

Содержание:

ВВЕДЕНИЕ

Актуальность темы. К концу XX в. стала складываться, сначала в рамках кибернетики, а затем информатики, информационная картина мира. Строение и функционирование сложных систем различной природы (биологических, социальных, технических) оказалось невозможным объяснить, не рассматривая общих закономерностей информационных процессов.

Автоматизированная система обработки информации (АСОИ) представляет собой совокупность информации, экономико-математических методов и моделей, технических, программных, технологических средств и штата специалистов, предназначенную для обработки информации и принятия управленческих решений

По мере развития и усложнения средств, методов и форм автоматизации процессов обработки информации повышается зависимость общества от степени безопасности используемых им информационных технологий, от которых порой зависит благополучие, а иногда и жизнь многих людей.

Здесь необходимо введение каких-либо мер по защите информации от несанкционированного доступа к ним, такой мерой можно считать кодирование. Во всех его формах и видах представления.

Целью исследования курсовой работы является изучение методов кодирования данных.

В связи с поставленной целью курсовой работы, были определенными следующие **задачи:**

1. Рассмотреть основные определения кодирования. Алфавит кодирования;
2. Изучить способ кодирования информации с помощью систем счисления;
3. Охарактеризовать способы кодирования текстовой информации;
4. Выявить способы кодирования целых и действительных чисел;
5. Рассмотреть кодирование графических данных;
6. Проследить особенности кодирования звуковой информации.

Объектом исследования курсовой работы является система кодирования данных.

Предметом исследования курсовой работы являются методы и приемы кодирования данных.

Структура курсовой работы. Курсовая работа состоит из введения, двух глав, разделенных на параграфы, заключения и списка литературы.

ГЛАВА 1. СУЩНОСТЬ КОДИРОВАНИЯ ИНФОРМАЦИИ

1.1. Основные определения кодирования. Алфавит кодирования

Для автоматизации работы с данными, относящимися к различным типам очень важно унифицировать их форму представления. Для этого обычно используется приём кодирования, т.е. выражение данных одного типа через данные другого типа.

Существует множество понятий и определений «кода». Но мы в своей работе будем опираться на следующее: «Код (code) – это совокупность знаков, символов и правил представления информации».[10, с.40]

В частности можно различать двоичный и троичный код. Алфавит первого ограничен двумя символами (0, 1), а второго – тремя символами (-1, 0, +1). Сигналы, реализующие коды, обладают одной из следующих характеристик:

- ○ униполярный код (значения сигнала равны 0, +1, либо 0, -1);
- полярный код (значения сигнала равны +1, -1);
- биполярный код (значения сигнала равны 0, +1, -1).

Кодируемые элементы входного алфавита обычно называют символами. Символом, как правило, является цифра, буква, знак пунктуации или иероглиф естественного языка, знак препинания, знак пробела, специальный знак, символ операции. Кроме того, учитываются управляющие («непечатные») символы.

Кодирующие (обозначающие) элементы выходного алфавита называются знаками; количество различных знаков в выходном алфавите назовем значностью; количество знаков в кодирующей последовательности для одного символа –

разрядностью кода.

Последовательным является такой код, в котором знаки следуют один за другим во времени. Параллельным – тот, в котором знаки передаются одновременно, образуя символ. [10, с.41]

Естественные человеческие языки – системы кодирования понятий для выражения мыслей посредством речи. К языкам близко примыкают азбуки – системы кодирования компонентов языка с помощью графических символов.

Естественные языки обладают большой избыточностью для экономии памяти, объем которой ограничен, имеет смысл ликвидировать избыточность текста или уплотнить текст.

Существуют несколько способов уплотнения текста.

1. *Переход от естественных обозначений к более компактным.* Этот способ применяется для сжатия записи дат, номеров изделий, уличных адресов и т.д. Идея способа показана на примере сжатия записи даты. Обычно мы записываем дату в виде 10. 05. 01. , что требует 6 байтов памяти ЭВМ. Однако ясно, что для представления дня достаточно 5 битов, месяца - 4, года - не более 7, т.е. вся дата может быть записана в 16 битах или в 2-х байтах.

2. *Подавление повторяющихся символов.* В различных информационных текстах часто встречаются цепочки повторяющихся символов, например пробелы или нули в числовых полях. Если имеется группа повторяющихся символов длиной более 3, то ее длину можно сократить до трех символов. Сжатая таким образом группа повторяющихся символов представляет собой триграф $S P N$, в котором S – символ повторения; P – признак повторения; N - количество символов повторения, закодированных в триграфе. В других схемах подавления повторяющихся символов используют особенность кодов ДКОИ, КОИ- 7, КОИ-8 , заключающуюся в том, что большинство допустимых в них битовых комбинаций не используется для представления символьных данных.

3. *Кодирование часто используемых элементов данных.* Этот способ уплотнения данных также основан на употреблении неиспользуемых комбинаций кода ДКОИ. Для кодирования, например, имен людей можно использовать комбинации из двух байтов диграф PN , где P – признак кодирования имени, N – номер имени. Таким образом может быть закодировано 256 имен людей, чего обычно бывает достаточно в информационных системах. Другой способ основан на отыскании в

текстах наиболее часто встречающихся сочетания букв и даже слов и замене их на неиспользуемые байты кода ДКОИ.

4. *Посимвольное кодирование.* Семибитовые и восьмибитовые коды не обеспечивают достаточно компактного кодирования символьной информации. Более пригодными для этой цели являются 5 - битовые коды, например международный телеграфный код МГК-2. Перевод информации в код МГК-2 возможен с помощью программного перекодирования или с использованием специальных элементов на основе больших интегральных схем (БИС). Пропускная способность каналов связи при передаче алфавитно-цифровой информации в коде МГК-2 повышается по сравнению с использованием восьмибитовых кодов почти на 40%.

5. *Коды переменной длины.* Коды с переменным числом битов на символ позволяют добиться еще более плотной упаковки данных. Метод заключается в том, что часто используемые символы кодируются короткими кодами, а символы с низкой частотой использования - длинными кодами. Идея такого кодирования была впервые высказана Хаффманом, и соответствующий код называется кодом Хаффмана. Использование кодов Хаффмана позволяет достичь сокращения исходного текста почти на 80%.

Использование различных методов уплотнения текстов кроме своего основного назначения - уменьшения информационной избыточности - обеспечивает определенную криптографическую обработку информации. Однако наибольшего эффекта можно достичь при совместном использовании как методов шифрования, так и методов кодирования информации [6, с.95].

Надежность защиты информации может быть оценена временем, которое требуется на расшифрование (разгадывание) информации и определение ключей.

Если информация зашифрована с помощью простой подстановки, то расшифровать ее можно было бы, определив частоты появления каждой буквы в зашифрованном тексте и сравнив их с частотами букв русского алфавита. Таким образом определяется подстановочный алфавит и расшифровывается текст.

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 передаче сообщений на русском языке такими знаками будут русские буквы. Вообще, буквой будем называть элемент некоторого конечного множества (набора) отличных друг от друга знаков. Множество знаков, в котором определен их порядок, назовем алфавитом.[18, с.86]

Рассмотрим некоторые примеры алфавитов.

1. Алфавит прописных русских букв:

А Б В Г Д Е Ё 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

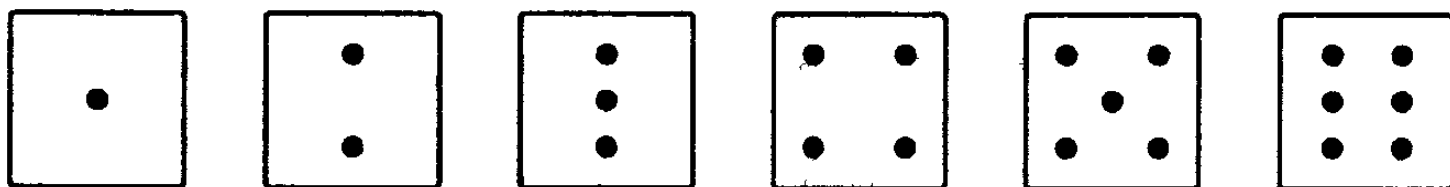
2. Алфавит Морзе:

• —	— • —	• • — •	— — — —
— • • •	• — • •	• • • •	• — — —
• — —	— —	— • — •	• • — —
— — •	— •	— — — •	• • • —
— • •	— — —	— — — —	• • • • —
•	• — — —	— — • —	• • • •
• • • —	• — •	— • • —	— • • • •
— — •	• • •	— • — —	— — • • •
• •	—	• • — • •	— — — • •
• — — —	• • —	• • — —	— — — — •
• — • —			

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

`	1	2	3	4	5	6	7	8	9	0	-	=
~	!	@	#	\$	%	^	&	*	()	_	+
q	w	e	r	t	y	u	i	o	p	[]	
Q	W	E	R	T	Y	U	I	O	P	{	}	
a	s	d	f	g	h	j	k	l	;	:	"	
A	S	D	F	G	H	J	K	L	:	«	»	
z	x	c	v	b	n	m	,	.	/			
Z	X	C	V	B	N	M	<	>	?			
й	ц	у	к	е	н	г	ш	щ	з	х	ъ	
Й	Ц	У	К	Е	Н	Г	Ш	Щ	З	Х	Ъ	
ф	ы	в	а	п	р	о	л	д	ж	э		
Ф	Ы	В	А	П	Р	О	Л	Д	Ж	Э		
я	ч	с	м	и	т	ь	б	ю				
Я	Ч	С	М	И	Т	Ь	Б	Ю				

4. Алфавит знаков правильной шестигранной игральной кости:



5. Алфавит арабских цифр:

0123456789

6. Алфавит шестнадцатеричных 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»: + -

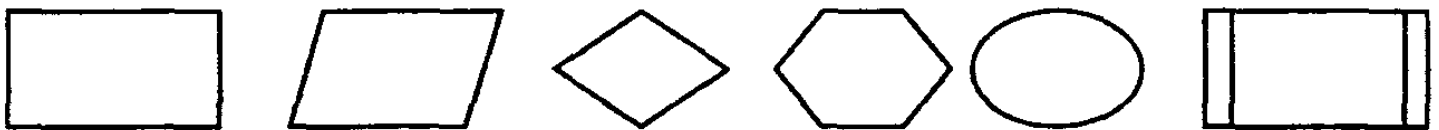
10. Алфавит прописных латинских букв:

ABCDEFGHIJKLMNOPQRSTUVWXYZ

11. Алфавит римской системы счисления:

I V X L C D M

12. Алфавит языка блок-схем изображения алгоритмов:



В канале связи сообщение, составленное из символов (букв) одного алфавита, может преобразовываться в сообщение из символов (букв) другого алфавита. 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 рисунке приведена схема, иллюстрирующая процесс передачи сообщения в случае перекодировки, а также воздействия помех[3, с.11].



1.2 Кодирование информации с помощью систем счисления

Для записи информации о количестве объектов используются числа. Числа записываются с использованием особых знаковых систем, которые называются системами счисления. Алфавит систем счисления состоит из символов, которые называются цифрами. Например, в десятичной системе счисления числа записываются с помощью десяти всем хорошо известных цифр: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Система счисления — это знаковая система, в которой числа записываются по определенным правилам с помощью символов некоторого алфавита, называемых цифрами. [7, с.6]

Все системы счисления делятся на две большие группы: позиционные и непозиционные системы счисления.

Римская непозиционная система счисления. Самой распространенной из непозиционных систем счисления является римская. В качестве цифр в ней используются: I (1), V (5), X (10), L (50), C (100), D (500), M (1000).

Значение цифры не зависит от ее положения в числе. Например, в числе XXX (30) цифра X встречается трижды и в каждом случае обозначает одну и ту же величину - число 10, три числа по 10 в сумме дают 30.

Величина числа в римской системе счисления определяется как сумма или разность цифр в числе. Если меньшая цифра стоит слева от большей, то она вычитается, если справа — прибавляется. Например, запись десятичного числа 1998 в римской системе счисления будет выглядеть следующим образом:

$$MCMXCVIII = 1000 + (1000 - 100) + (100 - 10) + 5 + 1 + 1 + 1.$$

Позиционные системы счисления. Первая позиционная система счисления была придумана еще в Древнем Вавилоне, причем вавилонская нумерация была шестидесятеричной, то есть в ней использовалось шестьдесят цифр! Интересно, что до сих пор при измерении времени мы используем основание, равное 60 (в 1 минуте содержится 60 секунд, а в 1 часе — 60 минут).

В XIX веке довольно широкое распространение получила двенадцатеричная система счисления. До сих пор мы часто употребляем дюжину (число 12): в сутках две дюжины часов, круг содержит тридцать дюжин градусов и так далее.

В позиционных системах счисления количественное значение цифры зависит от ее позиции в числе.

Наиболее распространенными в настоящее время позиционными системами счисления являются десятичная, двоичная, восьмеричная и шестнадцатеричная. Каждая позиционная система имеет определенный алфавит цифр и основание.

В позиционных системах счисления основание системы равно количеству цифр (знаков в ее алфавите) и определяет, во сколько раз различаются значения одинаковых цифр, стоящих в соседних позициях числа.

Десятичная система счисления имеет алфавит цифр, который состоит из десяти всем известных, так называемых арабских, цифр, и основание, равное 10, двоичная — две цифры и основание 2, восьмеричная — восемь цифр и основание 8, шестнадцатеричная — шестнадцать цифр (в качестве цифр используются и буквы латинского алфавита) и основание 16.

Таблица 2.2. Позиционные системы счисления

Система счисления	Основание	Алфавит цифр
Десятичная	10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Двоичная	2	0, 1
Восьмеричная	8	0, 1, 2, 3, 4, 5, 6, 7
Шестнадцатеричная	16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A (10), B (11), C (12), D (13), E (14), F (15)

Таблица 1.1

Позиционные системы счисления

Двоичная система счисления. В двоичной системе счисления основание равно 2, а алфавит состоит из двух цифр (0 и 1). Следовательно, числа в двоичной системе в развернутой форме записываются в виде суммы степеней основания 2 с коэффициентами, в качестве которых выступают цифры 0 или 1. [10, с.47]

Любое неотрицательное число в позиционной системе счисления может быть представлено в виде:



где a - основание системы счисления,

x_i - разряды (числа от 0 до $a-1$), их обозначения образуют алфавит системы счисления,

a_i - весовые коэффициенты (веса) разрядов,

n - число разрядов целой части числа,

r - число разрядов дробной части числа.

Например, число в десятичной системе счисления $57310 = 5 \cdot 10^2 + 7 \cdot 10^1 + 3 \cdot 10^0$

Число в двоичной системе счисления:



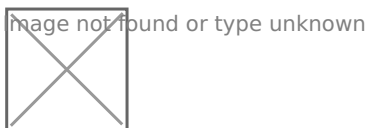
Позиционная система счисления - такая, в которой весовые коэффициенты определяются позицией разряда.

В вычислительной технике наиболее распространены: двоичная (binary, BIN), десятичная (decimal, DEC), шестнадцатеричная (hexadecimal, HEX) и непозиционная двоично-десятичная (binary coded decimal, BCD) системы исчисления. В BCD системе вес каждого разряда равен степени 10, как в десятичной системе, а каждая цифра i -го разряда кодируется 4-мя двоичными цифрами. Восьмеричная система счисления (octal, OCT) применяется реже. Следует отметить, что в цифровые устройства используют только двоичную систему счисления, так как построены на основе устройств с двумя состояниями. Другие системы счисления используются

человеком только для удобства записи, т.е. для сокращенного (и часто более удобного) представления двоичных чисел.

Формат двоичного числа

	5	2	1	6	3	1	8	4	2	1
315	114	813	412	211	110					
	9	8	7	6	5	4	3	2	1	0
						1	0	0	0	1
						1	1	1	1	0
						1	0	1		



Микропроцессоры обрабатывают упорядоченные двоичные наборы. Минимальной единицей информации является один бит (Binary digit).

Далее следуют - тетрада, или ниббл (4 бита), байт (byte, 8 бит), двойное слово (DoubleWord 16 бит) или длинное (LongWord 32 бит) и учетверенное (32 бита) слова. Младший бит обычно занимает крайнюю правую позицию, старший - соответственно крайнюю левую, т.е. "старшинство" разрядов убывает слева направо.

Шестнадцатеричное представление двоичных чисел. Для перевода числа из двоичной системы в шестнадцатеричную, его необходимо разбить, начиная справа на тетрады - группы по 4 двоичных цифры - и каждую группу представить шестнадцатеричной цифрой из таблицы. Для обратного перевода каждая HEX цифра заменяется четверкой двоичных, незначащие нули слева иногда (но не всегда) отбрасываются. Нуль обычно не отбрасывается, если старший значащий знак обозначен буквой (см. таблицу).

Таблица 1.2

Двоичные и шестнадцатеричные (HEX) коды целых чисел от 0 до 15

Число10	Число2	Hex-код
---------	--------	---------

0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E

Пример: число $57310 = 10001111012 = 023D16 = 023Dh$

Вывод по 1 главе

Для автоматизации работы с данными, относящимися к различным типам используется приём кодирования, т.е. выражение данных одного типа через данные другого типа.

Мы в своей работе будем опираться на следующее определение: «Код (code) – это совокупность знаков, символов и правил представления информации».

Сигналы, реализующие коды, обладают одной из следующих характеристик:

- - униполярный код (значения сигнала равны 0, +1, либо 0, -1);
 - полярный код (значения сигнала равны +1, -1);
 - биполярный код (значения сигнала равны 0, +1, -1).

Последовательным является такой код, в котором знаки следуют один за другим во времени. Параллельным – тот, в котором знаки передаются одновременно, образуя символ.

Естественные языки обладают большой избыточностью, поэтому для экономии памяти, объем которой ограничен, имеет смысл ликвидировать избыточность текста или уплотнить текст.

Существуют несколько способов уплотнения текста.

- 1. Переход от естественных обозначений к более компактным.
 2. Подавление повторяющихся символов.
 3. Кодирование часто используемых элементов данных.
 4. Посимвольное кодирование.
 5. Коды переменной длины.

Информация передается в виде сообщений. Дискретная информация записывается с помощью некоторого конечного набора знаков, которые называются буквами. Буква в данном расширенном понимании - любой из знаков, которые некоторым соглашением установлены для общения.

В канале связи сообщение, составленное из символов (букв) одного алфавита, может преобразовываться в сообщение из символов (букв) другого алфавита. Саму процедуру преобразования сообщения называют перекодировкой.

Для записи информации о количестве объектов используются числа. Числа записываются с использованием особых знаковых систем, которые называются системами счисления.

Все системы счисления делятся на две большие группы: позиционные и непозиционные системы счисления. Самой распространенной из непозиционных систем счисления является римская. Наиболее распространенными в настоящее время позиционными системами счисления являются десятичная, двоичная, восьмеричная и шестнадцатеричная.

Первая позиционная система счисления была придумана еще в Древнем Вавилоне, в XIX веке довольно широкое распространение получила двенадцатеричная система счисления. В настоящее время человечество приняло в использование десятичную систему счисления.

В позиционных системах счисления количественное значение цифры зависит от ее позиции в числе. Перевод из одной системы счисления в другую производится с помощью определенных правил, а в частности для двоичных, восьмеричных и шестнадцатеричных с использованием триад и тетрад, обеспечивающих удобство перевода.

ГЛАВА 2. КОДИРОВАНИЕ ИНФОРМАЦИИ В СИСТЕМЕ ОБРАБОТКИ ИНФОРМАЦИИ

2.1. Кодирование текстовой информации

В процессе обработки информации, т.е. преобразования информации из одной формы представления (знаковой системы) в другую осуществляется кодирование. Средством кодирования служит таблица соответствия, которая устанавливает взаимно однозначное соответствие между знаками или группами знаков двух различных знаковых систем. При вводе знака алфавита в компьютер путем нажатия соответствующей клавиши на клавиатуре выполняется его кодирование, т. е.

преобразование в компьютерный код. При выводе знака на экран монитора или принтер происходит обратный процесс — декодирование, когда из компьютерного кода знак преобразуется в графическое изображение.

Текст состоит из символов, поэтому символ можно считать минимальным элементом текста. Если собрать все возможные символы, которые могут встретиться в тексте: латинские буквы, буквы кириллицы, знаки препинания и т. д., и каждому из этих символов присвоить свой уникальный номер (код символа), то текст можно записать в виде набора чисел.

Для хранения кода одного символа может быть выделен один байт. С помощью одного байта можно закодировать 256 различных символов, учитывая, что каждый бит принимает значение 0 или 1, и количество их возможных сочетаний в байте равно $2^8=256$. Этого вполне достаточно для представления текстовой информации, включая прописные и строчные буквы русского и латинского алфавитов, цифры, знаки, псевдографические символы и т. д [19, с.24].

Кодирование заключается в том, что каждому символу ставится в соответствие уникальный десятичный код от 0 до 255 или соответствующий ему двоичный код от 00000000 до 11111111. Таким образом, человек различает символы по их начертанию, а компьютер — по их коду. Важно, что присвоение символу конкретного кода — это вопрос соглашения, которое фиксируется в кодовой таблице.

Технически это выглядит очень просто, однако всегда существовали достаточно веские организационные сложности. В первые годы развития вычислительной техники они были связаны с отсутствием необходимых стандартов, а в настоящее время вызваны, наоборот, избытком одновременно действующих и противоречивых стандартов. Для того чтобы весь мир одинаково кодировал текстовые данные, нужны единые таблицы кодирования, а это пока невозможно из-за противоречий между символами национальных алфавитов, а также противоречий корпоративного характера.

Существует несколько различных стандартов кодирования символов, но первоосновой для всех стал стандарт ASCII (American Standard Code for Information Interchange — американский стандартный код для информационного обмена).

В ASCII закреплены две таблицы кодирования: базовая и расширенная. В базовой таблице определены значения кодов с 0 по 127, а в расширенной — с 128 по 255. В базовой таблице находятся символы латинского алфавита, цифры, знаки

арифметических операций и знаки препинания (табл. 1.1). Кроме того, за кодами с 0 по 32 закреплены специальные функции: перевод строки, ввод пробела и т. д. Расширенная таблица содержит символы национальных алфавитов различных стран мира и так называемые символы псевдографики, с помощью которых можно, например, рисовать таблицы [19, с.25].

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		▶	! :	0	@	P	'	р	А	Р	а	▬	┌	└	р	Ё
1	␣	◀	! :	1	A	Q	а	q	Б	С	б	▬	┌	└	Р	ё
2	␣	↕	! :	2	B	R	б	r	В	Т	в	▬	┌	└	Р	ё
3	␣	↕	! :	3	C	S	с	s	Г	У	г	▬	┌	└	Р	ё
4	␣	↕	! :	4	D	T	д	t	Д	Ф	д	▬	┌	└	Р	ё
5	␣	↕	! :	5	E	U	е	u	Е	Х	е	▬	┌	└	Р	ё
6	␣	↕	! :	6	F	V	ф	v	Ж	Ц	ж	▬	┌	└	Р	ё
7	␣	↕	! :	7	G	W	г	w	З	Ч	з	▬	┌	└	Р	ё
8	␣	↕	! :	8	H	X	х	x	И	Ш	и	▬	┌	└	Р	ё
9	␣	↕	! :	9	I	Y	й	y	Й	Щ	й	▬	┌	└	Р	ё
A	␣	↕	! :	A	J	Z	з	z	К	Ъ	к	▬	┌	└	Р	ё
B	␣	↕	! :	B	K	{	и	{	Л	Ы	л	▬	┌	└	Р	ё
C	␣	↕	! :	C	L		о		М	Ь	м	▬	┌	└	Р	ё
D	␣	↕	! :	D	M	~	п	~	Н	Э	н	▬	┌	└	Р	ё
E	␣	↕	! :	E	N	▬	о	▬	О	Ю	о	▬	┌	└	Р	ё
F	␣	↕	! :	F	O	▬	о	▬	П	Я	о	▬	┌	└	Р	ё

Рис.2.1 Таблица кодов ASCII (расширенная)

Для языков, использующих кириллицу, в том числе и для русского, пришлось полностью менять вторую половину таблицы ASCII, приспособив её под кириллический алфавит. В частности, для представления символов кириллицы используется так называемая «альтернативная кодировка».

Альтернативная кодировка не подошла для ОС Windows. Пришлось передвинуть русские буквы в таблице на место псевдографики, и получили кодировку Windows 1251 (Win-1251). Кодировка символов русского языка, известная как кодировка Windows-1251, была введена «извне» - компанией Microsoft, но, учитывая широкое распространение операционных систем и других продуктов этой компании в

России, она глубоко закрепились и нашла широкое распространение.

Базовая таблица кодировки ASCII

32 пробел	48 0	64 @	80 P	96 `	112 p
33 !	49 1	65 A	81 Q	97 a	113 q
34 "	50 2	66 B	82 R	98 b	114 r
35 #	51 3	67 C	83 S	99 c	115 s
36 \$	52 4	68 D	84 T	100 d	116 t
37 %	53 5	69 E	85 U	101 e	117 u
38 &	54 6	70 F	86 V	102 f	118 v
39 ' .	55 7	71 G	87 W	103 g	119 w
40 ()	56 8	72 H	88 X	104 h	120 x
41)	57 9	73 I	89 Y	105 i	121 y
42 *	58 :	74 J	90 Z	106 j	122 z
43 +	59 ;	75 K	91 [107 k	123 {
44 ,	60 <	76 L	92 \	108 l	124
45 -	61 =	77 M	93]	109 m	125 }
46 .	62 >	78 N	94 ^	110 n	126 ~
47 /	63 ?	79 O	95 _	111 o	127

Кодировка Windows 1251

128 Ъ	144 ђ	160	176 *	192 А	208 Р	224 а	240 р
129 Ѓ	145 ‘	161 Ÿ	177 ±	193 Б	209 С	225 б	241 с
130 ,	146 ’	162 ŷ	178	194 В	210 Т	226 в	242 т
131 ђ	147 “	163 Ј	179 i	195 Г	211 У	227 г	243 у
132 „	148 ”	164 њ	180 r	196 Д	212 Ф	228 д	244 ф
133 ...	149 •	165 Г	181 μ	197 Е	213 Х	229 е	245 х
134 †	150 –	166	182 ¶	198 Ж	214 Ц	230 ж	246 ц
135 ‡	151 —	167 §	183 ·	199 З	215 Ч	231 з	247 ч
136 ‘	152 ‘	168 È	184 è	200 И	216 Ш	232 и	248 ш
137 ‰	153 ™	169 ©	185 №	201 Й	217 Щ	233 й	249 щ
138 Љ	154 љ	170 €	186 €	202 К	218 Ъ	234 к	250 ъ
139 ‘	155 ‘	171 “	187 “	203 Л	219 Ы	235 л	251 ы
140 Њ	156 њ	172 –	188 j	204 М	220 Ь	236 м	252 ь
141 К	157 к	173 -	189 S	205 Н	221 Э	237 н	253 э
142 Ћ	158 ћ	174 ©	190 s	206 О	222 Ю	238 о	254 ю
143 Љ	159 љ	175 ĩ	191 i	207 П	223 Я	239 п	255 я

Сейчас существует несколько различных кодовых таблиц для русских букв (КОИ-8, CP-1251, CP-866, Mac, ISO), причём тексты, созданные в одной кодировке, могут совершенно неправильно отображаться в другой. Решается такая проблема с помощью специальных программ перевода текста из одной кодировки в другую [4. с.67].

Другая распространённая кодировка носит название КОИ-8 (код обмена информацией, восьмизначный) – её происхождение относится к временам действия Совета Экономической Взаимопомощи государств Восточной Европы. Сегодня кодировка КОИ – 8 имеет широкое распространение в компьютерных сетях на территории России и в российском секторе Интернета.

Международный стандарт, в котором предусмотрена кодировка символов русского языка, носит названия ISO (International Standard Organization – Международный институт стандартизации). На практике данная кодировка используется редко.

Если проанализировать организационные трудности, связанные с созданием единой системы кодирования текстовых данных, то можно прийти к выводу, что они вызваны ограниченным набором кодов (256). Математикам требуется использовать в формулах специальные математические знаки, переводчикам необходимо создавать тексты, где могут встретиться символы из различных алфавитов, экономистам необходимы символы валют (\$, F, A). Для решения этой проблемы была разработана универсальная система кодирования текстовой информации — UNICODE. В этой кодировке для каждого символа отводится не один, а два байта, то есть шестнадцать битов. Очевидно, что если, кодировать символы не восьмиразрядными двоичными числами, а числами с большим разрядом то и диапазон возможных значений кодов станет на много больше. Шестнадцать разрядов позволяют обеспечить уникальные коды для 65536 различных символов — этого поля вполне достаточно для размещения в одной таблице символов большинства языков планеты. Этого хватает на латинский алфавит, кириллицу, иврит, африканские и азиатские языки, различные специализированные символы: математические, экономические, технические и многое другое.

Несмотря на тривиальную очевидность такого подхода, простой механический переход на данную систему долгое время сдерживался из-за недостатков ресурсов средств вычислительной техники (в системе кодирования UNICODE все текстовые документы становятся автоматически вдвое длиннее). Но во второй половине 90-х годов технические средства достигли необходимого уровня обеспечения ресурсами, и сегодня мы наблюдаем постепенный перевод документов и программных средств на универсальную систему кодирования UNICODE.

2.2. Кодирование целых и действительных чисел

Естественным представлением целого неотрицательного числа является двоичная система счисления. Кодирование отрицательных чисел производится тремя наиболее употребительными способами, в каждом из которых крайний левый бит - знаковый. Отрицательному числу соответствует единичный бит, а положительному - нулевой.

1. *Прямой код.* Изменение знака производится просто, путем инверсии бита знака. Пусть $00001001 = 9$, тогда $10001001 = -9$. Если при сложении двух чисел в этом коде знаки совпадают, то трудностей нет. Если знаки различаются необходимо найти наибольшее число, вычесть из него меньшее, а результату присвоить знак

наибольшего слагаемого.

2. *Обратный код*, инверсный или дополнительный "до 1". Изменение знака производится просто - инверсией всех бит: $00001001 = 9$, а $11110110 = -9$. Сложение также выполняется просто, т.к. знаковые биты можно складывать. При переносе единицы из левого (старшего) бита, она должна складываться с правым (младшим). Например: $7 + (-5) = 2$.

$$00000111 = 7$$

$$11111010 = -5 \text{ (инверсия } 00000101 = 5)$$

$$1\ 00000001$$

$$1$$

$$00000010 = 2$$

Сложение в обратном коде происходит быстрее, т.к. не требуется принятие решения, как в предыдущем случае. Однако суммирование бита переноса требует дополнительных действий. Другим недостатком этого кода является представление нуля двумя способами, т.к. инверсия $0...00$ равна $1...11$ и сумма двух разных по знаку, но равных по значению чисел дает $1...11$. Например: $(00001001 = 9) + (11110110 = -9) = 11111111$. Кстати, из этого примера понятно, почему код называется дополнительным "до 1". Этих недостатков лишен код, дополнительный до 2.

3. *Дополнительный* или дополнительный "до 2" код. Число с противоположным знаком находится инверсией исходного и добавлением к результату единицы. Например, найти код числа -9.

$$00001001 = 9 \quad 11110111 = -9$$

$$11110110 - \text{инверсия } 00001000 - \text{инверсия}$$

$$1\ 1$$

$$11110111 = -9 \quad 00001001 = 9$$

Проблемы двух нулей нет. $+0 = 00000000$, $-0 = 11111111 + 1 = 00000000$ (перенос из старшего бита не учитывается). Сложение производится по обычным для неотрицательных чисел правилам.

00001001 = 9

11110111 = -9

1 00000000

Из этого примера видно, что в каждом разряде двух равных по модулю чисел складываются две единицы, что и определило название способа. Этот метод применяется наиболее часто, и когда говорят о дополнительном коде, то имеется в виду дополнительный "до 2-х" код.

Примеры однобайтных целых чисел:

D7 D6 D5 D4 D3 D2 D1 D0 Целое двоичное без знака (256 чисел)

S D6 D5 D4 D3 D2 D1 D0 Целое двоичное со знаком (от -27 до

+(27-1)

D3 D2 D1 D0 D3 D2 D1 D0 Десятичное двоично-кодированное

старшая цифра младшая цифра (упакованный формат)

Числа с плавающей точкой. Вещественные числа хранятся в показательной форме, т.е. в виде двух составляющих: мантииссы и порядка. Различия в способах такого представления чисел заключаются в количестве байтов, отводимых под порядок и мантииссу и небольших отличиях в форме их хранения. Например в четырехбайтовом формате под мантииссу отводится 3 байта и один байт для хранения порядка (KB - короткий вещественный формат):

Пример одного из вариантов формата:

байт 3 байт 2 байт 1 байт 0

s s

порядок мантисса (от 2⁻¹ до 1)

Старший разряд старшего байта хранит знак мантиссы, следующий за ним - знак порядка.

В другом варианте знак мантиссы может храниться в старшем разряде байта 2, а сам порядок храниться в "смещенной" форме, в виде числа E-127.

Тогда представление числа D: $D = \pm M * 2^{(E-127)}$, где мантисса, E - смещенный порядок, хранящийся в старшем байте. При этом может быть принято соглашение о неявном присутствии единицы слева от десятичной точки, так что мантисса будет принимать значения от 1 до 2. Соответственно в зависимости от вариаций формата будут слегка отличаться и диапазоны представления чисел. Так, в последнем варианте, где у нормализованной мантиссы первая значащая цифра (единица) мысленно находится слева от запятой, а справа располагаются 23 разряда - 1,xx..xx, $M_{max} = 1,111..11 = 1 + 1/2 + 1/4 + 1/8 + \dots = 2$, а $M_{min} = 1,000..00 = 1$ для положительных чисел (SM=0) и -1 и -2 для отрицательных, (SM=1). Порядок числа $E_{max} = 11111110 = 254$, а $E_{min} = 00000001 = 1$. Теперь можно определить диапазон представления положительных чисел от $+D_{max} = M_{max} * 2^{(254-127)} = 3,4 * 10^{38}$ до $+D_{min} = M_{min} * 2^{(1-127)} = 1,17 * 10^{-38}$. Точность определяется числом достоверных десятичных цифр. При 23 двоичных разрядах мантиссы 223 примерно равно 107, т.е. достоверными являются только 6-7 значащих десятичных знаков, а не 38. Необходимо отметить, что значения порядка 11111111 и 00000000 по международным стандартам IEEE 754 и IEEE 854 предназначены для кодирования денормализованных чисел, отрицательной и положительной бесконечностей, неопределенности и так называемых "Не-чисел" [7, с.89].

Целые числа кодируются двоичным кодом достаточно просто - необходимо взять целое число и делить его пополам до тех пор, пока частное не будет равно единице. Совокупность остатков от каждого деления, записанная справа налево вместе с последним частным, и образует двоичный аналог десятичного числа.

Для кодирования целых чисел от 0 до 255 достаточно иметь 8 разрядов двоичного кода (8 бит). 16 бит позволяют закодировать целые числа от 0 до 65535, а 24 - уже более 16,5 миллионов различных значений.

Для кодирования действительных чисел используют 80-разрядное кодирование. При этом число предварительно преобразовывают в нормализованную форму:

$$3,1414926 = 0,31415926 \cdot 10^1$$

$$300\ 000 = 0,3 \cdot 10^6$$

Первая часть числа называется мантиссой, а вторая – характеристикой. Большую часть из 80 бит отводят для хранения мантиссы (вместе со знаком) и некоторое фиксированное количество разрядов отводят для хранения характеристики.

2.3. Кодирование графических данных

Телевизионный способ представления видеоинформации во многом повлиял на формирование принципов представления графической информации в компьютере. Одной из основополагающих идей телевидения является создание изображения на базе системы точек, «обегаемых» по очереди, например, электронным лучом. Подобную систему точек принято называть растром. Посмотрев через лупу на экран телевизора или на фотографию, вы увидите множество точек различных цветов, составляющих растр (см. рис. 1.1).

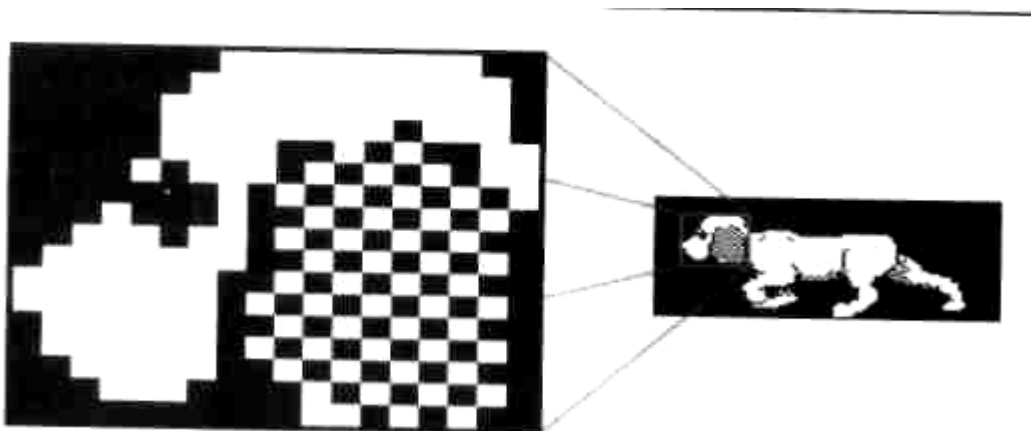
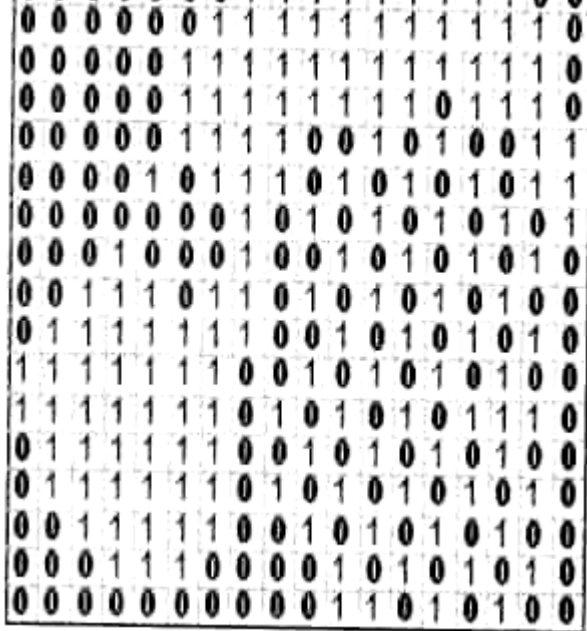


Рис. 2.2. Растровое изображение под увеличением



Поскольку линейные координаты и

индивидуальные свойства каждой точки (яркость) можно выразить с помощью целых чисел, то можно сказать, что растровое кодирование позволяет использовать двоичный код для представления графических данных. Закодировав каждый цвет каким-нибудь числом, можно представить изображение в виде последовательности чисел. Для чёрно-белого изображения на каждую точку будет достаточно одного бита: белый цвет будет обозначаться единицей, а чёрный — нулём, как показано на рисунке 1.2.

Рис. 2.3. Цифровое представление чёрно-белого изображения

Общепринятым на сегодняшний день считается представление чёрно-белых иллюстраций в виде комбинации точек с 256 градациями серого цвета, и, таким образом, для кодирования яркости любой точки обычно достаточно восьмиразрядного двоичного числа.

Когда изображение включает оттенки серого цвета, требуется кодировать каждый оттенок цвета. Очевидно, что чем больше число оттенков, тем большее количество битов необходимо на каждую точку изображения. Справедлива общая формула

$$N = 2^i,$$

где N есть число цветов, а i — необходимое количество битов информации [19, с. 27].

Например, для кодирования 256 градаций серого потребуется 8 битов, то есть 1 байт. В этом случае 0 будет означать чёрный цвет, 255 — белый, а числа от 1 до 254 — серые цвета различной яркости.

Цветные изображения кодируются сложнее. Часто при кодировании цвета используют трёхбайтовое кодирование, когда каждый байт представляет собой интенсивность одного из трёх базовых цветов — красного, зелёного и синего. Это связано с природной цветовой чувствительностью глаза человека. Практически любой цвет, воспринимаемый глазом, можно получить, смешивая три этих базовых цвета. Например, пурпурный цвет получается от смешения красного и синего, а жёлтый — от смешения красного и зелёного. Меняя пропорции, можно получить различные оттенки. Если смешать все три базовых цвета в одинаковой пропорции, то получится серый цвет. Такой способ кодирования цвета называется RGB (Red, Green, Blue).

Если яркость каждого из базовых цветов кодировать числом от 0 до 255, то потребуется 3 байта (то есть 24 бита) на каждую точку. В этом случае белый цвет будет кодироваться тремя числами (255, 255, 255), чёрный — (0, 0, 0). Коды (255, 255, 200) будут обозначать блёкло-жёлтый цвет, а (100, 0, 100) — тёмно-фиолетовый.

Режим представления цветной графики с использованием 24 двоичных разрядов называется полноцветным (True Color).

Каждому из основных цветов можно поставить в соответствие дополнительный цвет, т.е. цвет, дополняющий основной цвет до белого. Нетрудно заметить, что для любого из основных цветов дополнительным будет цвет, образованный суммой пары остальных основных цветов. Соответственно дополнительными цветами являются: голубой (Cyan), пурпурный (Magenta) и жёлтый (Yellow). Принцип декомпозиции произвольного цвета на составляющие компоненты можно применять не только для основных цветов, но и для дополнительных, т.е. любой цвет можно представить в виде суммы голубой, пурпурной и жёлтой составляющей. Такой метод кодирования цвета принят в полиграфии, но в полиграфии используется ещё и четвёртая краска — чёрная (Black). Поэтому данная система кодирования обозначается четырьмя буквами CMYK (чёрный цвет обозначается буквой K, потому, что буква B уже занята синим цветом), и для представления цветной графики в этой системе надо иметь 32 двоичных разряда. Такой режим также называется полноцветным.

Если уменьшить количество двоичных разрядов, используемых для кодирования цвета каждой точки, то можно сократить объём данных, но при этом диапазон кодируемых цветов заметно сокращается. Кодирование цветной графики 16-разрядными двоичными числами называется режимом High Color.

Отдельный элемент графического изображения имеет широко распространённое название — пиксель (от англ. picture element — элемент изображения). Чем гуще сетка пикселей на мониторе, тем лучше качество изображения. Размер графической сетки обычно представляется в форме произведения числа точек в горизонтальной строке на число строк: $M \times N$. Мы можем настраивать режим работы монитора, например: 800x600 или 1024x768 пикселей.

По существу, кодирование цветов отдельных пикселей аналогично кодированию символов текста. Таблица нумерации цветов есть своеобразный алфавит, пользуясь которым, компьютер записывает графическую информацию. В итоге, как и в случае с текстом, получается последовательность целых чисел, которая стандартным образом сохраняется в памяти.

С практической точки зрения очень важно понимать, что количество используемых на рисунке цветов существенным образом сказывается на размере графического файла. Для записи чёрно-белого изображения, имеющего 100 точек по вертикали и 100 точек по горизонтали, потребуется $100 \times 100 = 10000$ битов (1250 байтов). Изображение такого же размера, но использующее 256 градаций серого цвета, потребует $100 \times 100 \times 8 = 80\,000$ битов (10000 байтов). Изображение, использующее 24-битный код, потребует:

$100 \times 100 \times 24 = 240\,000$ битов (30 000 байтов).

2.4. Кодирование звуковой информации

Информация, в том числе графическая и звуковая, может быть представлена в аналоговой или дискретной форме. При аналоговом представлении физическая величина принимает бесконечное множество значений, причем ее значения изменяются непрерывно. При дискретном представлении физическая величина принимает конечное множество значений, причем ее величина изменяется скачкообразно. Примером аналогового представления графической информации может служить, скажем, живописное полотно, цвет которого изменяется непрерывно, а дискретного — изображение, напечатанное с помощью струйного

принтера и состоящее из отдельных точек разного цвета. Примером аналогового хранения звуковой информации является виниловая пластинка (звуковая дорожка изменяет свою форму непрерывно), а дискретного — аудиокомпакт-диск (звуковая дорожка которого содержит участки с различной отражающей способностью). Звуковая информация из аналоговой формы в дискретную преобразуется путем дискретизации, т.е. разбиения непрерывного (аналогового) звукового сигнала на отдельные элементы. В процессе дискретизации производится кодирование, т. е. присвоение каждому элементу конкретного значения в форме кода.

Дискретизация — это преобразование непрерывных изображений и звука в набор дискретных значений, каждому из которых присваивается значение его кода.

Как известно из курса физики, звук — это колебания среды, непрерывный сигнал с меняющейся амплитудой (рис. 1.3).

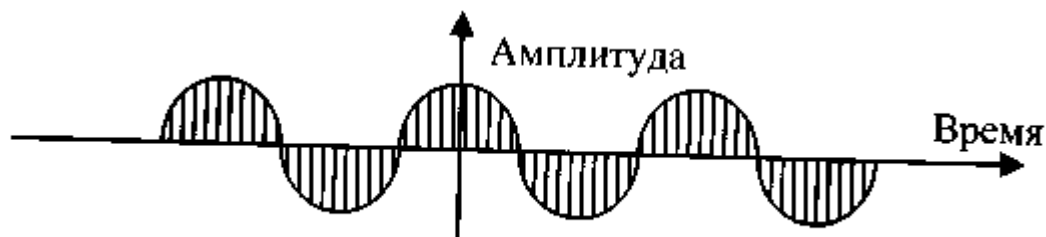


Рис.2.4. Звуковая волна

При кодировании звука этот сигнал надо представить в виде последовательности нулей и единиц. Как, например, это происходит в микрофоне? Через равные промежутки времени, очень часто (десятки тысяч раз в секунду) измеряется амплитуда колебаний. Каждое измерение производится с ограниченной точностью и записывается в двоичном виде. Частота, с которой записывается амплитуда, называется частотой дискретизации [19. с. 29].

Полученный ступенчатый сигнал сначала сглаживается посредством аналогового фильтра, а затем преобразуется в звук с помощью усилителя и динамика.

На качество воспроизведения закодированного звука в основном влияют два параметра: частота дискретизации— количество измерений амплитуды за секунду в герцах и глубина кодирования звука — размер в битах, отводимый под запись значения амплитуды.

Например, при записи на компакт-диски (CD) используются 16-разрядные значения, а частота дискретизации равна 44 032 Гц. Эти параметры обеспечивают превосходное качество звучания речи и музыки. Для стереозвука отдельно записывают данные для левого и для правого канала.

Приёмы и методы работы со звуковой информацией пришли в вычислительную технику наиболее поздно. К тому же, в отличие от числовых, текстовых и графических данных, у звукозаписей не было столь же длительной и проверенной истории кодирования. В итоге методы кодирования звуковой информации двоичным кодом далеки от стандартизации. Множество отдельных компаний разработали свои корпоративные стандарты, но среди них можно выделить два основных направления [6. с. 115].

1. Метод FM (Frequency Modulation) основан на том, что теоретически любой сложный звук можно разложить на последовательность простейших гармонических сигналов разных частот, каждый из которых представляет собой правильную синусоиду, а, следовательно, может быть описан числовыми параметрами, т.е. кодом. В природе звуковые сигналы имеют непрерывный спектр, т.е. являются аналоговыми. Их разложение в гармонические ряды и представление в виде дискретных цифровых сигналов выполняют специальные устройства – аналогово-цифровые преобразователи (АЦП). Обратное преобразование для воспроизведения звука, закодированного числовым кодом, выполняют цифро-аналоговые преобразователи (ЦАП). При таких преобразованиях неизбежны потери информации, связанные с методом кодирования, поэтому качество звукозаписи обычно получается не вполне удовлетворительным и соответствует качеству звучания простейших электромузыкальных инструментов с окрасом характерным для электронной музыки. В то же время данный метод копирования обеспечивает весьма компактный код, поэтому он нашёл применение ещё в те годы, когда ресурсы средств вычислительной техники были явно недостаточны.

2. Метод таблично волнового (Wave-Table) синтеза лучше соответствует современному уровню развития техники. В заранее подготовленных таблицах хранятся образцы звуков для множества различных музыкальных инструментов. В технике такие образцы называют сэмплами. Числовые коды выражают тип инструмента, номер его модели, высоту тона, продолжительность и интенсивность звука, динамику его изменения, некоторые параметры среды, в которой происходит звучание, а также прочие параметры, характеризующие особенности звучания. Поскольку в качестве образцов исполняются реальные звуки, то его качество получается очень высоким и приближается к качеству звучания реальных

музыкальных инструментов.

Вывод по 2 главе

Средством кодирования служит таблица соответствия, которая устанавливает взаимно однозначное соответствие между знаками или группами знаков двух различных знаковых систем.

Кодирование текстовой информации заключается в том, что каждому символу ставится в соответствие уникальный десятичный код от 0 до 255 или соответствующий ему двоичный код от 00000000 до 11111111. Сейчас существует несколько различных кодовых таблиц для символов (КОИ-8, CP-1251, CP-866, Mac, ISO, Win-1251, CP866 (DOS)), причём тексты, созданные в одной кодировке, могут совершенно неправильно отображаться в другой. Решается такая проблема с помощью специальных программ перевода текста из одной кодировки в другую. Но сегодня мы наблюдаем постепенный перевод документов и программных средств на универсальную систему кодирования UNICODE.

Основой графического изображения является системы точек, которую принято называть растром. Отдельный элемент графического изображения имеет широко распространённое название — пиксель (от англ. picture element — элемент изображения). Чем гуще сетка пикселей на мониторе, тем лучше качество изображения. Поскольку линейные координаты и индивидуальные свойства каждой точки (яркость) можно выразить с помощью целых чисел, то можно сказать, что растровое кодирование позволяет использовать двоичный код для представления графических данных. Закодировав каждый цвет каким-нибудь числом, можно представить изображение в виде последовательности чисел. Кодирование цветов отдельных пикселей аналогично кодированию символов текста. Таблица нумерации цветов есть своеобразный алфавит, пользуясь которым, компьютер записывает графическую информацию. В итоге, как и в случае с текстом, получается последовательность целых чисел, которая стандартным образом сохраняется в памяти.

Звуковая информация для представления в двоичном виде из аналоговой формы в дискретную преобразуется путем дискретизации, т.е. разбиения непрерывного (аналогового) звукового сигнала на отдельные элементы. В процессе дискретизации производится кодирование, т. е. присвоение каждому элементу конкретного значения в форме кода. При кодировании звука этот сигнал представляется в виде последовательности нулей и единиц. Приёмы и методы работы со звуковой

информацией пришли в вычислительную технику наиболее поздно. В итоге методы кодирования звуковой информации двоичным кодом далеки от стандартизации. Множество отдельных компаний разработали свои корпоративные стандарты, но среди них можно выделить два основных направления.

ЗАКЛЮЧЕНИЕ

В ходе выполнения курсовой работы нами были решены следующие задачи:

1. Рассмотрены основные определения кодирования. Алфавит кодирования;
2. Изучены способ кодирование информации с помощью систем счисления;
3. Охарактеризованы способы кодирования текстовой информации;
4. Выявлены способы кодирования целых и действительных чисел;
5. Рассмотрены методы кодирование графических данных;
6. Прослежены особенности кодирования звуковой информации.

Проведен анализ литературы в данной области знаний. Выявлены основные определения и понятия данной темы. Рассмотрен и проработан материал по практике перевода в различные системы счисления. Проведена классификация методов кодирования и уплотнения текста. Выявлена классификация алфавитов кодирования применяемых для хранения, обработки и передачи информации между человеком и техникой.

Мы выяснили, что средством кодирования служит таблица соответствия, которая устанавливает взаимно однозначное соответствие между знаками или группами знаков двух различных знаковых систем.

Подробно рассмотрели, как происходит кодирование информации в памяти компьютера. А конкретно кодирование текстовой информации с помощью различных систем кодирования. Методы представления графической информации и звука в виде понятном компьютеру.

Результат исследования темы «Кодирование информации в системах обработки данных», приведенный в курсовой работе является для нас очень важным и необходимым, как в учебной среде, так и в собственной жизни. Работа требует от себя усидчивости и мобилизации ресурсов. Знания, полученные при выполнении работы нам очень пригодятся.

СПИСОК ИСПОБЪЗУЕМОЙ ЛИТЕРАТУРЫ

1. Баричев С. В. Введение в криптографию. Электронный сборник.- М.: Вече1998. -244с
2. Ведеев Д. А. Защита данных в компьютерных сетях. Открытые системы.- М.: Дрофа 1995, №3.-180с
3. Горбоконенко В.Д., Шикина В.Е. Кодирование информации: Методические указания / сост.:,. Ульяновск: УлГТУ, 2006. – 56 с.
4. Ефремова О., Морозов В., Угринович Н. Курс компьютерной технологии с основами информатики. Учебное пособие для старших классов. – М.: АСТ; АБФ, 2000
5. Зегжда П. Теория и практика. Обеспечение информационной безопасности. – М: Альфа 1996.-173с
6. Келим Ю.М. Вычислительная техника: Учеб. пособие для студ. сред. проф. образования / Юрий Михайлович Келим. – М.: Издательский центр «Академия», 2005. – 384 с.
7. Кутугина Е.С. Арифметические и логические основы построения компьютера: Учеб. Пособие. – Изд. 3-е, перераб. и доп. – Томск, 2007. – 104 с.
8. Левин В.К. Защита информации в информационно-вычислительных системах и сетях // Программирование.- СПб.: Питер 1994. - N5.-160с
9. Леонтьев В.П. ПК: универсальный справочник пользователя.- М: Айрис 2000. - 165с
10. Максимов Н.В., Партыка Т.Л., Попов И.И. Архитектура ЭВМ и вычислительных систем: Учебник. – М.:ФОРУМ: ИНФРА-М, 2006. – 512 с. ил. (Профессиональное образование).
11. Правовая информатика и управление в сфере предпринимательства. Учебное пособие. – М.: Юристъ 1998. – 432с
12. Рожнова В.С.. Автоматизированные системы обработки учетно-аналитической информации [Текст] / под ред.- М.: Финансы и статистика, 2002. – 418с.
13. Романец Ю. В., Тимофеев П.А., Шаньгин В.Ф. Защита информации в компьютерных системах и сетях / Под ред.В.Ф. Шаньгина.-2-е изд., перераб. и доп.-М.: Радио и связь. 2001-376 с: ил.
14. Савельев А. Я. Основы информатики: Учебник для вузов. – М.: Оникс 2001.-370с
15. Семакин И.Г., Хеннер Е.К. Информатика. 10-й класс. – М.: Лаборатория Базовых Знаний, 2001
16. Симонович С.В. Информатика. Базовый курс. – СПб: Питер 2000. -137с
17. Сяо Д., Керр Д., Мэдник С. Защита ЭВМ.- М.: Айрис 1982.- 226с

18. Угринович Н.Д. Информатика и информационные технологии. Учебник для 10-11 классов / Н.Д. Угринович. –М.:БИНОМ. Лаборатория знаний, 2003. -512 с.: ил.
19. Яновский А.В., Воронкова И.А. Информационные технологии: Учеб. Пособие. – Томск, 2007. – 148 с.