

Содержание:

Введение

В мире идет постоянный обмен потоками информации. Источниками могут быть люди, технические устройства, различные вещи, объекты неживой и живой природы. Получать сведения может как один объект, так и несколько.

Для более качественного обмена данными одновременно осуществляется кодирование и обработка информации на стороне передатчика (подготовка данных и преобразование их в форму, удобную для трансляции, обработки и хранения), пересылка и декодирование на стороне приемника (преобразование кодированных данных в исходную форму). Это взаимосвязанные задачи: источник и приемник должны обладать сходными алгоритмами обработки сведений, иначе процесс кодирования-декодирования будет невозможен. Кодирование и обработка графической и мультимедийной информации обычно реализуются на основе вычислительной техники.

Кодирования информации - проблема, имеющая достаточно давнюю историю, гораздо более давнюю, нежели история развития вычислительной техники, которая обычно шла параллельно с историей развития проблемы сжатие и шифровки информации.

Цель курсовой работы провести анализ методов кодирования данных.

Для достижения поставленной цели были сформулированы следующие задачи:

- изучить способы представления информации в ЭВМ;
- рассмотреть способы кодирования и декодирования информации числовой, текстовой и графической;
- исследовать более подробно кодирование текстовой информации в компьютере.

Объектом исследования курсовой работы является способы предоставления информации в ЭВМ

Предметом исследования является методы кодирования данных

В процессе исследовательской работы были применены совокупность методов сравнительно-правового анализа, классификаций и обобщений, а также описания - один из важных методов изучения рассматриваемого вопроса.

Структурная работа состоит из введения, трех глав, заключения, содержащего основанные на результатах проведенного исследования выводы, библиографического списка.

Глава 1 Способы представления информации в ЭВМ

Окружающий мир заполнен информацией. Ежесекундно через органы чувств человек получает сотни сигналов и сообщений. Столь значимую составляющую жизни невозможно оставить без внимания, появилась даже специальная область знаний, специализирующаяся на информационных процессах и явлениях, - информатика. Ее основное оружие - умные электронно-вычислительные машины, способные быстро и точно анализировать данные и манипулировать ими. Способы представления информации в ЭВМ отличаются от тех, к которым мы привыкли, и обеспечивают максимальную эффективность вычислительного процесса.

Информация - понятие глобальное, дать ее всеохватное определение затруднительно. До сих пор не существует единого общенаучного термина, каждая область знаний оперирует собственным представлением об информации. Для простоты можно определить ее как данные о состоянии окружающего мира во всех его проявлениях^[1].

Информация имеет смысл лишь тогда, когда ее кто-нибудь воспринимает или использует. В отличие от энергии или массы, которые, как известно, не пропадают, а лишь трансформируются, информация вполне может исчезнуть.

Основная задача информатики - научиться собирать, хранить и передавать данные. Реализация этого - дело непростое. Информация бывает разная, и каждый ее вид требует к себе особого подхода.

1.1 Виды информации

На сегодняшний день люди научились работать с огромным разнообразием данных, отличающихся природой происхождения и структурой.

Самые востребованные виды информации:

1.Графическая информация – это самый первый вид данных, которым человечество научилось манипулировать. Она доступна для восприятия и не требует особенных преобразований. Наскальные рисунки – древнейшее хранилище информации об окружающем мире. На смену им пришли живопись, фотография и технические чертежи.

2.Числовая информация позволяет описать количественные характеристики объектов. Важность данных этого типа взлетела до небес при развитии торговли и денежного обмена. Чтобы успешно хранить и передавать числовую информацию, пришлось придумать специальные системы символов. Каждая культура считала деньги по-своему, так образовались разные системы счисления.

3.Текстовая информация – это закодированная особыми символами человеческая речь. С изобретением письменности стало возможным передавать любые концепции на сколь угодно большое расстояние, а также передавать знания следующим поколениям. Для удобства манипуляций с текстовой информацией человечеству пришлось изобрести бумагу и книгопечатание.

4.Звуковая информация долго не поддавалась человеку. Лишь в конце XIX века появились первые звукозаписывающие устройства, позволяющие воспринимать и сохранять данные.

5.Видеоинформация – живая графика – покорила человека с изобретением кинематографического аппарата.

Все эти данные могут быть записаны, обработаны электронно-вычислительными машинами и переданы от человека к человеку. Они могут храниться без потерь на протяжении долгого времени. Существуют и другие разновидности информации, с которыми человечество еще не научилось работать, например, тактильная или вкусовая.

Деление данных на виды имеет для информатики большое значение. Формы представления информации в ЭВМ базируются на ее особенностях, а каждый вид данных имеет специфическую структуру. Так, символьная и графическая информация обрабатываются машиной по-разному.

Главные остановочные пункты в жизненном цикле информации следующие:

- восприятие и сбор;
- хранение;

- передача;
- воспроизведение или отображение.

Во время длительного хранения или передачи данные могут искажаться или теряться. Значительные ошибки нарушают или полностью изменяют суть информации, следовательно, их нужно любой ценой избежать.

Для облегчения манипуляций с данными была придумана кодировка. Суть процесса кодирования в том, что информация по строго определенным правилам переводится в другую форму, над ней осуществляется некая операция, а затем происходит обратное преобразование.

Одна из первых удачных попыток кодирования – световые сигналы. Мигание источников света – удобный способ передачи информации на большие расстояния. С развитием техники люди придумали еще множество способов шифровать данные: электрические сигналы, радиоволны.

Кодирование обеспечивает большую сохранность и защищенность данных, позволяет увеличить скорость передачи информации и облегчить ее обработку.

Таким образом, информационный цикл приобретает следующий вид:

- сбор;
- кодирование;
- хранение;
- передача;
- декодирование;
- воспроизведение.

В процессе жизненного цикла данные могут неоднократно подвергаться кодированию и декодированию с использованием разных кодовых систем. Это необходимо для приведения информации в более удобное состояние для осуществления конкретной операции.

1.2 Материальная основа информации

Производить какие-либо реальные манипуляции можно только над материальными объектами, имеющими определенные характеристики, которые можно зафиксировать и измерить. Представление информации в ЭВМ базируется на электрических сигналах.

Работающие с данными узлы машины представлены огромным количеством крошечных элементов, которые в любой момент времени пребывают в одном из двух состояний: включенном или выключенном. Конкретная технология реализации может отличаться для разных ЭВМ и даже для разных блоков одной машины. Выключенное состояние обозначается нулем – отсутствие сигнала, включенное – единицей[2].

Количество информации, получаемое от одного структурного элемента, – один бит. Слово «бит» образовано от выражения binary digit (двоичная цифра). 1 бит – минимальная единица информации. Вся информация в компьютерах представлена последовательностью битов – нулей и единиц. Поразительно, какое огромное многообразие данных может быть зашифровано таким простым способом!

Представление информации в ЭВМ в виде отдельных точечных значений называется дискретным. По сравнению с аналоговым, оно проще реализуется и позволяет легче оперировать большими объемами данных.

Двоичный код

Последовательность битов, представляющая некоторые данные, называется двоичным кодом. С его помощью может быть закодирована любая информация: числовая, символьная, графическая.

Правила, по которым данные преобразуются в машинный код, специфичны для каждого типа. Отдельные значения в них могут совпадать, поэтому декодирование всегда производится в зависимости от контекста.

Двоичное представление информации в ЭВМ имеет ряд преимуществ:

- удобство реализации: двухпозиционные элементы гораздо проще и надежнее трех- и более позиционных;
- помехоустойчивость: сигнал, в котором присутствуют только два возможных состояния гораздо проще для восприятия;
- простота вычислений: двоичная арифметика максимально проста.

Математическая основа представления информации в ЭВМ – система счисления с основанием 2. Она намного проще привычной нам десятичной системы, оперирует всего двумя цифрами – нулем и единицей – и определяет правила всех математических операций, производимых над двоичным кодом.

Существует три способа представления числовой информации в ЭВМ:

- число с фиксированной точкой;
- число с плавающей точкой;
- двоично-десятичное представление.

У чисел с фиксированной точкой, как следует из названия, место точки (запятой), отделяющей дробную часть строго определено и зафиксировано.

1. Если точка находится после последней значащей цифры – число является целым.
2. Расположение точки перед первой значащей цифрой соответствует правильной дроби (меньше единицы по модулю).

Для определения знака выделен самый первый разряд. У положительных чисел в нем расположен 0, у отрицательных – 1.

Главное преимущество такой формы представления – отсутствие погрешностей округления при вычислениях. Главный недостаток – ограниченный диапазон значений, зависящий от разрядной сетки конкретной ЭВМ.

Числа с плавающей точкой представлены сочетанием значений мантиссы и порядка. Такая форма записи называется полулогарифмической.

Точность вычислений для такого представления зависит от длины мантиссы: при округлении могут отсекаются лишние разряды.

И числа с фиксированной точкой, и мантисса, и порядок чисел с плавающей точкой представлены в двоичной системе.

Двоично-десятичное представление чисел обеспечивается специальными процессорами в составе ЭВМ. Число обрабатывается как десятичное, но каждая его цифра представляется двоичной тетрадой. Это позволяет сократить время обработки больших массивов десятичных чисел.

Текстовая информация

Для представления текстовой информации в ЭВМ используются специальные таблицы, в которых каждому символу ставится в соответствие уникальный двоичный код.

В наиболее популярной кодировке ASCII (Американский стандарт кода для международного обмена) на один символ выделено 8 бит – 1 байт. Для кодовой единицы такого размера существует 256 уникальных комбинаций, следовательно, можно закодировать 256 разных текстовых символов. Помимо букв разных алфавитов, в таблице учитываются математические операторы, знаки препинания и прочие специальные элементы.

Конечно, 256 комбинаций – слишком мало для нашего мультикультурного мира. Некоторые языки сами по себе содержат больше букв. В таблицах Unicode увеличили размер базовой единицы кода, выделив под нее 2 байта (16 бит). Это позволило увеличить предел кодирования до 65536 элементов[3].

Растровая графика

Представление графической информации в ЭВМ называют матричным. Оно основано на разбиении изображения на ряды точек (пикселей). Для каждого пикселя информация о положении, цвете и яркости сохраняется отдельно.

В черно-белых изображениях для точки достаточно указать "степень серости" – одну из 256 градаций серого цвета. Для этого выделяется 1 байт (8 бит).

Цветные иллюстрации требуют больше данных. Чтобы закодировать информацию о цвете точки, его представляют в виде композиции трех базовых цветов: красного, зеленого и синего. Это модель Red-Green-Blue – RGB. Кодирование одной точки цветного изображения требует 24 разрядов – по одному байту (8 бит) для каждой составляющей.

Векторная графика

Изображение может быть описано и по-другому. Для этого оно разбивается на элементарные фигуры – отрезки, дуги, круги. Каждая часть может быть описана с помощью математических формул. Так, круг представляется совокупностью координаты центра и радиуса окружности. Такой способ описания графики называется векторным.

Глава 2. Способы кодирования и декодирования информации числовой, текстовой и графической

Эксплуатация электронно-вычислительной техники для обработки данных стала важным этапом в процессе совершенствования систем управления и планирования. Но такой метод сбора и обработки информации несколько отличается от привычного, поэтому требует преобразования в систему символов, понятных компьютеру.

Кодирование данных – это обязательный этап в процессе сбора и обработки информации.

Как правило, под кодом подразумевают сочетание знаков, которое соответствует передаваемым данным или некоторым их качественным характеристикам. А кодирование – это процесс составления зашифрованной комбинации в виде списка сокращений или специальных символов, которые полностью передают изначальный смысл послания. Кодирование еще иногда называют шифрованием, но стоит знать, что последняя процедура предполагает защиту данных от взлома и прочтения третьими лицами[4].

Цель кодирования заключается в представлении сведений в удобном и лаконичном формате для упрощения их передачи и обработки на вычислительных устройствах. Компьютеры оперируют лишь информацией определенной формы, поэтому так важно не забывать об этом во избежание проблем. Принципиальная схема обработки данных включает в себя поиск, сортировку и упорядочивание, а кодирование в ней встречается на этапе ввода сведений в виде кода.

Вопрос о том, что такое кодирование и декодирование, может возникнуть у пользователя ПК по различным причинам, но в любом случае важно донести корректную информацию, которая позволит юзеру успешно продвигаться в потоке информационных технологий дальше. Как вы понимаете, после процесса обработки данных получается выходной код. Если такой фрагмент расшифровать, то образуется исходная информация. То есть декодирование – это процесс, обратный шифрованию.

Если во время кодирования данные приобретают вид символьных сигналов, которые полностью соответствуют передаваемому объекту, то при декодировании из кода изымается передаваемая информация или некоторые ее характеристики.

Получателей закодированных сообщений может быть несколько, но очень важно, чтобы сведения попали в руки именно к ним и не были раскрыты раньше третьими лицами. Поэтому стоит изучить процессы кодирования и декодирования информации. Именно они помогают обмениваться конфиденциальными сведениями между группой собеседников.

2.1 Кодирование и декодирование текстовой информации

При нажатии на клавиатурную клавишу компьютер получает сигнал в виде двоичного числа, расшифровку которого можно найти в кодовой таблице – внутреннем представлении знаков в ПК. Стандартом во всем мире считают таблицу ASCII.

Однако мало знать, что такое кодирование и декодирование, необходимо еще понимать, как располагаются данные в компьютере. К примеру, для хранения одного символа двоичного кода электронно-вычислительная машина выделяет 1 байт, то есть 8 бит. Эта ячейка может принимать только два значения: 0 и 1. Получается, что один байт позволяет зашифровать 256 разных символов, ведь именно такое количество комбинаций можно составить. Эти сочетания и являются ключевой частью таблицы ASCII. К примеру, буква S кодируется как 01010011. Когда вы нажимаете ее на клавиатуре, происходит кодирование и декодирование данных, и мы получаем ожидаемый результат на экране.

Половина таблицы стандартов ASCII содержит коды цифр, управляющих символов и латинских букв. Другая ее часть заполняется национальными знаками, псевдографическими знаками и символами, которые не имеют отношения к математике. Совершенно ясно, что в различных странах эта часть таблицы будет отличаться. Цифры при вводе также преобразовываются в двоичную систему вычисления согласно стандартной сводке.

Легче будет понять, что такое кодирование информации, на примере. Рассмотрим коды английского символа «С» и русской буквы «С». Заметим, что взяты символы заглавные, и их коды отличаются от строчных. Английский символ будет выглядеть как 01000010, а русский - 11010001. То, что для человека на экране монитора выглядит одинаково, компьютер воспринимает совершенно по-разному. Необходимо также обратить внимание на то, что коды первых 128 символов

остаются неизменны, а начиная от 129 и далее одному двоичному коду могут соответствовать различные буквы в зависимости от используемой кодовой таблицы. К примеру, десятичный код 194 может соответствовать в КОИ8 букве «б», в CP1251 - «В», в ISO - «Т», а в кодировках CP866 и Mac вообще этому коду не соответствует ни один символ. Поэтому, когда при открытии текста мы вместо русских слов видим буквенную-символьную абракадабру, это означает, что такое кодирование информации нам не подходит и нужно выбрать другой конвертор символов[5].

2.2 Примеры кодирования и декодирования чисел

Кодирование чисел

В двоичной системе счисления, которую активно используют компьютеры, встречаются лишь две цифры – 0 и 1.

Действия с образующимися числами двоичной системы изучает двоичная арифметика. Большинство законов основных математических действий для таких цифр остаются актуальными.

Предлагаем рассмотреть 2 способа кодировки числа 45. Если эта цифра встречается в пределах текстового фрагмента, то каждая ее составляющая будет закодирована, согласно таблице стандартов ASCII, 8 битами. Четверка превратится в 01000011, а пятерка – в 01010011.

Если число 45 используется для вычислений, то будет задействована специальная методика преобразования в восьмиразрядный двоичный код 001011012, для хранения которого нужен будет всего лишь 1 байт.

В двоичной системе исчисления берутся всего два варианта значения — 0 и 1. Все основные операции с двоичными числами использует наука под названием двоичная арифметика. Эти действия имеют свои особенности. Возьмем, к примеру, число 45, набранное на клавиатуре. Каждая цифра имеет свой восьмиразрядный код в кодовой таблице ASCII, поэтому число занимает два байта (16 бит): 5 - 01010011, 4 - 01000011 . Для того чтобы использовать это число в вычислениях, оно переводится по специальным алгоритмам в двоичную систему исчисления в виде восьмиразрядного двоичного числа: 45 - 00101101.

Кодирование на двух нижних каналах характеризует метод представления информации сигналами, которые распространяются по среде транспортировки. Кодирование можно рассматривать как двухступенчатое. И ясно, что на принимающей стороне реализуется симметричное декодирование.

Логическое кодирование данных изменяет поток бит созданного кадра MAC-уровня в последовательность символов, которые подлежат физическому кодированию для транспортировки по каналу связи. Для логического кодирования используют разные схемы:

- 4B/5B — каждые 4 бита входного потока кодируются 5-битным символом (табл 1.1). Получается двукратная избыточность, так как $2^4 = 16$ входных комбинаций показываюся символами из $2^5 = 32$. Расходы по количеству битовых интервалов составляют: $(5-4)/4 = 1/4$ (25%). Такая избыточность разрешает определить ряд служебных символов, которые служат для синхронизации. Применяется в 100BaseFX/TX, FDDI

- 8B/10B — аналогичная схема (8 бит кодируются 10-битным символом) но уже избыточность равна 4 раза (2^8 входных в 2^{10} выходных).

- 5B/6B — 5 бит входного потока кодируются 6-битными символами. Применяется в 100VG-AnyLAN

- 8B/6T — 8 бит входного потока кодируются шестью троичными (T = ternary) цифрами (-,0,+). К примеру: 00h: +-00+-; 01h: 0+--+ = 0; Код имеет избыточность $3^6/2^8 = 729/256 = 2,85$. Скорость транспортировки символов в линию является ниже битовой скорости и их поступления на кодирования. Применяется в 100BaseT4.

- Вставка бит — такая схема работает на исключение недопустимых последовательностей бит. Ее работу объясним на реализации в протоколе HDLC. Тут входной поток смотрится как непрерывная последовательность бит, для которой цепочка из более чем пяти смежных 1 анализируется как служебный сигнал (пример: 01111110 является флагом-разделителем кадра). Если в транслируемом потоке встречается непрерывная последовательность из 1, то после каждой пятой в выходной поток передатчик вставляет 0. Приемник анализирует входящую цепочку, и если после цепочки 011111 он видит 0, то он его отбрасывает и последовательность 011111 присоединяет к остальному выходному потоку данных. Если принят бит 1, то последовательность 011111 смотрится как служебный символ. Такая техника решает две задачи — исключать длинные

монотонные последовательности, которые неудобные для самосинхронизации физического кодирования и разрешает опознание границ кадра и особых состояний в непрерывном битовом потоке[6].

Избыточность логического кодирования разрешает облегчить задачи физического кодирования — исключить неудобные битовые последовательности, улучшить спектральные характеристики физического сигнала и др. Физическое/сигнальное кодирование пишет правила представления дискретных символов, результат логического кодирования в результате физические сигналы линии. Физические сигналы могут иметь непрерывную (аналоговую) форму — бесконечное число значений, из которого выбирают допустимое распознаваемое множество. На уровне физических сигналов вместо битовой скорости (бит/с) используют понятие скорость изменения сигнала в линии которая измеряется в бодах (baud). Под таким определением определяют число изменений различных состояний линии за единицу времени. На физическом уровне проходит синхронизация приемника и передатчика. Внешнюю синхронизацию не используют из-за дороговизны реализации еще одного канала. Много схем физического кодирования являются самосинхронизирующимися — они разрешают выделить синхросигнал из принимаемой последовательности состояний канала.

Скремблирование на физическом уровне разрешает подавить очень сильные спектральные характеристики сигнала, размазывая их по некоторой полосе спектра. Очень сильные помеха искажают соседние каналы передачи. При разговоре о физическом кодировании, возможное использование следующие термины:

- транзитное кодирование — информативным есть переход из одного состояния в другое
- потенциальное кодирование — информативным есть уровень сигнала в конкретные моменты времени
- полярное — сигнал одной полярности реализуется для представления одного значения, сигнал другой полярности для — другого. При оптоволоконное транспортировке вместо полярности используют амплитуды импульса
- униполярное — сигнал одной полярности реализуется для представления одного значения, нулевой сигнал — для другого

- биполярное — используется отрицательное, положительное и нулевое значения для представления трех состояний

- двухфазное — в каждом битовом интервале присутствует переход из одного состояния в другое, что используется для выделения синхросигнала.

2.3 Популярные схемы кодирования, которые применяются в локальных сетях

AMI/ABP

AMI — Alternate Mark Inversion или же ABP — Alternate bipolare, биполярная схема, которая использует значения $+V$, $0V$ и $-V$. Все нулевые биты имеют значения $0V$, единичные — чередующимися значениями $+V$, $-V$. Применяется в DSx (DS1 — DS4), ISDN. Такая схема не есть полностью самосинхронизирующейся — длинная цепочка нулей приведет к потере синхронизации.

MAMI

MAMI — Modified Alternate Mark Inversion, или же ASI — модифицированная схема AMI, импульсами чередующейся полярности кодируется 0, а 1 — нулевым потенциалом. Применяется в ISDN (S/T — интерфейсы).

B8ZS

B8ZS — Bipolar with 8 Zero Substitution, схема аналогичная AMI, но для синхронизации исключает цепочки 8 и более нулей (за счет вставки бит).

HDB3

HDB3 — High Density Bipolar 3, схема аналогичная AMI, но не допускает передачи цепочки более трех нулей. Вместо последовательности из четырех нулей вставляется один из четырех биполярных кодов.

Манчестерское кодирование

Manchester encoding — двухфазное полярное/униполярное самосинхронизирующееся кодирование. Текущий бит узнается по направлению смены состояния в середине битового интервала: от $-V$ к $+V$: 1. От $+V$ к $-V$: 0. Переход в начале интервала может и не быть. Применяется в Ethernet. (В начальных версиях — униполярное).

Дифференциальное манчестерское кодирование

Differential manchester encoding — двухфазное полярное/униполярное самосинхронизирующиеся код. Текущий бит узнается по наличию перехода в начале битового интервала (рис. 4.1), например 0 — есть переход (Вертикальный фрагмент), 1 — нет перехода (горизонтальный фрагмент). Можно и наоборот определять 0 и 1. В середине битового интервала переход есть всегда. Он нужен для синхронизации. В Token Ring применяется измененная версия такой схемы, где кроме бит 0 и 1 определены также два бита j и k (Рис. 4.2). Здесь нет переходов в середине интервала. Бит k имеет переход в начале интервала, а j — нет.

MLT-3

Трехуровневое кодирование со скремблированием который не самосинхронизуется. Используются уровни (+V, 0, -V) постоянные в линии каждого битового интервала. При передаче 0 значения не меняются, при передаче 1 — меняются на соседние по цепочке +V, 0, -V, 0, +V и тд. (рис. 5). Такая схема является усложненным вариантом NRZI. Применяется в FDDI и 100BaseTX.

NRZ и NRZI

NRZ — Non-return to zero (без возврата к нулю), биполярная нетранзитивная схема (состояния меняются на границе), которая имеет 2 варианта. Первый вариант это недифференциальное NRZ (используется в RS-232) состояние напрямую отражает значение бита (рис. 6.а). В другом варианте — дифференциальном, NRZ состояние меняется в начале битового интервала для 1 и не меняется для 0. (рис.6.б). Привязки 1 и 0 к определенному состоянию нету[7].

NRZI — Non-return to zero Inverted, измененная схема NRZ (рис. 6.в). Тут состояния изменяются на противоположные в начале битового интервала 0, и не меняются при передаче 1. Возможна и обратная схема представления. Используются в FDDI, 100BaseFX.

RZ

RZ — Return to zero (с возвратом к нулю), биполярная транзитивная самосинхронизирующаяся схема. Состояние в определенный момент битового интервала всегда возвращается к нулю. Имеет дифференциальный/недифференциальный варианты. В дифференциальном привязки 1 и 0 к состоянию нету. (рис. 7.а).

FM 0

FM 0 — Frequency Modulation 0 (частотная модуляция), самосинхронизирующийся полярный код. Меняется на противоположное на границе каждого битового интервала. При передаче 1 в течение битового интервала состояние неизменное. При передаче 0, в середине битового интервала состояние меняется на противоположное. (рис. 8). Используется в LocalTalk.

РАМ 5

РАМ 5 — Pulse Amplitude Modulation, пятиуровневое биполярное кодирование, где пара бит в зависимости от предыстории оказывается одним из 5 уровней потенциала. Нужна неширокая полоса частот (вдвое ниже битовой скорости). Используется в 1000BaseT.

2В1Q

Здесь пара бит оказывается одним четверичным символом (Quaternary symbol), где каждому соответствует один из 4 уровней сигнала.

4В3Т — блок из 4 бит (16 состояний) кодируется тремя троичными символами (27 символов). Из множества возможных методов изменений рассмотрим MMS43, который используется в интерфейсе BRI сетей ISDN (таблица). Тут применяются специальные методы для исключения постоянной составляющей напряжения в линии, в следствии чего кодирования ряда комбинаций зависит от предыстории — состояния, где находится кодер. Пример: последовательность бит 1100 1101 будет представлена как: + + + — 0 -.

Схемы, которые не являются самосинхронизирующимися, вместе с логическим кодированием и определением фиксированной длительности битовых интервалов разрешают достигать синхронизации. Старт-бит и стоп-бит служат для синхронизации, а контрольный бит вводит избыточность для повышения достоверности приема.

Глава 3. Кодирование текстовой информации в компьютере

Кодирование текстовой информации в компьютере – порой неотъемлемое условие корректной работы устройства или отображения того или иного фрагмента. Как

происходит этот процесс в ходе работы компьютера с текстом и визуальной информацией, звуком – все это мы разберем в курсовой работе.

Электронная вычислительная машина (которую мы в повседневной жизни называем компьютером) воспринимает текст весьма специфично. Для нее кодирование текстовой информации очень важно, поскольку она воспринимает каждый текстовый фрагмент в качестве группы обособленных друг от друга символов.

В роли символов для компьютера выступают не только русские, английские и другие буквы, но и еще знаки препинания, а также другие знаки. Даже пробел, которым мы разделяем слова при печатании на компьютере, устройство воспринимает как символ. Чем-то очень напоминает высшую математику, ведь там, по мнению многих профессоров, ноль имеет двойное значение: он и является числом, и одновременно ничего не обозначает. Даже для философов вопрос пробела в тексте может стать актуальной проблемой. Шутка, конечно, но, как говорится, в каждой шутке есть доля правды.

Итак, для восприятия информации компьютеру необходимо запустить процессы обработки. А какая вообще бывает информация?

3.1 Двоичный код

Информация может быть текстовой, числовой, звуковой, графической. Компьютер должен запустить процессы, обеспечивающие кодирование текстовой информации, чтобы вывести на экран то, что мы, например, печатаем на клавиатуре. Мы будем видеть символы и буквы, это понятно. А что же видит машина? Она воспринимает абсолютно всю информацию – и речь сейчас идет не только о тексте – в качестве определенной последовательности нулей и единиц. Они составляют основу так называемого двоичного кода. Соответственно, процесс, который преобразует поступающую на устройство информацию в понятную ему, имеет название “двоичное кодирование текстовой информации”.

Почему наибольшее распространение в электронных машинах получило именно кодирование информации двоичным кодом? Текстовой основой, которая кодируется при помощи нулей и единиц, может быть абсолютно любая последовательность символов и знаков. Однако это не единственное преимущество, которое имеет двоичное текстовое кодирование информации. Все

дело в том, что принцип, на котором устроен такой способ кодирования, очень прост, но в то же время достаточно функционален. Когда есть электрический импульс, его маркируют (условно, конечно) единицей. Нет импульса – маркируют нулем. То есть текстовое кодирование информации базируется на принципе построения последовательности электрических импульсов. Логическая последовательность, составленная из символов двоичного кода, называется машинным языком. В то же время кодирование и обработка текстовой информации при помощи двоичного кода позволяют осуществлять операции за достаточно краткий промежуток времени[8].

Цифра, воспринимаемая машиной, кроет в себе некоторое количество информации. Оно равно одному биту. Это касается каждой единицы и каждого нуля, которые составляют ту или иную последовательность зашифрованной информации.

Соответственно, количество информации в любом случае можно определить, просто зная количество символов в последовательности двоичного кода. Они будут численно равны между собой. 2 цифры в коде несут в себе информацию объемом в 2 бита, 10 цифр – 10 бит и так далее. Принцип определения информационного объема, который кроется в том или ином фрагменте двоичного кода, достаточно прост, как вы видите.

Все дело в том, что закодировать один символ, который мы видим на экране, можно при помощи условной единицы измерения, называемой байтом. Как написано выше, у двоичного кода есть так называемая информационная нагрузка. Напомним, что численно она равняется суммарному количеству нулей и единиц в выбранном фрагменте кода. Так вот, 8 бит составляют 1 байт. Комбинации сигналов при этом могут быть самыми разными, как это легко можно заметить, нарисовав на бумаге прямоугольник, состоящий из 8 ячеек равного размера.

Выходит, что закодировать текстовую информацию можно при помощи алфавита, имеющего мощность 256 символов. В чем заключается суть? Смысл кроется в том, что каждый символ будет обладать своим двоичным кодом. Комбинации, “привязываемые” к определенным символам, начинаются от 00000000 и заканчиваются 11111111. Если переходить от двоичной к десятичной системе счисления, то кодировать информацию в такой системе можно от 0 до 255.

Не стоит забывать о том, что сейчас есть различные таблицы, которые используют кодировку букв русского алфавита. Это, например, ISO и КОИ-8, Mac и CP в двух вариациях: 1251 и 866. Легко убедиться в том, что текст, закодированный в одной

из таких таблиц, не отобразится корректно в отличной от данной кодировке. Это происходит из-за того, что в разных таблицах к одному и тому же двоичному коду соответствуют различные символы.

Поначалу это было проблемой. Однако в настоящее время в программах уже встроены специальные алгоритмы, которые конвертируют текст, приводя его к корректному виду. 1997 год ознаменовался созданием кодировки под названием Unicode. В ней каждый символ имеет в своем распоряжении сразу 2 байта. Это позволяет закодировать текст, имеющий гораздо большее количество символов. 256 и 65536: есть ведь разница?

Кодирование текстовой и графической информации имеет некоторые схожие моменты. Как известно, для вывода графической информации используется периферийное устройство компьютера под названием “монитор”. Графика сейчас (речь идет сейчас именно о компьютерной графике) широко используется в самых разных сферах. Благо, аппаратные возможности персональных компьютеров позволяют решать достаточно сложные графические задачи.

Обрабатывать видеоинформацию стало возможным в последние годы. Но текст при этом значительно “легче” графики, что, в принципе, понятно. Из-за этого конечный размер файлов графики необходимо увеличивать. Преодолеть подобные проблемы можно, зная суть, в которой представляется графическая информация.

Давайте для начала разберемся, на какие группы подразделяется данный вид информации. Во-первых, это растровая. Во-вторых, векторная.

Растровые изображения достаточно схожи с клетчатой бумагой. Каждая клетка на такой бумаге закрашивается тем или иным цветом. Такой принцип чем-то напоминает мозаику. То есть получается, что в растровой графике изображение разбивается на отдельные элементарные части. Их именуют пикселями. В переводе на русский язык пиксели обозначают “точки”. Логично, что пиксели упорядочены относительно строк. Графическая сетка состоит как раз из определенного количества пикселей. Ее также называют растром. Принимая во внимание эти два определения, можно сказать, что растровое изображение является не чем иным, как набором пикселей, которые отображаются на сетке прямоугольного типа[9].

Растр монитора и размер пикселя влияют на качество изображения. Оно будет тем выше, чем больше растр у монитора. Размеры раstra - это разрешение экрана, о котором наверняка слышал каждый пользователь. Одной из наиболее важных характеристик, которые имеют экраны компьютера, является разрешающая

способность, а не только разрешение. Оно показывает, сколько пикселей приходится на ту или иную единицу длины. Обычно разрешающая способность монитора измеряется в пикселях на дюйм. Чем больше пикселей будет приходится на единицу длины, тем выше будет качество, поскольку “зернистость” при этом снижается.

Кодирование текстовой и звуковой информации, как и другие виды кодирования, имеет некоторые особенности. Речь сейчас пойдет о последнем процессе: кодировании звуковой информации.

Представление звукового потока (как и отдельного звука) может быть произведено при помощи двух способов.

При этом величина может принимать действительно огромное количество различных значений. Причем эти самые значения не остаются постоянными: они очень быстро изменяются, и этот процесс непрерывен.

Если же говорить о дискретном способе, то в этом случае величина может принимать только ограниченное количество значений. При этом изменение происходит скачкообразно. Закодировать дискретно можно не только звуковую, но и графическую информацию. Что касается и аналоговой формы, кстати.

Аналоговая звуковая информация хранится на виниловых пластинках, например. А вот компакт-диск уже является дискретным способом представления информации звукового характера.

В самом начале мы говорили о том, что компьютер воспринимает всю информацию на машинном языке. Для этого информация кодируется в форме последовательности электрических импульсов – нулей и единиц. Кодирование звуковой информации не является исключением из этого правила. Чтобы обработать на компьютере звук, его для начала нужно превратить в ту самую последовательность. Только после этого над потоком или единичным звуком могут совершаться операции.

Когда происходит процесс кодирования, поток подвергается временной дискретизации. Звуковая волна непрерывна, она развивается на малые участки времени. Значение амплитуды при этом устанавливается для каждого определенного интервала отдельно.

Итак, что же мы выяснили в данной главе? Во-первых, абсолютно вся информация, которая выводится на монитор компьютера, прежде чем там появиться, подвергается кодированию. Во-вторых, это кодирование заключается в переводе информации на машинный язык. В-третьих, машинный язык представляет собой не что иное, как последовательность электрических импульсов – нулей и единиц. В-четвертых, для кодирования различных символов существуют отдельные таблицы. И, в-пятых, представить графическую и звуковую информацию можно в аналоговом и дискретном виде. Вот, пожалуй, основные моменты, которые мы разобрали. Одной из дисциплин, изучающей данную область, является информатика. Кодирование текстовой информации и его основы объясняются еще в школе, поскольку ничего сложного в этом нет.

3.2 ASCII, символы: описание, таблица кодов и виды

Как известно, компьютер хранит информацию в двоичном виде, представляя её в качестве последовательности единиц и нулей. Чтобы перевести информацию в форму, удобную для человеческого восприятия, каждая уникальная последовательность цифр при отображении заменяется на соответствующий ей символ.

Одной из систем соотнесения бинарных кодов с печатными и управляющими символами является кодировка ASCII.

При сегодняшнем уровне развития компьютерных технологий от пользователя не требуется знание кода каждого конкретного символа. Однако общее понимание того, как осуществляется кодирование, является крайне полезным, а для некоторых категорий специалистов и вовсе необходимым.

В первоначальном виде кодировка была разработана в 1963 году и затем в течение 25 лет дважды обновлялась.

В исходном варианте таблица символов ASCII включала 128 символов, позже появилась расширенная версия, где первые 128 знаков были сохранены, а кодам с задействованным восьмым битом поставлены в соответствие отсутствовавшие ранее символы.

На протяжении многих лет данная кодировка являлась самой популярной в мире. В 2006 году ведущее место заняла Latin 1252, а с конца 2007 года по настоящее время лидирующую позицию прочно держит Юникод.

Каждый ASCII-символ имеет собственный код, состоящий из 8 знаков, представляющих собой нуль или единицу. Минимальным числом в таком представлении является нуль (восемь нулей в двоичной системе), который и является кодом первого элемента в таблице[\[10\]](#).

Максимальным двоичным кодом в исходной версии ASCII является нуль + семь единиц, а в расширенной версии – восемь единиц, поскольку подключается восьмой бит.

Управляющими символами называются знаки, не имеющие графического представления и используемые для организации текста, управления устройствами и т. д. Они могут обозначать начало или конец текста, табуляцию, генерацию звукового сигнала, различные операции для работы по телетайпу (телетайп – машина для передачи данных по электрическом канале), разрешение вывода данных на устройстве, отмену действия и др.

В таблице символов ASCII позиции с 0 по 31, а также 127 выделены под управляющие знаки. Всего таких знаков 33.

Оставшиеся 95 позиций отведены под знаки препинания и знаки математических операций, десятичные цифры, буквы латинского алфавита, различающиеся регистром: «А» прописной и «а» строчной соответствуют различные коды в таблице символов ASCII.

Если человек занимается разработкой программного обеспечения или выполняет некоторые другие задачи из области информационных технологий, ему необходимо знать номера ряда символов ASCII.

Как упоминалось выше, позиции 0-31 и 127 занимают управляющие символы. Номер 32 закреплен за пробелом, номера 33-47 и 58-64 отведены под знаки препинания и основные математические операции.

Латинские прописные буквы расположены по алфавиту и имеют номера с 65-го по 90-й. Строчные буквы также расположены по алфавиту, их позиции – с 97-й по 122-ю. Оставшиеся номера (91-96 и 123-126) закреплены за квадратными и фигурными скобками, косой и прямой чертой, а также некоторыми диакритическими знаками.

Поскольку исходный вариант кодировки разрабатывался в расчете на американского пользователя, в нём не были предусмотрены не только различные виды письменности и национальные алфавиты, но даже удобное использование диакритических знаков, активно употребляющихся в европейских языках.

Для формирования расширенной кодировки был задействован 8-й бит. Данная версия включает в себя символы национальных европейских алфавитов и фонетической транскрипции, элементы графики, использовавшиеся для рисования таблиц, ряд математических знаков.

Некоторые ASCII-символы сегодня используются редко. В частности, это касается знаков, служащих для рисования таблиц, поскольку за годы, прошедшие с момента разработки расширенной кодировки, были внедрены гораздо более удобные способы графического представления таблиц.

До появления расширенного варианта ASCII для отображения национальных алфавитов использовались переработанные версии кодировки, где на месте латинских букв располагались русские, греческие, арабские символы.

Два кода в таблице были отведены под переключение между стандартной US-ASCII и её национальным вариантом[11].

После того как ASCII стала включать не 128, а 256 знаков, распространение получил вариант кодировки, при котором исходная версия таблицы была сохранена в первых 128 кодах с нулевым 8-м битом. Знаки национальной письменности хранились в верхней половине таблицы (128-255-я позиции).

Знать непосредственно коды символов ASCII пользователю не требуется. Разработчику программного обеспечения обычно достаточно знать номер элемента в таблице, чтобы при необходимости рассчитать его код, используя бинарную систему.

После разработки в начале 70-х годов кодировок для скандинавских языков, китайского, корейского, греческого и др., созданием собственного варианта занялся и Советский Союз. Вскоре был разработан вариант 8-битной кодировки под названием КОИ8, сохраняющей первые 128 кодов символов ASCII и выделяющей столько же позиций под буквы национального алфавита и дополнительные знаки.

До внедрения Юникода КОИ8 доминировала в российском сегменте интернета. Существовали варианты кодировки как для русского, так и для украинского

алфавита.

Поскольку количество элементов даже в расширенной таблице не превышало 256, возможность вмещения в одну кодировку нескольких различных письменностей отсутствовала. В 90-е годы в Рунете появилась проблема «крокозябр», когда тексты, набранные русскими символами ASCII, отображались некорректно.

Проблема заключалась в несоответствии кодов различных вариантов ASCII друг другу. Вспомним, что на позициях 128-255 могли располагаться различные знаки, и при смене одной кириллической кодировки на другую все буквы текста заменялись на другие, имеющие идентичный номер в другой версии кодировки.

С появлением Юникода популярность ASCII резко пошла на убыль.

Причина этого кроется в том факте, что новая кодировка позволила вместить знаки почти всех письменных языков. При этом первые 128 символов ASCII соответствуют тем же символам в Юникоде.

В 2000-м ASCII была самой популярной кодировкой в интернете и использовалась на 60 % проиндексированных «Гуглом» веб-страниц. Уже к 2012 году доля таких страниц упала до 17 %, а место самой популярной кодировки занял Юникод (UTF-8).

Таким образом, ASCII является важной частью истории информационных технологий, однако её использование в дальнейшем видится малоперспективным.

Заключение

В ходе курсовой работы была рассмотрена задача кодирования, которая включает в себя:

- изучение способов представления информации в ЭВМ;
- рассмотрение способов кодирования и декодирования информации числовой, текстовой и графической;
- подробное исследование кодирования текстовой информации в компьютере.

Задача кодирования является одним из главных понятий информатики, так как кодирование предшествует передаче и хранению информации, и, соответственно, является основой их успешного осуществления.

При передаче сообщений по каналам связи могут возникать помехи, способные привести к искажению принимаемых знаков. Эта проблема решается с помощью помехоустойчивого кодирования. Помехоустойчивое кодирование передаваемой информации позволяет в приемной части системы обнаруживать и исправлять ошибки. Коды, применяемые при помехоустойчивом кодировании, называются корректирующими кодами. Впервые, исследование эффективного кодирования произвел Клод Шеннон. Для теории связи важнейшее значение имеют две теоремы, доказанные Шенноном.

В работе были рассмотрены эти теоремы, и можно прийти к выводу, что первая – затрагивает ситуацию с кодированием при передаче сообщения по линии связи, в которой отсутствуют помехи, искажающие информацию, т.е. эта теорема является эталоном, какими должны быть помехоустойчивые коды, вторая теорема относится к реальным линиям связи с помехами.

В ходе курсовой работы были составлены примеры кодирования, на основе первой теоремы Шеннона. Это кодирование является достаточно эффективным, так как получаемый код практически не имеет избыточности, но, к сожалению, в реальных линиях связи множество помех, и такой результат недостижим. Поэтому код Шеннона не является таким же эффективным как, например, код Хаффмена. Но, несмотря на это нужно отметить, что Клод Шеннон был одним из основателей теории кодирования и его работы внесли огромный вклад в развитие информатики.

Библиография

Нормативно-правовые документы

1. Конституция Российской Федерации от 12 декабря 1993 г. (с изм. и доп., вступ. в силу с 21.07.2014)

Научная литература

1. Босова Л.Л Информатика и ИКТ 6 класс [Текст]: Учебник/ Босова Л.Л .- М.: Изд-во "БИНОМ. Лаборатория знаний", 2016. - 208 с.;

2. Босова Л.Л Информатика и ИКТ 7 класс [Текст]: Учебник/ Босова Л.Л.- М.: Изд-во "БИНОМ. Лаборатория знаний", 2018. - 229 с.; .
3. Куликовский Л.Ф., Мотов В.В. Теоретические основы информационных процессов: Учеб. пособие для вузов по спец. "Автоматика и механизация процессов обработки и выдачи информации". - М.:Высшая школа, 2017. - 248с.
4. В.М. Лачинов, А.О.Поляков Информодинамика или Путь к Миру открытых систем <http://www.inftech.webservis.ru/it/information/index.html>
5. Энциклопедия "Кругосвет" www.krugosvet.ru
6. Инженер Мареев Новая теория информации www.eme.ru
7. Лидовский В.В. Теория информации. Учебное пособие. - М.: Компания Спутник+, 2014г.
8. Вернер М. Основы кодирования Учебник для вузов. - М.: Техносфера,2018. - 288с.
9. Котоусов А.С. Теоретические основы радиосистем. Радиосвязь, радиолокация, радионавигация. - М.: Радио и связь, 2014. - 224с.
10. Гаранин М.В., Журавлев В.И., Кунегин С.В. Системы и сети передачи информации: Учеб. пособие для вузов. - М.: Радио и связь, 2017. - 336с.
11. Липкин И.А. Статистическая радиотехника. Теория информации и кодирования. - М.: "Вузовская книга", 2012. - 216с.
12. Быховский М.А. Круги памяти (Очерки истории развития радиосвязи и вещания в XX столетии). Серия изданий "История электросвязи и радиотехники". - М.: МЦНТИ, ООО "Мобильные коммуникации", 2014. - 224с.
13. Рузайкин Г. Теория информации и информационные технологии //Открытые системы, 2018, № 07-08
14. 12. Просиз Дж. Цифровая связь. Пер. с англ. /Под ред. Д.Д.Кловского. - М.: Радио и связь, 2019. - 800с.
15. Шульгин В.И. Основы теории передачи информации. Учебное пособие. Ч.1-2. - Харьков: ХАИ, 2013.
16. Котоусов А.С. Теоретические основы радиосистем. Радиосвязь, радиолокация, радионавигация. - М.: Радио и связь, 2012
17. Скляр Б. Цифровая связь. Теоретические основы и практическое применение, 2-е издание. : Пер. с англ. - М.: Издательский дом "Вильямс", 2018. - 1104с.
18. Семакин И.Г. Информатика и ИКТ для 8-9 классов [Текст]: Учеб. по базовому курсу / Семакин И.Г., Залогова Л.А, Русаков С.В., Шестакова Л.В.. М.: Изд-во "БИНОМ. Лаборатория знаний", 2019. - 320 с.; .
19. Угринович Н.Д «Информатика и ИКТ» Базовый курс. 9 класс [Текст]: Учебник/ Угринович Н.Д.- М.: Изд-во "БИНОМ. Лаборатория знаний", 2017. 295 с.;

20. Могилев А.В. Информатика [Текст]: Учеб. для студентов пед. вузов/ Могилев А.В., Пак Н.И., Хённер Е.К. М.: Академия, 2014. - 848 с.
 21. Подласый И. П. Педагогика. Новый курс [Текст]: Учеб. для студентов пед. вузов/ Подласый И. П. - М.: Гуманит. изд. центр ВЛАДОС, 2018.
 22. Бочкин, А.И. Методика преподавания информатики [Текст] / Бочкин А.И.. - Минск : Высшая школа, 2014. - 431 с.
-
1. Рузайкин Г. Теория информации и информационные технологии //Открытые системы, 2018, № 07-08 [↑](#)
 2. Котоусов А.С. Теоретические основы радиосистем. Радиосвязь, радио-локация, радионавигация. - М.: Радио и связь, 2014. - 224с. [↑](#)
 3. Лидовский В.В. Теория информации. Учебное пособие. - М.: Компания Спутник+, 2014г. [↑](#)
 4. Куликовский Л.Ф., Мотов В.В. Теоретические основы информацион-ных процессов: Учеб. пособие для вузов по спец. "Автоматика и меха-низация процессов обработки и выдачи информации". - М.:Высшая школа, 2017. - 248с. [↑](#)
 5. Подласый И. П. Педагогика. Новый курс [Текст]: Учеб. для студен-тов пед. вузов/ Подласый И. П. - М.: Гуманит. изд. центр ВЛАДОС, 2018. [↑](#)
 6. Угринович Н.Д «Информатика и ИКТ» Базовый курс. 9 класс [Текст]: Учебник/ Угринович Н.Д.- М.: Изд-во "БИНОМ. Лаборатория знаний", 2017. 295 с.; [↑](#)
 7. Быховский М.А. Круги памяти (Очерки истории развития радио-связи и вещания в XX столетии). Серия изданий "История электросвязи и радиотехники". - М.: МЦНТИ, ООО "Мобильные коммуникации", 2014. - 224с. [↑](#)
 8. Скляр Б. Цифровая связь. Теоретические основы и практическое применение, 2-е издание. : Пер. с англ. - М.: Издательский дом "Виль-ямс", 2018. - 1104с. [↑](#)

9. Могилев А.В. Информатика [Текст]: Учеб. для студентов пед. вузов/ Могилев А.В., Пак Н.И., Хённер Е.К. М.: Академия, 2014. - 848 с. [↑](#)
10. Подласый И. П. Педагогика. Новый курс [Текст]: Учеб. для студентов пед. вузов/ Подласый И. П. - М.: Гуманит. изд. центр ВЛАДОС, 2018. [↑](#)
11. В.М. Лачинов, А.О.Поляков Информодинамика или Путь к Миру от-крытых систем <http://www.inftech.webservis.ru/it/information/index.html> [↑](#)