## § 4. КОДИРОВАНИЕ ИНФОРМАЦИИ.

### 4.1. АБСТРАКТНЫЙ АЛФАВИТ

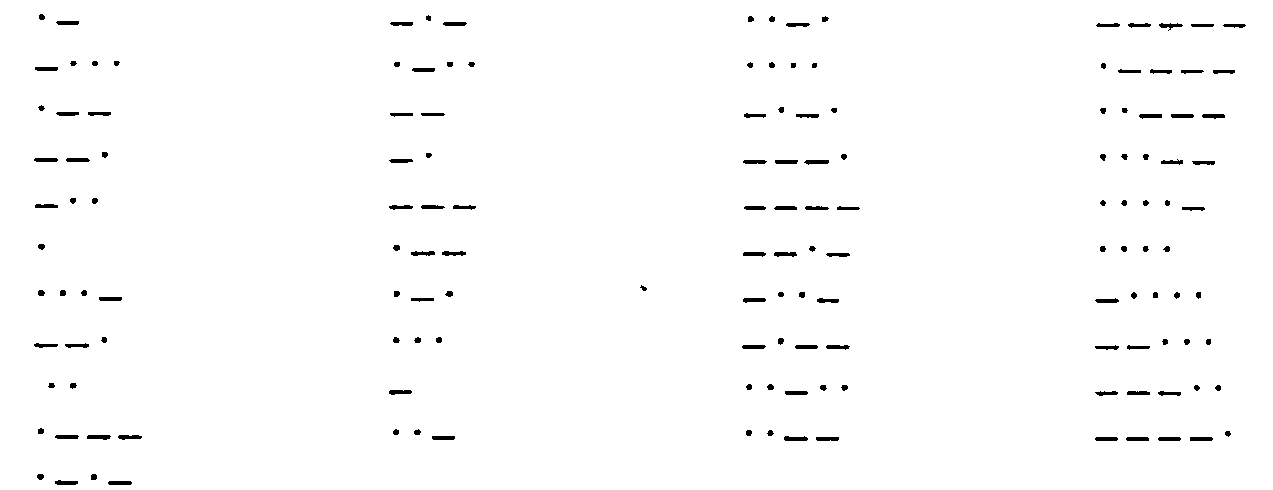
Информация передается в виде сообщений. Дискретная информация записывается с помощью некоторого конечного набора знаков, которые будем называть буквами, не вкладывая в это слово привычного ограниченного значения (типа «русские буквы» или «латинские буквы»). Буква в данном расширенном понимании - любой из знаков, которые некоторым соглашением установлены для общения. Например, при привычной передаче сообщений на русском языке такими знаками будут русские буквы - прописные и строчные, знаки препинания, пробел; если в тексте есть числа - то и цифры. Вообще, буквой будем называть элемент некоторого конечного множества (набора) отличных друг от друга знаков. Множество знаков, в котором определен их порядок, назовем алфавитом (общеизвестен порядок знаков в русском алфавите: А, Б,..., Я).

Рассмотрим некоторые примеры алфавитов.

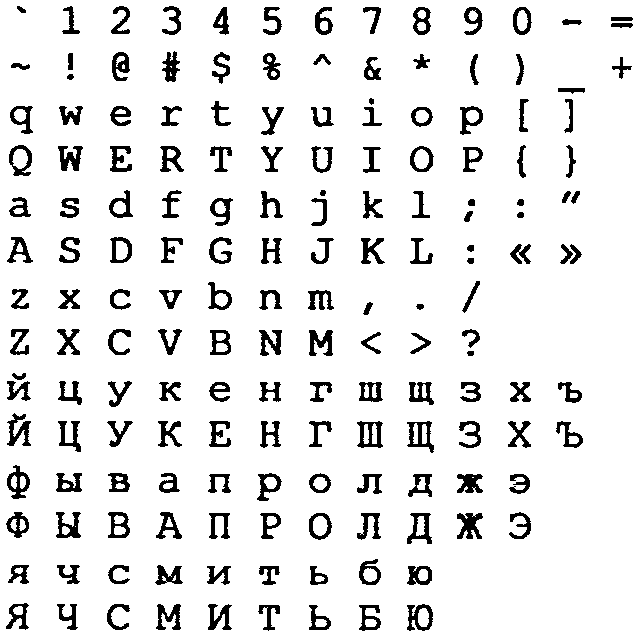
1, Алфавит прописных русских букв:

А Б В Г Д Е Е Ж З И Й К Л М Н О П Р С Т У Ф Х Ц Ч Ш Щ Ъ Ы Ь Э Ю Я

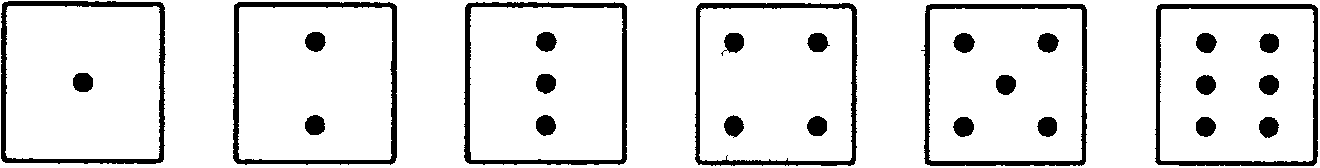
2. Алфавит Морзе:



3. Алфавит клавиатурных символов ПЭВМ IBM (русифицированная клавиатура):



4. Алфавит знаков правильной шестигранной игральной кости:



5. Алфавит арабских цифр:

0123456789

6. Алфавит шестнадцатиричных цифр:

0123456789ABCDEF

Этот пример, в частности, показывает, что знаки одного алфавита могут образовываться из знаков других алфавитов.

7. Алфавит двоичных цифр:

0 1

Алфавит 7 является одним из примеров, так называемых, «двоичных» алфавитов, т.е. алфавитов, состоящих из двух знаков. Другими примерами являются двоичные алфавиты 8 и 9:

8. Двоичный алфавит «точка, «тире»:. \_

9. Двоичный алфавит «плюс», «минус»: + -

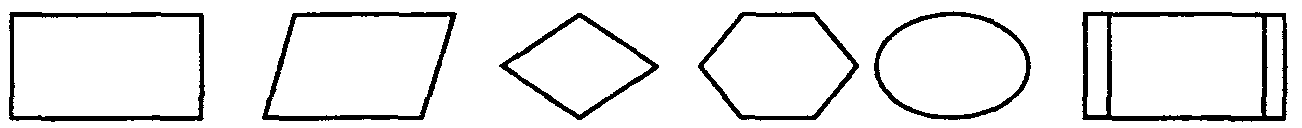
10. Алфавит прописных латинских букв:

ABCDEFGHIJKLMNOPQRSTUVWXYZ

11. Алфавит римской системы счисления:

I V Х L С D М

12. Алфавит языка блок-схем изображения алгоритмов:



13. Алфавит языка программирования Паскаль (см. в главе 3).

### 4.2. КОДИРОВАНИЕ И ДЕКОДИРОВАНИЕ

В канале связи сообщение, составленное из символов (букв) одного алфавита, может преобразовываться в сообщение из символов (букв) другого алфавита. Правило, описывающее однозначное соответствие букв алфавитов при таком преобразовании, называют кодом. Саму процедуру преобразования сообщения называют перекодировкой. Подобное преобразование сообщения может осуществляться в момент поступления сообщения от источника в канал связи (кодирование) и в момент приема сообщения получателем (декодирование). Устройства, обеспечивающие кодирование и декодирование, будем называть соответственно кодировщиком и декодировщиком. На рис. 1.5 приведена схема, иллюстрирующая процесс передачи сообщения в случае перекодировки, а также воздействия помех (см. следующий пункт).

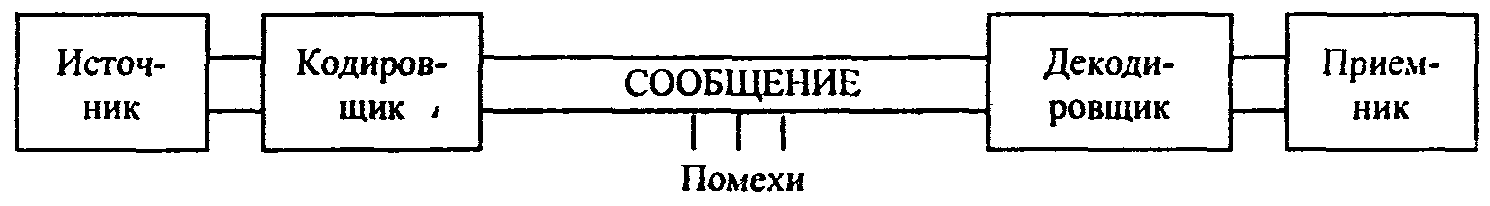
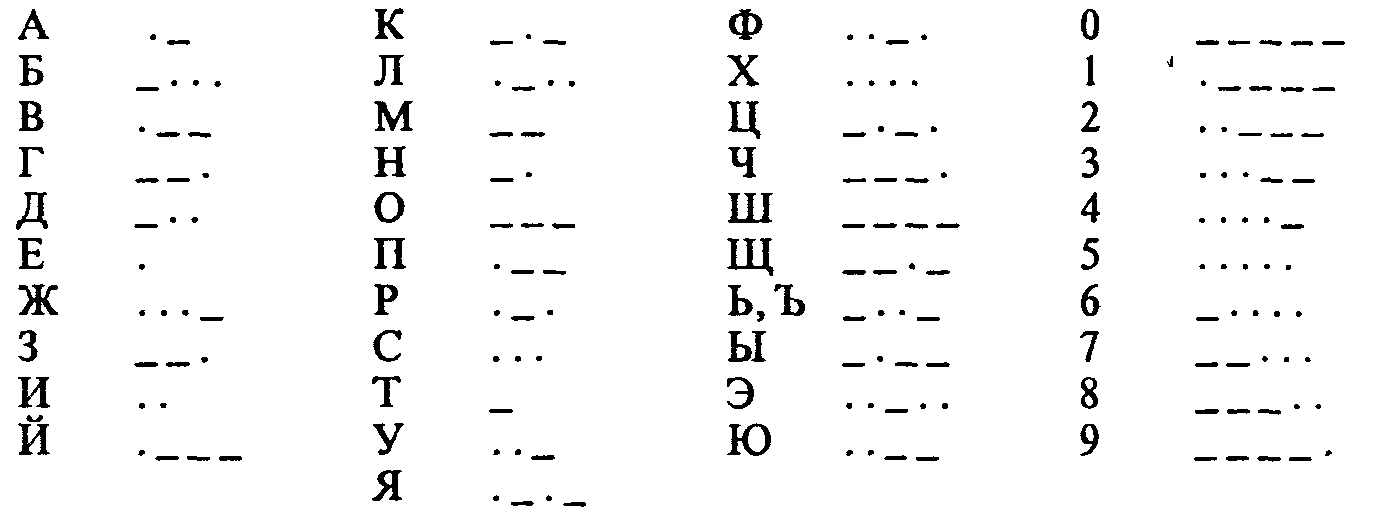


Рис. 1.5. Процесс передачи сообщения от источника к приемнику

Рассмотрим некоторые примеры кодов.

1. Азбука Морзе в русском варианте (алфавиту, составленному из алфавита русских заглавных букв и алфавита арабских цифр ставится в соответствие алфавит Морзе):



2. Код Трисиме (знакам латинского алфавита ставятся в соответствие комбинации из трех знаков: 1,2,3):

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| А | 111 | D | 121 | G | 131 | J211 | M221 | P231 | S311 | V321 | Y331 |
| В | 112 | E | 122 | H | 132 | K212 | N222 | Q232 | T312 | W322 | Z332 |
| С | 113 | F | 123 | I | 133 | L213 | О223 | R233 | U313 | X323 | .333 |

Код Трисиме является примером, так называемого, равномерного кода (такого, в котором все кодовые комбинации содержат одинаковое число знаков - в данном случае три). Пример неравномерного кода - азбука Морзе.

3. Кодирование чисел знаками различных систем счисления см. §3.

### 4.3. ПОНЯТИЕ О ТЕОРЕМАХ ШЕННОНА

Ранее отмечалось, что при передаче сообщений по каналам связи могут возникать помехи, способные привести к искажению принимаемых знаков. Так, например, если вы попытаетесь в ветреную погоду передать речевое сообщению человеку, находящемуся от вас на значительном расстоянии, то оно может быть сильно искажено такой помехой, как ветер. Вообще, передача сообщений при наличии помех является серьезной теоретической и практической задачей. Ее значимость возрастает в связи с повсеместным внедрением компьютерных телекоммуникаций, в которых помехи неизбежны. При работе с кодированной информацией, искажаемой помехами, можно выделить следующие основные проблемы: установления самого факта того, что произошло искажение информации; выяснения того, в каком конкретно месте передаваемого текста это произошло; исправления ошибки, хотя бы с некоторой степенью достоверности.

Помехи в передачи информации - вполне обычное дело во всех сферах профессиональной деятельности и в быту. Один из примеров был приведен выше, другие примеры - разговор по телефону, в трубке которого «трещит», вождение автомобиля в тумане и т.д. Чаще всего человек вполне справляется с каждой из указанных выше задач, хотя и не всегда отдает себе отчет, как он это делает (т.е. неалгоритмически, а исходя из каких-то ассоциативных связей). Известно, что естественный язык обладает большой **избыточностью** (в европейских языках - до 7%), чем объясняется большая помехоустойчивость сообщений, составленных из знаков алфавитов таких языков. Примером, иллюстрирующим устойчивость русского языка к помехам, может служить предложение «в словох всо глосноо зомононо боквой о». Здесь 26% символов «поражены», однако это не приводит к потере смысла. Таким образом, в данном случае избыточность является полезным свойством.

Избыточность могла бы быть использована и при передаче кодированных сообщений в технических системах. Например, каждый фрагмент текста («предложение») передается трижды, и верным считается та пара фрагментов, которая полностью совпала. Однако, большая избыточность приводит к большим временным затратам при передаче информации и требует большого объема памяти при ее хранении. Впервые теоретическое исследование эффективного кодирования предпринял К.Шеннон.

**Первая теорема** Шеннона декларирует возможность создания системы эффективного кодирования дискретных сообщений, у которой среднее число двоичных символов на один символ сообщения асимптотически стремится к энтропии источника сообщений (в отсутствии помех).

Задача эффективного кодирования описывается триадой:

*Х = {X* 4*i*} - кодирующее устройство - *В.*

Здесь *X, В -* соответственно входной и выходной алфавит. Под множеством *хi* можно понимать любые знаки (буквы, слова, предложения). *В -* множество, число элементов которого в случае кодирования знаков числами определяется основанием системы счисления (например, *т* = 2). Кодирующее устройство сопоставляет каждому сообщению *хi* из *Х* кодовую комбинацию, составленную из *пi* символов множества *В.* Ограничением данной задачи является отсутствие помех. Требуется оценить минимальную среднюю длину кодовой комбинации.

Для решения данной задачи должна быть известна вероятность *Рi* появления сообщения *хi,* которому соответствует определенное количество символов *пi* алфавита *В.* Тогда математическое ожидание количества символов из *В* определится следующим образом:

*n cр* = *пiРi* (средняя величина).

Этому среднему числу символов алфавита *В* соответствует максимальная энтропия *Нтаx* = *n*ср log *т.* Для обеспечения передачи информации, содержащейся в сообщениях *Х* кодовыми комбинациями из *В,* должно выполняться условие H4mах ≥ *Н(х),* или *пcр* log *т* ≥ *- Рi* log *Рi.* В этом случае закодированное сообщение имеет избыточность *пcр* ≥ *H(x) /* log *т,* *n*min = *H(x) /* log *т.*

Коэффициент избыточности

*К*u = (*H*max – *H*(*x*)) / *H*max = (*n*cp – *n*min) / *n*cp

Выпишем эти значения в виде табл. 1.8. Имеем:

*N*min = *H*(*x*) / log*2* = 2,85, *K*u *=* (2,92 - 2,85) / 2,92 = 0,024,

т.е. код практически не имеет избыточности. Видно, что среднее число двоичных символов стремится к энтропии источника сообщений.

**Таблица 1.8 Пример к первой теореме Шеннона**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *N* | *Рхi* | *xi* | Код | *ni* | *пi-* ∙ *Рi* | *Рхi* ∙ log *Рхi* |
| 1 | 0,19 | X1 | 10 | 2 | 0,38 | -4,5522 |
| 2 | 0,16 | X2 | 001 | 3 | 0,48 | -4,2301 |
| 3 | 0.16 | X3 | 011 | 3 | 0,48 | -4,2301 |
| 4 | 0,15 | X4 | 101 | 3 | 0,45 | -4,1054 |
| 5 | 0,12 | X5 | 111 | 3 | 0,36 | -3,6706 |
| 6 | 0,11 | X6 | 111 | 3 | 0,33 | - 3,5028 |
| 7 | 0,09 | X7 | 1011 | 4 | 0,36 | -3,1265 |
| 8 | 0,02 | X8 | 1001 | 4 | 0,08 | -3,1288 |
|  | Σ=1 | Σ=2,92 | | | | Σ=2,85 |

**Вторая теорема Шеннона** гласит, что при наличии помех в канале всегда можно найти такую систему кодирования, при которой сообщения будут переданы с заданной достоверностью. При наличии ограничения пропускная способность канала должна превышать производительность источника сообщений. Таким образом, вторая теорема Шеннона устанавливает принципы помехоустойчивого кодирования. Для дискретного канала с помехами теорема утверждает, что, если скорость создания сообщений меньше или равна пропускной способности канала, то существует код, обеспечивающий передачу со сколь угодно мглой частотой ошибок.

Доказательство теоремы основывается на следующих рассуждениях. Первоначально последовательность *Х* = *{xi}* кодируется символами из *В* так, что достигается максимальная пропускная способность (канал не имеет помех). Затем в последовательность из *В* длины *п* вводится *r* символов и по каналу передается новая последовательность из *п + r* символов. Число возможных последовательностей длины и + *т* больше числа возможных последовательностей длины *п.* Множество всех последовательностей длины *п* + *r* может быть разбито на *п* подмножеств, каждому из которых сопоставлена одна из последовательностей длины *п.* При наличии помехи на последовательность из *п* + *r* выводит ее из соответствующего подмножества с вероятностью сколь угодно малой.

Это позволяет определять на приемной стороне канала, какому подмножеству принадлежит искаженная помехами принятая последовательность длины *п + r,* и тем самым восстановить исходную последовательность длины *п.*

Эта теорема не дает конкретного метода построения кода, но указывает на пределы достижимого в создании помехоустойчивых кодов, стимулирует поиск новых путей решения этой проблемы.

### 4.4. МЕЖДУНАРОДНЫЕ СИСТЕМЫ БАЙТОВОГО КОДИРОВАНИЯ

Информатика и ее приложения интернациональны. Это связано как с объективными потребностями человечества в единых правилах и законах хранения, передачи и обработки информации, так и с тем, что в этой сфере деятельности (особенно в ее прикладной части) заметен приоритет одной страны, которая благодаря этому получает возможность «диктовать моду».

Компьютер считают универсальным преобразователем информации. Тексты на естественных языках и числа, математические и специальные символы - одним словом все, что в быту или в профессиональной деятельности может быть необходимо человеку, должно иметь возможность быть введенным в компьютер.

В силу безусловного приоритета двоичной системы счисления при внутреннем представлении информации в компьютере кодирование «внешних» символов основывается на сопоставлении каждому из них определенной группы двоичных знаков. При этом из технических соображений и из соображений удобства кодирования-декодирования следует пользоваться равномерными кодами, т.е. двоичными группами равной длины.

Попробуем подсчитать наиболее короткую длину такой комбинации с точки зрения человека, заинтересованного в использовании лишь одного естественного алфавита - скажем, английского: 26 букв следует умножить на 2 (прописные и строчные) - итого 52; 10 цифр, будем считать, 10 знаков препинания; 10 разделительных знаков (три вида скобок, пробел и др.), знаки привычных математических действий, несколько специальных символов (типа #, $, & и др.) — итого ~ 100. Точный подсчет здесь не нужен, поскольку нам предстоит решить простейшую задачу: имея, скажем, равномерный код из групп по *N* двоичных знаков, сколько можно образовать разных кодовых комбинаций. Ответ очевиден *К* = 2N. Итак, при *N* = 6 *К* = 64 - явно мало, при *N* = 7 *К* = 128 - вполне достаточно.

Однако, для кодирования нескольких (хотя бы двух) естественных алфавитов (плюс все отмеченные выше знаки) и этого недостаточно. Минимально достаточное значение *N* в этом случае 8; имея 256 комбинаций двоичных символов, вполне можно решить указанную задачу. Поскольку 8 двоичных символов составляют 1 байт, то говорят о системах «байтового» кодирования.

Наиболее распространены две такие системы: EBCDIC (Extended Binary Coded Decimal Interchange Code) и ASCII (American Standard Information Interchange).

Первая - исторически тяготеет к «большим» машинам, вторая чаще используется на мини- и микро-ЭВМ (включая персональные компьютеры). Ознакомимся подробнее именно с ASCII, созданной в 1963 г.

В своей первоначальной версии это - система семибитного кодирования. Она ограничивалась одним естественным алфавитом (английским), цифрами и набором различных символов, включая «символы пишущей машинки» (привычные знаки препинания, знаки математических действий и др.) и «управляющие символы». Примеры последних легко найти на клавиатуре компьютера: для микро-ЭВМ, например, DEL - знак удаления символа.

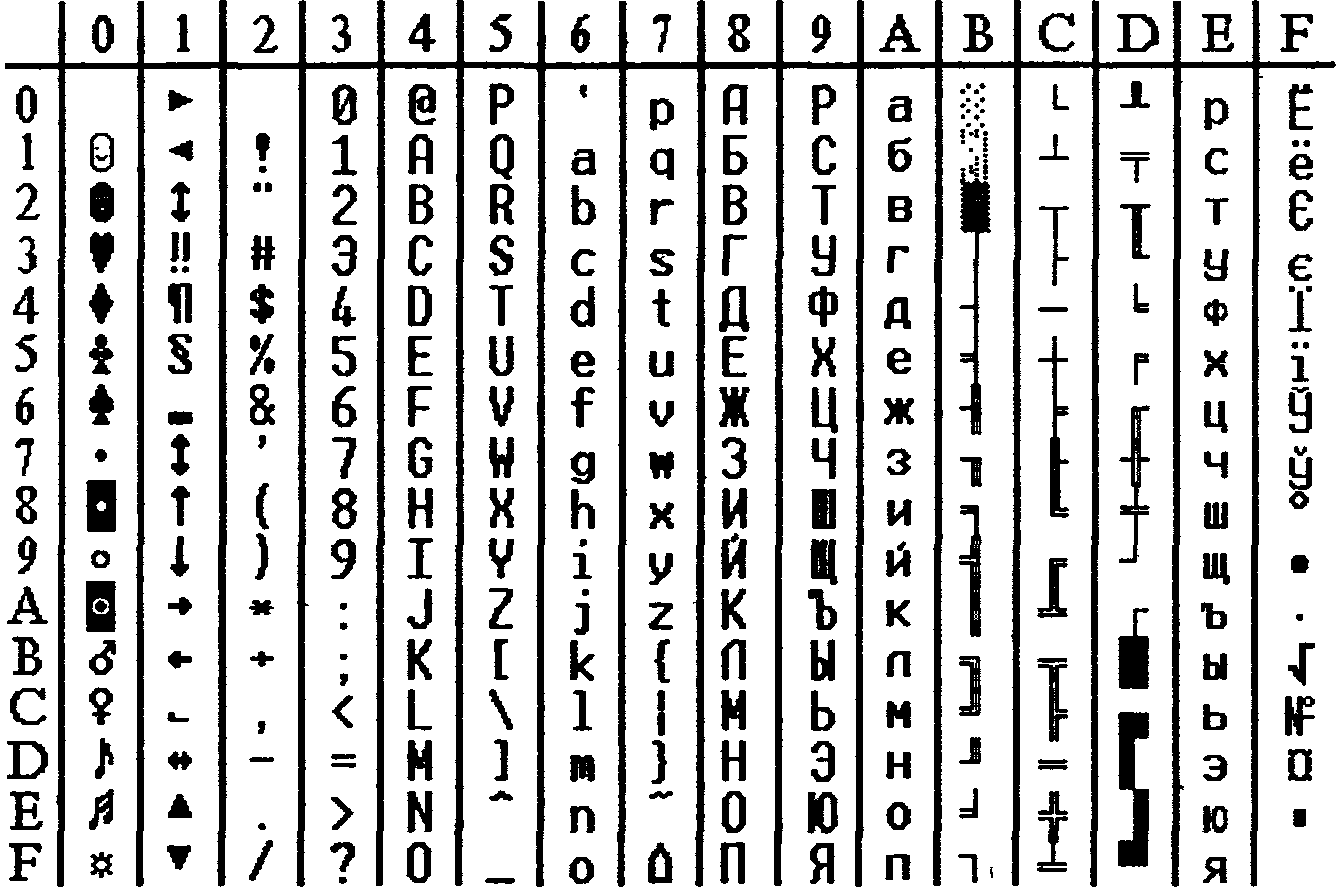
В следующей версии фирма IBM перешла на расширенную 8-битную кодировку. В ней первые 128 символов совпадают с исходными и имеют коды со старшим битом равным нулю, а остальные коды отданы под буквы некоторых европейских языков, в основе которых лежит латиница, греческие буквы, математические символы (скажем, знак квадратного корня) и символы псевдографики. С помощью последних можно создавать таблицы, несложные схемы и др.

Для представления букв русского языка (кириллицы) в рамках ASCII было предложено несколько версий. Первоначально был разработан ГОСТ под названием КОИ-7, оказавшийся по ряду причин крайне неудачным; ныне он практически не используется.

В табл. 1.9 приведена часто используемая в нашей стране модифицированная альтернативная кодировка. В левую часть входят исходные коды ASCII; в правую часть (расширение ASCII) вставлены буквы кириллицы взамен букв, немецкого, французского алфавитов (не совпадающих по написанию с английскими), греческих букв, некоторых спецсимволов.

Знакам алфавита ПЭВМ ставятся в соответствие шестнадцатиричные числа по правилу: первая - номер столбца, вторая - номер строки. Например: английская 'А' - код 41, русская 'и' - код А8.

Таблица 1.9 Таблица кодов ASCII (расширенная)



Одним из достоинств этой системы кодировки русских букв является их естественное упорядочение, т.е. номера букв следуют друг за другом в том же порядке, в каком сами буквы стоят в русском алфавите. Это очень существенно при решении ряда задач обработки текстов, когда требуется выполнить или использовать лексикографическое упорядочение слов.

Из сказанного выше следует, что даже 8-битная кодировка недостаточна для кодирования всех символов, которые хотелось бы иметь в расширенном алфавите. Все препятствия могут быть сняты при переходе на 16-битную кодировку Unicode, допускающую 65536 кодовых комбинаций.