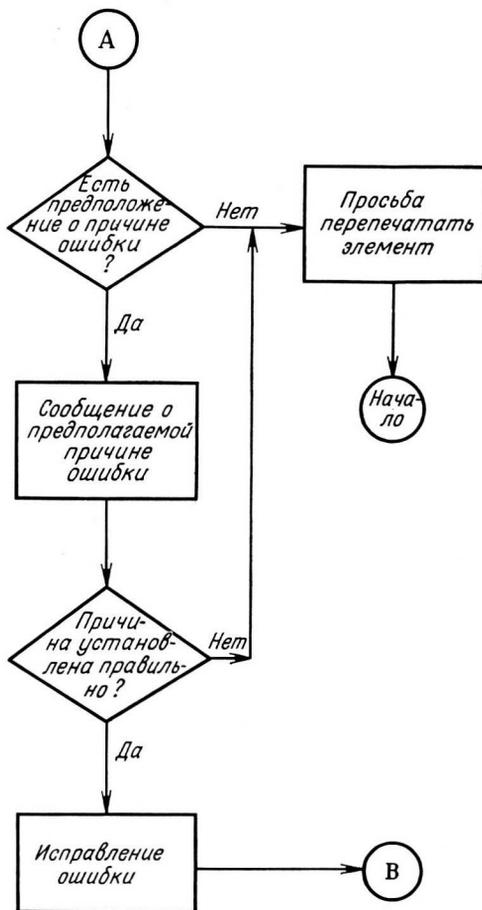
На последующих страницах приведены блок-схемы трех программ РЕДАКТИРОВАНИЕ, СОРТИРОВКА и ПОИСК, а также краткая блок-схема МОНИТОРА.

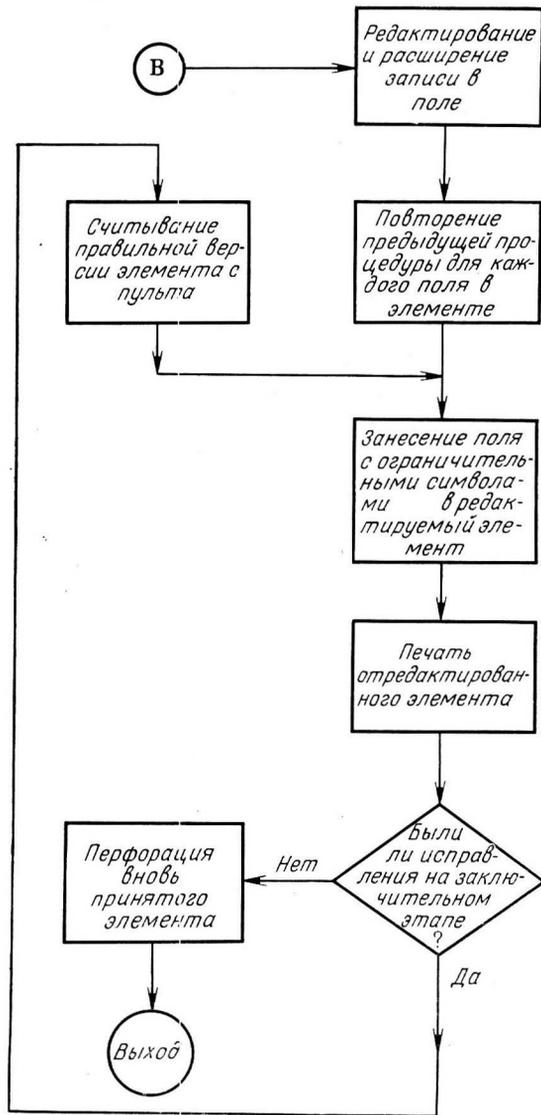


Блок-схема программы МОНИТОР

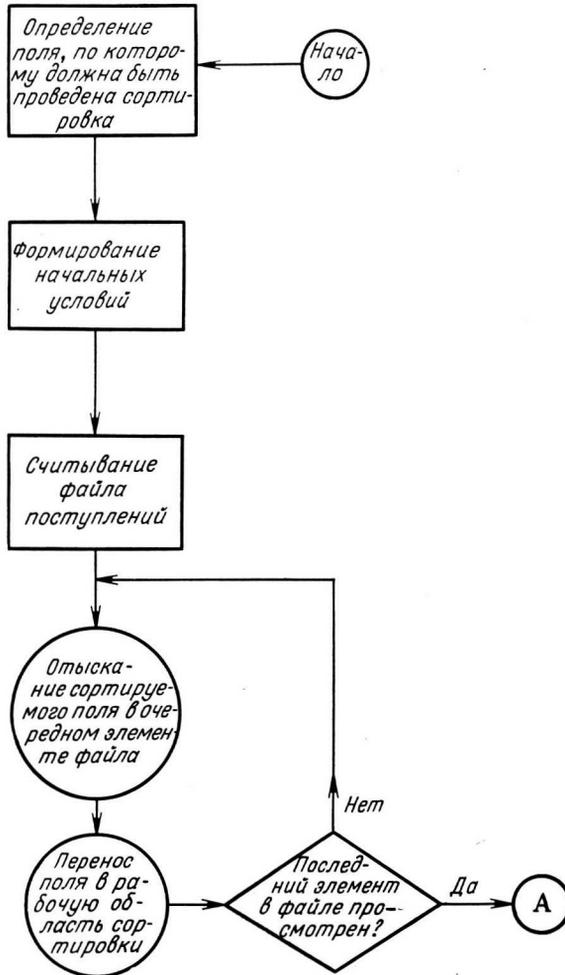
# 1. Программа РЕДАКТИРОВАНИЕ

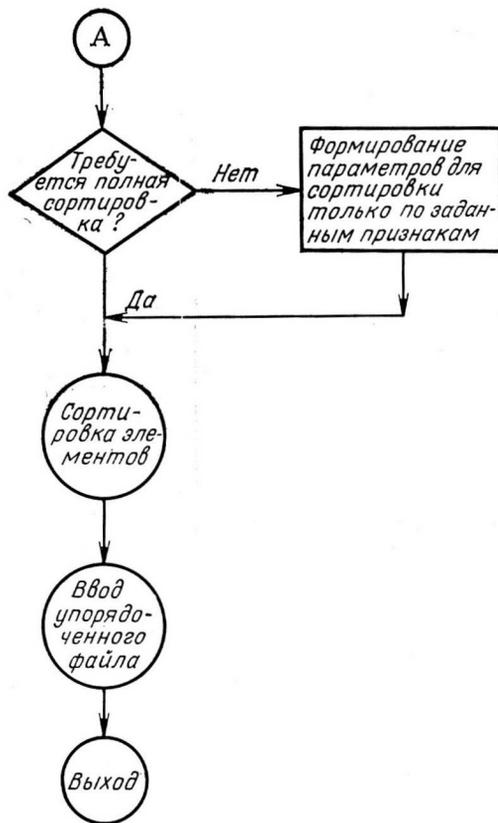




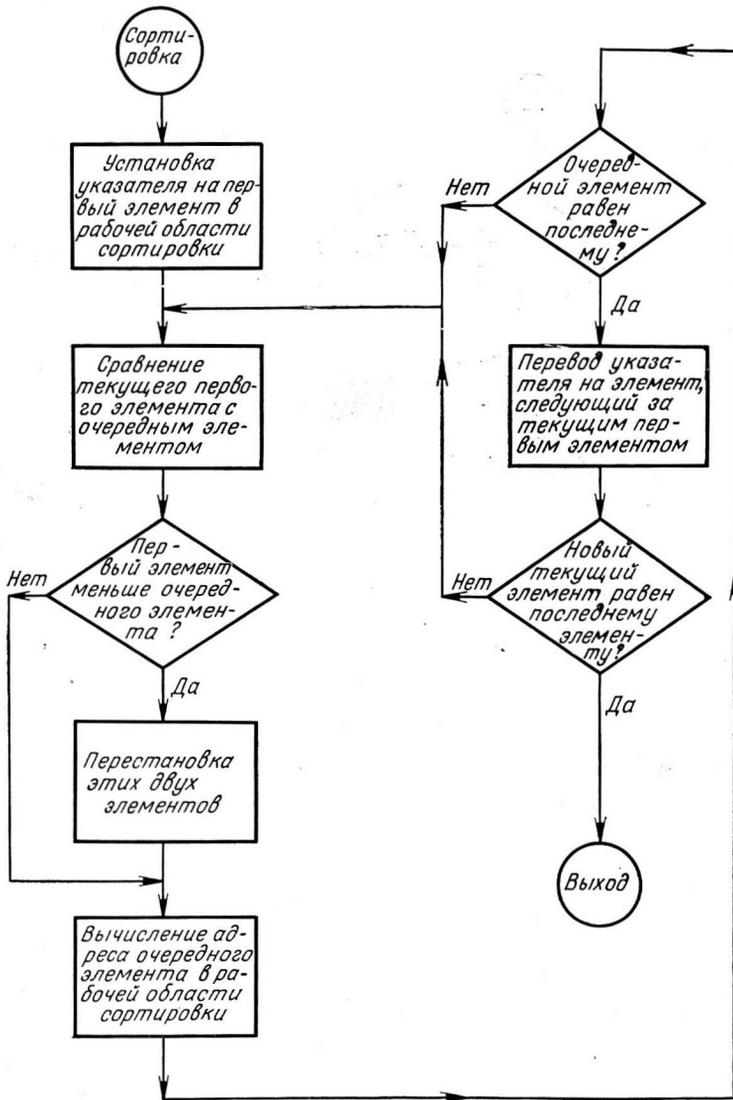


## 2. Программа СОРТИРОВКА

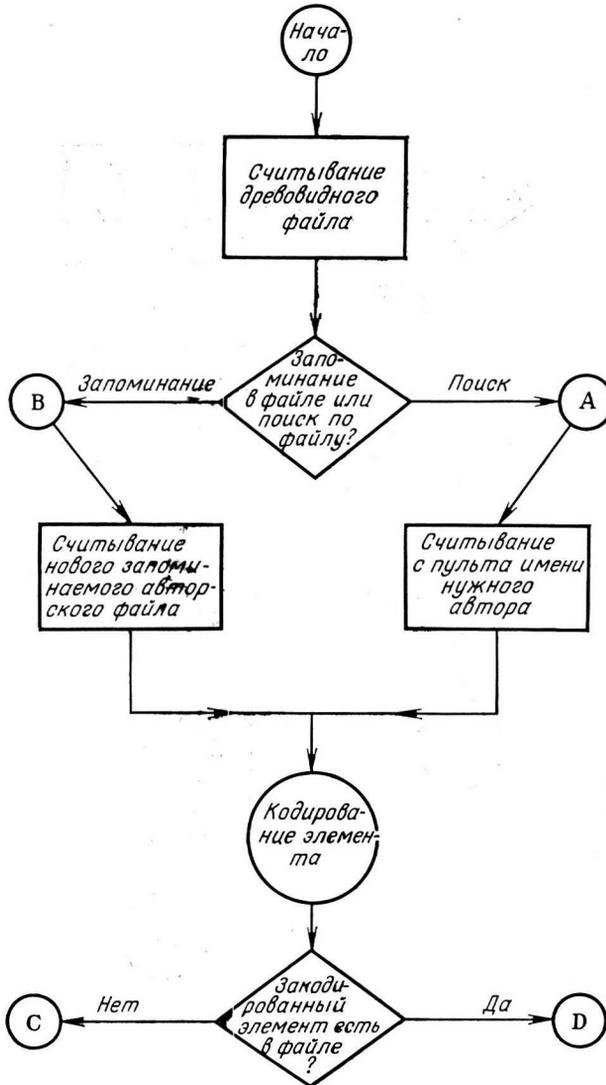


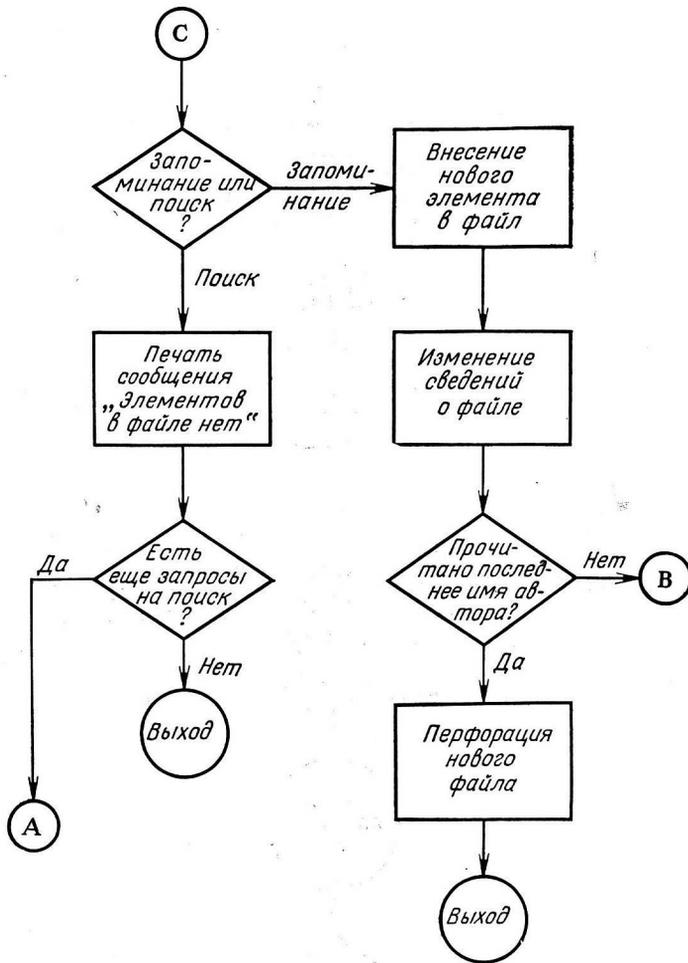


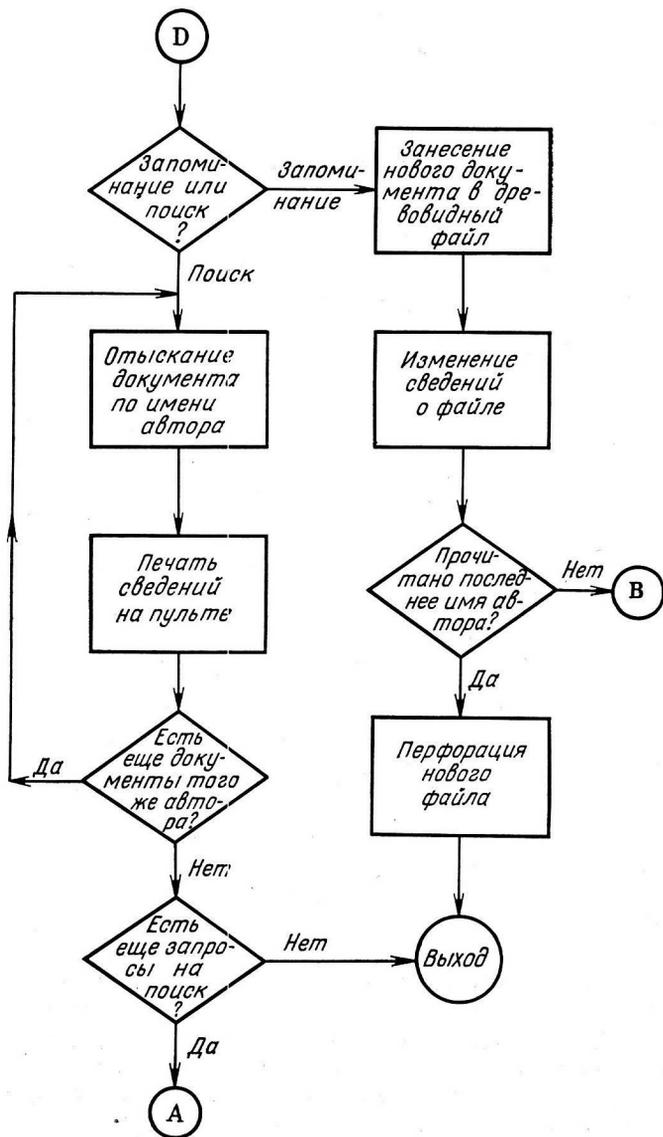
(Подробное описание блока «Сортировка элементов» приведено на следующей странице.—Ред.)



### 3. Программа ПОИСК







## ОБЩИЕ ПРИНЦИПЫ АВТОМАТИЗИРОВАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ

---

### 9.1. ФАКТЫ ИЛИ ДОКУМЕНТЫ?

Цель этой книги состояла в том, чтобы познакомить читателя, не имеющего специальной подготовки, с вычислительной машиной как средством автоматизации библиотечной службы, наметить в общих чертах логический путь передачи машине все большего числа различных библиотечных процедур и кратко рассмотреть отдельные методы, которые могут пригодиться при разработке автоматизированной библиотечной системы. В данной заключительной главе предпринята попытка обсудить некоторые предположения о дальнейших путях автоматизации библиотечного дела, при этом будут учтены некоторые возражения, касающиеся описанных выше методов.

Прежде всего необходимо подчеркнуть, что, хотя, как показали предыдущие рассмотрения, с помощью машины и автоматизируется значительная часть библиотечной работы, тем не менее функции библиотеки по существу остаются теми же самыми. Они заключаются в информационном обслуживании библиотечных пользователей путем хранения большого количества различных документов и предоставления в отдельных случаях этих документов абонентам.

Другими словами, хотя автоматизированная библиотека и может весьма успешно удовлетворять специфические реальные запросы, все же основной поток информации поступает к пользователям из документов, которые остались в том же виде, как они были написаны различными авторами. Описанная выше система на самом деле является системой поиска документов, а не системой поиска фактов.

Можно рассмотреть противоположную концепцию информационно-поисковой системы. В этой связи уместно сослаться на работу Ликлидера [3] «Библиотеки будущего», в которой он высказывает предположение, что существующие системы поиска документов в конечном счете будут заменены информационной системой, основанной на диалоговом режиме общения с пользователем и сообщаемой ему в ответ на информационный запрос сведения в форме конкретных фактов. Ликлидер пишет:

Одна из главных задач, стоящих перед системными специалистами, заключается в том, чтобы обеспечить выборку информации из документа,

связанного с неким подразделом знаний, и передачу ее в формируемую тематическую подборку, соответствующую этому подразделу знаний.

Такого рода подход, который можно рассматривать как последний шаг на намеченном нами выше пути автоматизации библиотеки, подразумевает несколько исходных предположений относительно информационного поиска, требующих тщательного изучения. Ликлидер считает, что не следует ограничивать применение этой идеи одной или двумя строго определенными областями науки и техники, т. е. создавать специализированную систему вроде тех, что используются при программированном обучении, а скорее, напротив, нужно разрабатывать на основе этой идеи систему общего назначения, применимую ко всем областям человеческих знаний, такую систему, в которой машина используется в качестве громадного хранилища знаний, накопленных поколениями, причем вся эта информация тщательно упорядочена в форме совокупностей фактов, сообщаемых системой в ответ на соответствующие запросы. Для успешного решения такой задачи потребуются надлежащим образом определенный внутренний машинный язык, посредством которого могут быть компактно записаны все хранящиеся в машине факты, и Ликлидер упоминает в своей работе об экспериментах с языками, основанными на исчислении предикатов, которое может служить базой такого внутреннего языка.

Существует несколько возражений против такого мнения. Во-первых, абсолютно невозможно найти или придумать для изложения фактов язык, который был бы достаточно общим и в то же самое время был настолько формален, что его удобно использовать в автоматизированной системе. Как признает сам Ликлидер:

Небольшое высказывание на естественном языке превращается в большое выражение исчисления предикатов;

более того,

до сих пор нет метода автоматического перевода предложений естественного языка в выражения исчисления предикатов.

Кроме того, для разговорных систем, подобных той, что предложена выше, характерна логическая сжатость ответов при ведении диалога, что делает их привлекательными для специалистов в области логики, но затрудняет общение с ними остальных пользователей.

Даже если допустить, что такой язык может быть разработан, все равно остается ряд серьезных сомнений относительно этого подхода к информационной системе.

Трудно представить себе реальную систему, работающую успешно в условиях, когда возможный круг вопросов, на которые она должна отвечать, весьма разнообразен, причем эти вопросы могут оказаться достаточно сложными и общими. Проведенный недавно опрос [6] показал, что в условиях академической библиотеки около половины вопросов, возникающих у абонентов, носят специальный

характер, так что необходимость в справочно-информационной системе <sup>1)</sup> очевидна; однако любая автоматизированная информационная система должна быть строго ориентирована на некоторую заданную область знаний.

Другая проблема связана с вводом информации. Вопрос заключается в том, в какой форме новая информация должна поступать в систему, обеспечивающую ответы на запросы обязательно в виде справочного материала. Должна ли машина просматривать поступающие документы, переводя их при этом на некий всеобъемлющий внутренний язык, решать затем, содержатся ли какие-нибудь новые факты в поступивших документах, и если это так, то дополнять такими фактами уже имеющуюся информационную подборку, или же, напротив, вся вводимая информация будет рассматриваться как сплошной поток данных, который не делится на самостоятельные порции, понимаемые как отдельные документы? В последнем случае на рассмотрение системы представляется совокупность фактов как таковых, и система учитывает, связывает и запоминает или же оставляет без внимания те или иные факты на основании уже ранее кем-то проведенного подбора данных. Эта грандиозная картина, в которой машина выступает в качестве главного арбитра в мире без книг и авторов, выглядит как пример одного из машинных «фокусов-покусов».

Основное возражение против такого рода подхода заключается в том, что, по-видимому, неправильно считать, будто бы любые факты и знания могут быть упорядочены в виде строгой иерархии. Рассматривать человеческие знания как простую совокупность взаимосвязанных концепций — это значит вернуться к античным представлениям о человеческом познании! Если такой вопрос, как: «Какова точка кипения ртути?», можно считать требующим фактического ответа, который может быть выдан автоматизированной системой, то запрос типа: «Каковы истоки поэтического гения Мильтона?» или «Каковы причины возникновения первой мировой войны?», хотя он и сформулирован аналогичным образом, вряд ли относится к той категории вопросов, которые нам следует задавать системе. Упаси нас бог от выработанных машиной версий истории или философии.

Дело в том, что многое из того, что мы называем информацией, вообще не является эмпирическим фактом. Очень часто это просто мнение из информированного источника или заключение специалиста. При этом, чтобы выразить свой интерес определенным образом, мы должны задавать вопросы, например, такого рода: «Каковы по мнению такого-то причины возникновения войны?», однако вопрос, подобный данному, подозрительно похож на завуалированный запрос, касающийся некоего документа. В таком случае, несом-

<sup>1)</sup> То есть системе поиска фактов. — *Прим. ред.*

ненно, лучше прочитать, что сказал по данному поводу такой-то, чем изучать выработанную машиной и сформулированную в предикатном исчислении версию. При этом знание не представляет собой простой композиции из тщательно определенных фактов и не является чем-то четко упорядочиваемым в информационную иерархическую структуру. Машина не должна решать сверхчеловеческую задачу, заключающуюся в установлении скрытых связей между некоторого рода фактами и концепциями.

## 9.2. КОЛИЧЕСТВЕННЫЕ ОЦЕНКИ

Это приводит к другому широко используемому в области документалистики предположению о том, что оценочные коэффициенты, или словарная частотность слов, являются вполне приемлемым средством для группирования документов. Многие из упоминавшихся в предыдущих главах методов в конечном счете основываются на некой разновидности количественной оценки. Документы группируются друг с другом или связываются с поступившим запросом на поиск информации в зависимости от того, насколько при их анализе проявляется требуемая словарная частотность или насколько оказывается превышенным заданный критерий, полученный в результате численного расчета. Можно возражать против использования того или иного метода количественной оценки или вообще возражать против таких методов, как это делает Бар-Хиллел [7]. Он цитирует возражение Д. Дж. Фоскетта против:

попыток замаскировать общепринятые понятия малопонятным, а иногда и надуманным псевдонаучным жаргоном.

Все это дополняется «массой впечатляющих математических выкладок и диаграмм». Однако вместо таких общих негативных соображений более разумно обратиться к анализу отдельных областей информационного поиска, которые особенно пригодны для использования методов количественных оценок. В связи с этим первое возражение, которое следует высказать, касается идеи о том, что индекс должен быть своего рода миниатюрой документа. В результате применения процедуры координатного индексирования должен быть получен список терминов, которые точно описывают содержание документа. Как правило, это достигается с помощью некоего метода анализа наиболее часто используемых в документе слов, исключения незначачих слов и выбора оставшихся слов в качестве характерных терминов, предназначенных для описания документа. Следует отметить, что всякий такой метод подразумевает различные предположения относительно стиля изложения материала. Можно согласиться с тем, что постоянное повторение одного и того же является типичной чертой плохого стиля, что если автор использует для точного описания своей работы неизвестное слово,

то возникнут неприятности при поиске документа и что работы, которые мало добавляют к уже имеющейся информационной подборке, будут более легко восприняты системой, чем новые работы в новой области. Необходимо также иметь в виду, что это в значительной степени верно для существующих обычных библиотечных систем. Однако основная проблема заключается в том, что такая система не только возлагает на автора чрезмерную ответственность в отношении аккуратного описания работы, но также и ограничивает его набором терминов, которыми он может пользоваться, если не хочет, чтобы его работа оказалась навсегда потерянной.

Таким образом, есть два основных возражения против количественных оценок: во-первых, количественный анализ не может обеспечить получение точного миниатюрного описания документа и, во-вторых, даже если такое описание может быть получено, то использовать его для организации информационного поиска нежелательно, поскольку это накладывает на авторов работ неприемлемые ограничения.

Аналогичное возражение можно высказать относительно использования коэффициентов и математических формул. Подобно тому как в некоторых источниках проявляется тенденция рассматривать факты как самостоятельные категории, соответствующие какой-то заранее сформированной структуре, так и оценочным методом можно придать некий абстрактный смысл. Как замечает Айви [32]:

Слишком уж часто эти замечания (об оценках соответствия) оставляют человека с ощущением того, что есть некоторый внутренний смысл слова или документа, который теперь количественно описывается, тогда как на самом деле ничего не достигнуто, кроме новой интерпретации коэффициентов частотности.

Важно попытаться увидеть истоки этих тенденций; создается впечатление, что они происходят в основном от интерпретации термина «информация». Понятие «информация» в контексте теории информации точно определено и допускает математическую оценку, и это часто является отправной точкой для многих экспериментов в области документалистики.

Однако при использовании в библиотечном контексте слово «информация» приобретает более земной, ненаучный смысл. С таким понятием информации мы широко соприкасаемся в ситуации информационного поиска. Конечно, создается впечатление, что количественная концепция информации, используемая в теории информации, оказывает влияние на формирование концепции информационного поиска, однако в действительности математическая теория информации и информационный поиск имеют мало общего. Необходимо отказаться от подхода, связанного с попыткой обнаружить какие-то неотъемлемые связи между фактами, документами или предметами, и принять в качестве определяющего обстоятельства в данном случае точку зрения пользователя, в основе которой

лежит принцип группирования документов с позиции соответствия их конкретным запросам, возникающим при поиске информации. Как раз на эту разумную позицию и ориентируются некоторые экспериментальные методики по группированию документов. Однако необходимо подчеркнуть, что при этом не стараются использовать машину для решения сверхчеловеческой интеллектуальной задачи, а поручают ей установить порядок внутри данной группы документов согласно заданному контексту, не подгоняя документы под некоторую неизменную придуманную схему.

### 9.3. ОТВЕТСТВЕННОСТЬ ЗА СИСТЕМУ

Два типичных возражения, отмеченных в предыдущем разделе, необходимо учитывать для того, чтобы обеспечить оправданное использование автоматизированной библиотечной системы. Первое из них заключается в том, что методы количественных оценок не могут обеспечить составителя индексного указателя достаточно точным описанием документа, пригодным для непосредственного использования в интересах информационного поиска. Вообще говоря, до некоторой степени этот вопрос связан с конкретной реализацией системы. Если система продумана и разработана аккуратно, использует составленный пользователями и поддерживаемый обслуживающим персоналом тезаурус, то при поиске документов в такой системе могут быть достигнуты вполне успешные результаты. Однако, как отмечалось в предыдущей главе, необходимо предусмотреть в системе и использование обратной связи, особенно если при этом можно использовать мнения избранных экспертов. Система избирательного распространения информации предоставляет идеальную возможность для учета в системе некоторых количественных оценок, полученных через комментарии постоянных читателей в данной области. Обычно читатель редко требует сведения о всех предыдущих работах в данной области знаний. Гораздо чаще он хочет получить представительный обзор, охватывающий все фундаментальные книги и статьи, а также некоторые из наиболее интересных или оригинальных публикаций в этой области. Составление такого рода обзора, очевидно, является обязанностью сотрудника библиотеки, которому оказывается помощь со стороны экспертов в данной области. Истоки методик, основывающихся исключительно на количественных оценках, связаны с неудачей, постигшей некоторую часть специалистов в области вычислительной техники, когда они пытались достаточно тщательно проанализировать имеющийся библиотечный контекст. Однако если принять соответствующие меры и обеспечить взаимодействие экспертов и библиотечного персонала, то методы количественных оценок можно вполне успешно использовать в библиотечной системе.

Второе возражение, связанное с тем, что вычислительная система предъявляет пользователю жесткие требования, поднимает общий вопрос о том, кто несет ответственность за автоматизированную систему. До некоторой степени решение вопроса о том, кто несет ответственность за успех или неудачу рабочей системы, в некоторой мере влияет на то, какие из имеющихся в распоряжении средств обработки данных будут приняты в системе. Первым кандидатом в этом плане является пользователь системы. Действительно, от него исходят запросы на информацию, и, как отмечалось выше, он может оказать большую помощь в составлении и расширении тезауруса, или списков терминов, используемых в запросах. Однако система предназначается в первую очередь для того, чтобы обеспечить возможность установления связи между пользователем и машиной, и все, что препятствует достижению этой цели, должно быть отвергнуто. Это обстоятельство должно быть определяющим при рассмотрении вопроса о том, стоит ли попытаться использовать специальный язык для обеспечения связи между человеком и машиной. Эта связь должна быть удобной независимо от специфики выбранного языка и независимо от того, какие именно пользователи будут работать с системой.

Некоторые разработки, например требующие вычисления «пороговых» характеристик и даже те, что основаны на сложных комбинациях терминов, используемых в запросах, выглядят неудачными, когда они оцениваются с точки зрения простоты использования. В этой связи Дойл замечает:

Существующие автоматизированные поисковые системы увеличили физический и психологический разрыв между читателями и информационным хранилищем.

В связи с этим очевидна задача для библиотечного сотрудника. По крайней мере в случаях, когда возникают сложные или трудные ситуации, сотрудник библиотеки должен продолжать действовать так, чтобы обеспечить связь между системой и информационным хранилищем, если только система достаточно проста для использования в нормальных условиях. Тщательно составленный пакет программ с хорошими мониторными средствами и набором обслуживающих подпрограмм может оказаться при этом весьма и весьма полезным, однако совершенно недопустимо пытаться от отчаяния «дрессировать» пользователя.

Можно было бы выдвинуть в качестве лица, ответственного за правильность работы системы, самого автора документа. Авторы возражают против того, что автоматизированная система вынуждает их аккурратно излагать свои соображения. Однако можно высказать мысль, что это не так уж и плохо. В связи с автоматическим резюмированием документов Сайбург [35] замечает:

В конечном счете любые средства будут удачны или неудачны в зависимости от того, насколько удачно или неудачно автор выразил свои мысли в работе.

И далее,

Можно надеяться, что когда автоматическое резюмирование неизбежно станет широко использоваться, оно побудит авторов излагать ясно и подробно основные соображения своих статей.

Это представляется слабым оправданием неумения справиться с трудностями создания хорошей системы.

Всякие неразумные ограничения, накладываемые на автора или пользователя, должны быть отвергнуты. Поэтому всю ответственность за качество работы системы несут совместно программисты и сотрудники библиотеки. Программисты должны разработать широкий набор средств, простых и легких в использовании, причем особенно необходимо предусмотреть те средства, которые предотвращают тупиковые ситуации в процессе работы системы из-за пропуска запятой или неправильно написанного инициала. Этому аспекту не так уж часто уделяется внимание, но он так же важен, как и другие.

Сотрудник библиотеки должен знать, какие недостатки есть в системе, и добиться, чтобы программисты разработали средства для их устранения. Он должен быть хорошо знаком с системой и принципами ее работы, с тем чтобы он мог осуществлять связь между пользователем и хранилищем информации, и здесь уместно сказать, что библиотечный персонал должен уметь управлять машиной в такой же степени, в какой специалисты по вычислительной технике должны быть знакомы с требованиями библиотеки. Однако применение машины должно освободить библиотечный персонал от выполнения обширной канцелярской работы. Хотя из соображений экономии и накопления опыта может оказаться необходимым осуществлять ввод в эксплуатацию автоматизированной системы по частям, конечная цель состоит в том, чтобы возложить на машину как можно больше процедур библиотечного обслуживания. Это означает, что необходимо использовать в автоматизированной библиотечной системе средства автоматического ввода документов и оборудование для работы в режиме реального времени, если эта аппаратура экономична и эффективна.

Режим работы в реальном масштабе времени дает возможность использовать в пределах практической целесообразности все методы и средства, необходимые для создания эффективной поисковой системы, причем платой за это является увеличение объема программирования при разработке системы. Система, разработанная для использования в библиотеке, может оказаться столь же сложной, как любая система, разработанная для каких-то других целей. Идеальным решением проблемы представляется эффективная и легко используемая система для поиска и анализа документов, которая

снабжена емким внешним оборудованием, обеспечивающим контакт, взаимодействие и обратную связь с пользователями; кроме того, желательна организация между машиной и человеком двустороннего диалога, в котором машина либо действует как ведущее лицо, либо обращается к сотрудникам библиотеки за помощью в затруднительных случаях; и, наконец, нужно сочетание скорости, точности и мощности машины с знаниями эксперта и опытом сотрудника библиотеки.

## ПРИЛОЖЕНИЕ А

### О БЭКУСОВСКОЙ НОРМАЛЬНОЙ ФОРМЕ

---

Бэкусовская нормальная форма представляет собой формальный метод описания формата исходных данных для конкретной программы. Чаще всего этот метод используется в связи с языковыми компиляторами в тех случаях, когда операторы языка высокого уровня вводятся в качестве исходных данных для компилятора и их формат контролируется на корректность с целью установления неправильных операторов еще на ранней стадии компиляции, причем процесс проверки сопровождается диагностикой допущенных ошибок. Входящая в состав компилятора специальная подпрограмма редактирования разбивает каждый вводимый оператор на компоненты или основные элементы. Бэкусовская нормальная форма позволяет описывать допустимые структуры операторов и те основные термины, из которых могут составляться эти операторы. Термины определяются последовательно один за другим, причем в каждом определении используются либо основные элементы, либо ранее определенные термины. Лучше всего пояснить сказанное примером.

Допустим, что нам нужна программа, которая вычисляет результат, складывая, вычитая, умножая или деля любые два однозначных целых числа. Другими словами, любой оператор вида  $a +$ ,  $-$ ,  $\times$ , или  $\div b$  допустим, при условии что  $a$  и  $b$  являются однозначными числами. Дадим описание такого рода операторов с помощью бэкусовской нормальной формы. Основными элементами являются символы, представленные на клавиатуре пишущей машинки. В левой части каждого описательного выражения находится определяемый термин, а в правой — компоненты, которыми описывается этот термин (символ | следует читать как «или»):

⟨цифра⟩ ::= 1|2|3|4|5|6|7|8|9|0

⟨операция⟩ ::= +|-|×|÷

⟨арифметическое выражение⟩ ::= ⟨цифра⟩ ⟨операция⟩ ⟨цифра⟩

Если наша программа вводит и выполняет операторы до тех пор, пока не будет прочитан маркер конца @, то допустимая исходная запись описывается с помощью бэкусовской нормальной формы следующим образом:

⟨элемент ввода⟩ ::= (арифметическое выражение) @

Как видно из этого примера, бэкусовская нормальная форма является средством описания элементов ввода таким образом, что каждое описание оказывается очень похожим на те процедуры контроля, что включаются в компилятор. Есть еще одно важное обстоятельство, заключающееся в том, что определения бэкусовской нормальной формы являются рекурсивными. Например, если в указанном выше случае захотели бы допустить не только однозначные числа, но и числа любой длины в качестве операндов, то могли бы определить термин  $\langle \text{цифра} \rangle$ , как сделали это раньше, а затем написать следующее определение:

$$\langle \text{число} \rangle ::= \langle \text{цифра} \rangle \mid \langle \text{цифра} \rangle \langle \text{число} \rangle$$

В этом определении термин  $\langle \text{число} \rangle$  определяется либо как  $\langle \text{цифра} \rangle$ , т. е., как и прежде, в виде однозначного числа, либо как  $\langle \text{цифра} \rangle$ , за которой следует число. Фактически  $\langle \text{число} \rangle$  определяется само через себя, т. е. рекурсивно. Если это определение рассмотреть внимательно, то можно заметить, что  $\langle \text{число} \rangle$  может состоять из  $\langle \text{цифры} \rangle$  или  $\langle \text{цифры} \rangle$ , за которой следует вторая  $\langle \text{цифра} \rangle$ . Тем самым допускаются двузначные числа. Однако если допустить, что  $\langle \text{число} \rangle$  может быть определено как  $\langle \text{цифра} \rangle \langle \text{цифра} \rangle$ , то, заменяя во втором определении термин  $\langle \text{число} \rangle$  парой  $\langle \text{цифра} \rangle \langle \text{цифра} \rangle$ , можно расширить термин  $\langle \text{цифра} \rangle \langle \text{число} \rangle$  до комбинации вида  $\langle \text{цифра} \rangle \langle \text{цифра} \rangle \langle \text{цифра} \rangle$ . Следовательно, трехзначные числа также охватываются этим определением. Продолжая процесс подстановок, можно сколь угодно расширить определение, показав, что в качестве числа допускается последовательность цифр любой длины. Этот простой способ рекурсивного определения элемента через самого себя делает бэкусовскую нормальную форму весьма удобным средством для описания формата, и она, в частности, использована в настоящей работе для определения формата исходных данных для программы РЕДАКТИРОВАНИЕ.

## ПРИЛОЖЕНИЕ Б

### НЕКОТОРЫЕ ДЕЙСТВУЮЩИЕ СИСТЕМЫ

---

В настоящей работе было проведено общее обсуждение вопросов применения вычислительных машин в библиотечном деле. В этом приложении отмечаются некоторые из реализуемых проектов, а также сообщаются сведения о том, где можно получить информацию, касающуюся автоматизации библиотеки.

В США в настоящее время есть несколько вполне оперативных информационных систем, в том числе разработанная Сэлтоном система SMART, которая уже упоминалась нами в связи с использованием в ней нескольких методов поиска. В этой системе допускаются различные способы составления запросов на поиск информации, а также применяется несколько методов вычисления коэффициентов подобия между документами. Пользователь может выбирать из доступных методов наиболее удобный для себя, и результаты, полученные при эксплуатации этой системы, представляются вполне успешными. Другая американская система BOLD характеризуется весьма малым временем поиска, а также удачной организацией коллективного доступа к системе. Эта система допускает несколько «режимов» общения с пользователем, который может, например, просматривать индексный указатель, сортировать информацию, получать на устройствах отображения фрагменты документов и т. д. В «режиме беспорядочного перебора» системе можно задавать тематические вопросы.

В Англии также проявляется значительный интерес к библиотечным приложениям вычислительной техники, хотя имеющиеся проекты не столь обширны, как в Соединенных Штатах. Университетские библиотеки, например такие, как в Ньюкасле, где работает информационно-поисковая система MEDLARS, ориентированная на медицинскую тематику, в Белфасте, в Лафборо и в других городах, основываются на сложных автоматизированных системах, и интерес к такого рода системам очень быстро растет как в плане академическом, так и со стороны местных публичных библиотек.

Помимо известных академических и профессиональных журналов, для получения более полных сведений по вопросам автоматизации библиотек можно воспользоваться следующими двумя весьма солидными источниками:

1. Периодическое издание Program, в котором публикуются материалы о деятельности группы по машинным приложениям ASLIB, учрежденной вместе с Британским обществом по вычислительной технике.
2. Бюллетень новостей Управления по науке и технической информации, выходящий раз в квартал и содержащий сведения о текущих работах; бюллетень публикует представляющие широкий интерес материалы, персоналии и адреса групп, занимающихся разработкой различных проектов.

Растущий интерес к проблеме автоматизации библиотек подтверждается, например, научным исследованием, проводимым в Библиотеке Бодлея в Оксфорде, или недавно начавшимися переговорами между одним из ведущих издательств и крупной фирмой, производящей вычислительные машины; поэтому жизненно необходимой становится скорейшая организация всестороннего обмена идеями между специалистами в области вычислительной техники и ведущими библиотечными работниками.

## СПИСОК ЛИТЕРАТУРЫ

Список литературы включает два раздела, в первом из которых перечислены работы, в основном упоминаемые в тексте, а во втором — материалы для дополнительного чтения и справок. Оба раздела состоят из нескольких условных тематических рубрик. В библиографии приняты следующие сокращения:

*AD* American Documentation

*AP* Proceedings of the Association of Librarians

*CACM* Communications of the Association for Computing Machinery

*CB* Computer Bulletin

*CJ* Computer Journal

*Dat.* Datamation

*IBM JR & D* International Business Machines Corp. Journal of Research and Development

*JACM* Journal of the Association for Computing Machinery

*JIE* Journal of Industrial Engineering

*LR & TS* Library Resources and Technical Services

*TIB* Nordisk Tidskrift för Informationsbehandling

### 1. Цитируемые работы

#### *Книги и общие обзоры:*

- (1) Sharp J. R., Some Fundamentals of Information Retrieval, Deutsch, 1965.
- (2) Boaz, Dr M (ed.), Modern Trends in Documentation, Pergamon Press, 1959.
- (3) Lickliger J.C.R., Libraries of the Future, MIT Press, 1965.
- (4) Markuson B. (ed.), Proceedings of the 1963 conference of Libraries and Automation.
- (5) King G. W. et al., Automation and the Library of Congress, Washington, D.C., 1963.
- (6) Kimber R. T., Computer Applications in the Fields of Library Housekeeping and Information Processing, *Program*, No. 6, July 1967.
- (7) Bar-Hillel Y., Language and Information, Addison-Wesley, 1964.
- (8) Hieber C. E., An Analysis of Questions and Answers in Libraries, m.sc. thesis, Lehigh University.
- (9) Rosenberg V., The Application of Psychometric Techniques to Determine the Attitudes of Individuals Toward Information] Seeking, m.sc. thesis, Lehigh University.

#### *Деятельность библиотек:*

- (10) Report on Serials Computer Project University Library and UCSD Computer Center, University of California, San Diego, Part 1, 1961/2; Part 2, 1964.

- (11) Luhn H. P., Selective Dissemination of New Scientific Information with Aid of Electronic Processing Equipment, *AD*, Apr. 1961.
- (12) The I.B.M. DSD Technical Information Center, I.B.M. Technical Report TR 00.1103, Feb. 1964.

*Аппаратура и оборудование:*

- (13) King G. K., Table Look-up Procedures in Language Processing, Part I The Raw Text, *IBM JR & D*, Apr. 1961.
- (14) Craft J. L., Goldman, E. H., Strohm W. B., A Table Look-up for Processing of Natural Languages, *IBM JR & D*, July 1961.
- (15) Seeber R. R., Lindquist A. B., Associative Memory with Ordered Retrieval, *IBM JR & D*, Jan. 1962.
- (16) Johnson L. R., McAndrew M. H., On Ordered Retrieval from an Associative Memory, *IBM JR & D*, Apr. 1964.
- (17) Paine R. M. Preparation for Optical Character Recognition, *CJ*, Nov. 1966.

*Программное обеспечение, языки и средства обработки файлов:*

- (18) Salton G., Data Manipulation and Programming Problems in Automatic Information Retrieval, *CACM*, Mar. 1966.
- (19) Yngve V. H., An Introduction to COMMIT Programming, MIT Press, 1962.
- (20) Simmons R. F., Answering English Questions by Computer — a Survey, *CACM*, Jan. 1965.
- (21) Weizenbaum J., ELIZA — a Computer Program for the Study of Natural Language Communication between Man and Machine, *CACM*, Jan. 1966.

*Исправление ошибок и кодирование:*

- (22) Carlston C., Techniques for Replacing Characters that are Garbled on Input, *Computer Digest*, Aug. 1966.
- (23) Bourne C. P., Ford D. F., A Study of Methods for Systematically Abbreviating English Words and Names, *JACM*, Oct. 1961.
- (24) Smith B., SPECOL — A Computer Enquiry Language for the Non-programmer, *CJ*, Aug. 1968.

*Методы информационного поиска:*

- (25) Maron M. E., Automatic Indexing — An Experimental Enquiry, *JACM*, July 1961.
- (26) Maron M. E., Kuhns J. L., On Relevance, Probabilistic Indexing, and Information Retrieval, *JACM*, July 1960.
- (27) Borko H., Bernick M., Automatic Document Classification, *JACM*, Apr. 1963.
- (28) Borko H., Bernick M., Automatic Document Classification II — Additional Experiments, *JACM*, Apr. 1964.
- (29) Doyle L. B., Is Automatic Classification a Reasonable Application of Statistical Analysis of Text? *JACM*, Oct. 1965.
- (30) Doyle L. B., Semantic Road Maps for Literature Searchers, *JACM*, Oct. 1961.
- (31) Needham R. M., Parker-Rhodes A. F., Contributions to the Theory of Clumps, I and II.
- (32) Ivie E. L., Search Procedures Based on Measure of Relatedness between Documents, m. sc. thesis, MACTR. 29.

*Действующие системы и разработки:*

- (33) Salton G., The Evaluation of Automatic Retrieval Procedures-Selected Test Results using the SMART System, *AD*, July 1965.  
(34) Salton G., Lesk M. E., The SMART Automatic Document Retrieval System — an Illustration, *CACM*, June 1965.  
(35) Sieburg J., Automatic Abstraction of Legal Information, *Dat.*, Nov. 1966.

## 2. Материалы для дополнительного чтения и справок

*Статьи и книги по общим вопросам:*

- Becker J., Hayes R. M., Information Storage and Retrieval: Tools, Elements, Theories.  
Kent A. (ed.), Information Retrieval and Machine Translation (2 vols).  
Overhage C. F. J., Harman R. J. (ed.), Intrex—Report of a Planning Conference on Information Transfer Experiments, MIT Press, 1966.  
Rubinoff M. (ed.), Toward a National Information System (2nd Annual Colloquium of Information Retrieval), Macmillan, 1966.  
American Federation of Information Processing Societies and International Federation of Information Processing Societies (conference reports).

*Деятельность библиотек:*

- Bourne C. P., The Historical Development and Present State-of-the-Art of Mechanised Information Retrieval Systems, *AD*, Apr. 1961.  
Moss R., How do we Classify? *AP*, Feb. 1962.  
Prywes N. S., Browsing in an Automate Library in Computer Augmentation of Human Reasoning (edited by Sass M. A., Wilkinson, W. D.), Macmillan, 1965.  
Fasana P. J., Automating Cataloging Functions in Conventional Libraries, *LR & TS*, Fall. 1963.  
Perrault J. M., The Computer and Catalog Filing Rules, *LR & TS*, Summer 1965.  
Richmond P. A., Note on Updating and Searching Computerized Catalogs, *LR & TS*, Spring 1966.  
Nicholson N., Thurston W., Serials and Journals in the MIT Library, *AD*, Oct. 1958.  
McCann A., Applications of Machines to Library Techniques: Periodicals, *AD*, Oct. 1961.  
Vdovin G., Voigt M. J., Newman D., Perry C., Computer Processing of Serial Records, *LR & TS*, Winter 1963.  
Srygley T. D., Serials Record Instructions for a Computerised Serials System, *LR & TS*, Summer 1964.  
Hammer D. P., Reflections on the Development of an Automated Serials System, *LR & TS*, Spring 1965.  
Joyce T., Heedham R. M., The Thesaurus Approach to Information Retrieval, *AD*, July 1958.  
Holm B. E., Rasmusson L. E., The Development of a Technical Thesaurus, *AD*, July 1961.  
Cox N. S. M., Grose M. W. (ed.), Organisation and Handling of Bibliographic Records by Computer, Oriel Press, 1967.  
Enright B. J., Hanson D. G., An Experimental Periodicals Checking List, *Program*, Oct. 1967.  
Maidment W. R., Computer Methods in Public Libraries, *Program*, Apr. 1968.  
Kimber R. T., An Operational Computerised Circulation System with On-line Interrogation Capability, *Program*, Oct. 1968.

*Программное обеспечение и средства обработки файлов:*

- Papers on Design, Implementation and Application of Information Retrieval Oriented Languages, *CACM*, Jan. 1962.
- Canter J. D., Donaghy C. E. UPLIFTS — University of Pittsburgh Linear File Tandem System, *CACM*, Sept. 1965.
- Prywes N. S., Gray H. J., The Multi-list System for Real Time Storage and Retrieval, IFIPS Congress Report, 1962.
- Prywes N. S., A Storage and Retrieval System for Real Time Problem Solving.
- Langefors B., Information Retrieval in File Processing I and II, *TIB*, 1961.
- Fredkin E., Tree Memory, *CACM*, Sept. 1960.
- Salton G., Manipulation of Trees in Information Retrieval, *CACM*, Feb. 1962.
- Scidmore A. K., Weinberg B. L., Storage and Search Properties of a Tree-Organised Memory System, *CACM*, Jan. 1963.
- Sussenguth E. H. Jr., Use of Tree Structures for Processing Files, *CACM*, May 1963.
- Gill Prof. S., Why Real Time Computing is Different, *CB*, Jan. 1966.
- Colin A. J. T., A simple program for use in the «Conversational Mode», *CJ*, Nov. 1966.
- Cooper W. S., Fact Retrieval and Deductive Question-Answering Information Retrieval Systems, *JACM*, Apr. 1964.

*Исправление ошибок и кодирование:*

- Damerou F. J., A Technique for Computer Detection and Correction of Spelling Errors, *CACM*, Mar. 1964.
- Thorelli L. E. Automatic Correction of Errors in Text, *TIB* 2.1, 1962.
- Davidson L., Retrieval of Misspelled Names in an Airlines Passenger Record System, *CACM*, Mar. 1962.
- Brace D. A., Direct Coding of English Language Names, *CJ*, July 1963.
- Klein S., Simmons R., A Computational Approach to Grammatical Coding of English Words, *JACM*, July 1963.
- Milliman G., Alameda County's «People Information System», *Dat.*, Mar. 1967.
- Jackson M., Mnemonics, *Dat.*, Apr. 1967.

*Методы информационного поиска:*

- Salton G., Associative Document Retrieval Techniques Using Bibliographical Information, *JACM*, Oct. 1963.
- Luhn H. P., A Statistical Approach to Mechanised Encoding and Searching of Literary Information, *IBM JR & D*, Oct. 1957.
- Edmundson H. P., Wyllys, R. E., Automatic Abstracting and Indexing—Survey and Recommendations, *CACM*, May 1961.
- Stiles H. E., The Association Factor in Information Retrieval, *JACM*, Apr. 1961.
- Simmons R. F., Storage and Retrieval Aspects of Meaning in Directed Graph Structures, *CACM*, Mar. 1966.
- O'Connor J., Mechanical Indexing Methods and Their Testing, *JACM*, Oct. 1964.
- O'Connor J., Automatic Subject Recognition in Scientific Papers: an Empirical Study, *JACM*, Oct. 1965.
- Baker F. B., Information Retrieval on Latent Class Analysis, *JACM*, Oct. 1962.
- Dale A. C., Dale N., Some Clumping Experiments for Associative Document Retrieval, *AD*, Jan. 1965.

*Действующие системы и разработки:*

- Drew D. L., Summit R. K., Tanaka R. I., Whiteley R. B., An On-line Technical Library Reference Retrieval System, *AD*, Jan. 1966.
- Holzman A. G., Design of a Large Scale Information Retrieval System to Transfer Technology from Space to Industry, *JIE*, Nov. 1966.

## ОГЛАВЛЕНИЕ

Из предисловия к первому изданию . . . . .	5
Предисловие ко второму изданию . . . . .	6
1. Зачем нужна автоматизированная библиотечная система . . . . .	7
1.1. Проблематика обработки информации . . . . .	7
1.2. Изменение взглядов . . . . .	9
2. Предварительные сведения о вычислительных машинах . . . . .	10
2.1. Основные принципы . . . . .	10
2.2. Внешние носители информации . . . . .	13
2.3. Некоторые направления развития вычислительной техники . . . . .	17
3. Что нужно библиотечной системе . . . . .	22
3.1. Подход к вычислительной системе . . . . .	22
3.2. Целевое назначение системы . . . . .	24
3.3. Требования библиотечного персонала . . . . .	26
3.4. Требования библиотечных пользователей . . . . .	30
4. Аппаратное оборудование вычислительной машины . . . . .	33
4.1. Общие соображения . . . . .	33
4.2. Терминалы и клавиатуры . . . . .	33
4.3. Основная память . . . . .	36
4.4. Вспомогательные запоминающие устройства . . . . .	36
4.5. Другое оборудование . . . . .	37
5. Программное обеспечение и программирование . . . . .	39
5.1. Стандартное программное обеспечение . . . . .	39
5.2. Управление терминалами . . . . .	39
5.3. Командные языки . . . . .	40
5.4. Поисковый язык . . . . .	41
5.5. Естественный язык . . . . .	43
5.6. Аббревиатура . . . . .	45
5.7. Исключение ошибок . . . . .	46
6. Структура файлов . . . . .	48
6.1. Внешняя структура файлов . . . . .	48
6.2. Методы доступа . . . . .	49
6.3. Внутренняя структура файлов . . . . .	51
6.3.1. Последовательные файлы . . . . .	51
6.3.2. Списковые файлы . . . . .	52
6.3.3. Древоподобные файлы . . . . .	53
7. Информационный поиск . . . . .	56
7.1. Основные критерии информационного поиска . . . . .	56
7.2. Логические принципы запоминания и поиска информации . . . . .	57
7.3. Автоматическое сопоставление . . . . .	64
7.4. Автоматическая каталогизация . . . . .	68
7.5. Автоматическая классификация . . . . .	72
8. Некоторые примеры программной реализации . . . . .	77
8.1. Введение . . . . .	77
8.2. Программа РЕДАКТИРОВАНИЕ . . . . .	79
8.3. Программа ПОИСК . . . . .	86
9. Общие принципы автоматизированной информационной системы . . . . .	102
9.1. Факты или документы? . . . . .	102
9.2. Количественные оценки . . . . .	105
9.3. Ответственность за систему . . . . .	107
Приложение А. О бэкуповской нормальной форме . . . . .	111
Приложение Б. Некоторые действующие системы . . . . .	113
Список литературы . . . . .	115

### *УВАЖАЕМЫЙ ЧИТАТЕЛЬ!*

Ваши замечания о содержании книги, ее оформлении, качестве перевода и другие просим присылать по адресу:

129820, Москва, И-110, ГСП,  
1-й Рижский пер., д. 2,  
издательство «Мир».

Дж. Хенли

#### АВТОМАТИЗИРОВАННАЯ БИБЛИОТЕКА И ИНФОРМАЦИОННЫЕ СИСТЕМЫ

Редактор Л. Н. Бабынина  
Художник В. М. Новоселова  
Художественный редактор В. И. Шаповалов  
Технический редактор А. Д. Хомяков  
Корректор Т. С. Лаврова

Сдано в набор 19/IV 1974 г. Подписано к печати 4/IX 1974 г. Бумага книжн.-журн.  
60×90<sup>1/16</sup> = 3,75 бум. л. 7,50 усл. печ. л. Уч.-изд. л. 7,03. Изд. № 1/7510 Цена 50 коп.  
Зак. 0326

ИЗДАТЕЛЬСТВО «МИР», Москва, 1-й Рижский пер., 2

Ордена Трудового Красного Знамени Московская типография № 7 «Искра революции»  
Союзполиграфпрома при Государственном комитете Совета Министров СССР по делам  
издательств, полиграфии и книжной торговли  
Москва, К-1, Трехпрудный пер., 9

50 н.

МАТЕМАТИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ЭВМ

