# Тема 2. Измерение информации. Представление информации в компьютере

## Подходы к измерению информации

1. **Алфавитный (объемный) подход**

Наиболее простой подход, измеряющий количество информации как число знаков (сигналов) в сообщении, без учета ее смысла.

Электронная информация измеряется в битах, байтах, килобайтах и т.д. В общем случае можно использовать любые единицы измерения – количество слов, букв, страниц или любой другой показатель, который показывает сколько «места» занимает информация.

1. **Вероятностный (энтропийный) подход**

Используется в теории информации. Получение информации предполагает увеличение объема знания и уменьшение незнания, неопределенности, т.е. **энтропии**. В то же время полезность этого знания не оценивается.

За **единицу количества информации** **(бит)** принимается выбор одного из двух равновероятных вариантов («да» или «нет», «1» или «0», «ложь» или «истина»).

Если вариантов больше двух, используется формула **Хартли:**



*I* – информационный объем;

*N* – количество равновероятных вариантов выбора.

Если *N* = 2 вариантам, то *I* = 1 бит. Если N = 1 (всего один возможный вариант), то I = 0 (ответ очевиден, его подтверждение не дает новой информации).

Пример. В группе 16 студентов. Преподаватель наугад спрашивает одного из них. Каков информационный объем знания о том, кого именно спросили?

N = 16 I = log2 16 = 4 (24 = 16).

Ответ: 4 бита.

На практике варианты ответа обычно имеют разную вероятность. Тогда:

**,**

*hi* – информационный объем *i*-того варианта (его информативность);

*pi* – вероятность появления *i*-того варианта.

Пример. Покупатель может приобрести или не приобрести определенный товар. Но вероятность этих событий неодинакова. Поэтому узнав, купил он товар, или не купил, вы получите разное количество информации, не равное 1 биту.

Пусть вероятность покупки составляет 80%, а «не покупки» – 20%. Какое сообщение информативнее:

а) покупатель купил товар;

б) покупатель не купил товар?

а) p1 = 0,8 h1 = –log2 0,8 ≈ 0,322

б) p2 = 0,2 h2 = –log2 0,2 ≈ 2,322

Ответ: информативнее вариант с «не покупкой» товара.

Т.е., чем меньше вероятность появления варианта выбора, тем больше его информативность. Чем ближе вероятность к 1, тем ближе информативность к 0. Чем ближе вероятность к 0, тем ближе информативность к бесконечности.

Пример. Сообщение о том, что завтра будут занятия для студента не очень информативно (высокая вероятность события – он и так об этом знает). Но если сообщить студенту, что завтра пары отменяются (маленькая вероятность события) – такое сообщение содержит больше информации.

1. **Семантический (смысловой) подход**

Наименее формализованный подход. Связан с субъективной, эмоциональной оценкой полученной информации. Оценивается новизна, интересность информации. Оценку дает *получатель,* поэтому объемзависит не только от самой информации, но и от качеств получателя. Оцениваться может и синтаксис, и семантика.

1. **Прагматический подход**

связана с определением количества информации, необходимым для достижения поставленной цели и определяет степень ценности, полезности информации. Ценность информации обычно измеряют в тех же единицах, что и целевая функция (руб. в экономике, м/с – скорость космического аппарата, оценка для студента, число голосов на выборах и т.д.).

Например, в экономической системе прагматическую ценность ИТ можно определить приростом экономического эффекта (прибыли, рентабельности) от внедрения этой технологии.

## Единицы измерения информации. Представление чисел

Минимальной единицей информации является **бит**. Бит может быть равен 0 или 1, но можно обозначить эти два варианта и по-другому: да/нет, вкл./выкл., истина/ложь, true/false. Физически это выражается в двух различных состояниях устройств: есть ток или нет тока, намагничено – не намагничено, отражает свет – не отражает свет.

Это означает, что в компьютере используется *двоичная система счисления* с двумя цифрами {0; 1}. Для человека наиболее привычна *десятичная система* с десятью цифрами {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}. Любое десятичное число можно перевести в двоичную систему:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 010 | 02 |  | 810 | 10002 |  | 1610 | 1 00002 |
| 110 | 12 |  | 910 | 10012 |  | 1710 | 1 00012 |
| 210 | 102 |  | 1010 | 10102 |  | 1810 | 1 00102 |
| 310 | 112 |  | 1110 | 10112 |  | 1910 | 1 00112 |
| 410 | 1002 |  | 1210 | 11002 |  | 2010 | 1 01002 |
| 510 | 1012 |  | 1310 | 11012 |  | 2110 | 1 01012 |
| 610 | 1102 |  | 1410 | 11102 |  | 2210 | 1 01102 |
| 710 | 1112 |  | 1510 | 11112 |  | ... |  |

Можно заметить, что числа, равные степеням двойки (4=22, 8=23, 16=24) записываются как 1 и столько же нулей, что и степень 2. Например:

25610 = 28 = 1 0000 00002

А числа перед ними состоят из одних единиц:

310 = 22 – 1 = 112

710 = 23 – 1 = 1112

1510 = 24 – 1 = 11112

Исторически сложилось, что биты объединяются в группы по 8. Группа из 8 бит называется **байт**. Байт является минимальной адресуемой ячейкой памяти компьютера. Т.е. вы не можете записать в память один бит, только целый байт.

Один байт позволяет хранить числа от 0 до 255:

0000 00002 = 010

0000 00012 = 110

...

1111 11112 = 28 – 1 = 25510

Байт – это очень маленькая единица информации. Чтобы записать число жителей России (142 млн. чел.) потребуется 4 байта.

00001000 01110110 10111111 10000000

Для измерения информации используются более крупные единицы – килобайты, мегабайты, гигабайты и др.

Однако с этими приставками существует небольшая путаница. Дело в том, что

210 = 1024 ≈ 1000 =103.

В физике «кило-» означает 1000, но в информатике под килобайтом или килобитом подразумевается 1024 байта или бита соответственно. Из-за этого, например, флешка, на которой написано «4ГБ» в компьютере будет отображаться как 3,8ГБ. В России во избежание в соответствующем ГОСТе указаны различные сокращения: ГБ = 1000 байт, а Гбайт = 1024 байт.

**Двоичные приставки**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Приставка | По степеням 10 (СИ/ГОСТ) | | По степеням 2 (ГОСТ) | |
| кило- | 103Б | кБ | 210Б = 1024 Б | Кбайт |
| мега- | 106Б | МБ | 220Б ≈ 1,05∙106Б | Мбайт |
| гига- | 109Б | ГБ | 230Б ≈ 1,07∙109Б | Гбайт |
| тера- | 1012Б | ТБ | 240Б ≈ 1,10∙1012Б | Тбайт |
| пета- | 1015Б | ПТ | 250Б ≈ 1,13∙1015Б | Пбайт |
| экса- | 1018Б | ЭБ | 260Б ≈ 1,15∙1018Б | Эбайт |
| зетта- | 1021Б | ЗБ | 270Б ≈ 1,18∙1021Б | Збайт |
| йотта- | 1024Б | ЙБ | 280Б ≈ 1,21∙1024Б | Йбайт |

На сегодня йоттабайт – самая крупная единица измерения информации. По оценкам экспертов, в 2011г. в мире накоплено в общей сложности 1,8 ЗБ информации, т.е. около 1 800 000 000 000 000 000 000 байт.

Еще одна путаница связана с тем, что скорость передачи информации в сети (доступа к Интернету) измеряется в битах (килобитах, мегабитах). Т.е. «Интернет на скорости 35Мбит/с» – это около 4МБ/с, в 8 раз меньше.

Компьютер не умеет работать ни с чем, кроме двоичных чисел. Поэтому вся остальная информация – текст, даты, рисунки, звук, видео – *кодируется* в числовом виде.

**Кодирование дат и времени**

Любая дата вместе со временем (с точностью до мс) представляется в виде числа дней, прошедших с 1 января 1900г. Поэтому даты можно вычитать одну из другой, получится число дней между ними. Но если ошибиться с форматом представления, то вместо даты отобразится некоторое большое число. Например, 1 января 2000г. = 36526.

**Кодирование текста**

Каждый символ текста представляется в виде определенного числа (кода). Это соответствие записано в виде таблицы.

Наиболее распространенной такой таблицей является **ASCII** (читается «аски»). В ней каждому символу ставится в соответствие 1 байт, т.е. всего она содержит 256 символов (от 0 до 255).

Вот ее фрагмент:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Код** | **32** | **33** | **34** | **35** | **36** | **37** | **38** | **39** | **40** | **41** | **42** | **43** | **44** | **45** | **46** | **47** |
| **Символ** | пробел | ! | " | # | $ | % | & | ' | ( | ) | \* | + | , | - | . | / |
| **Код** | **48** | **49** | **50** | **51** | **52** | **53** | **54** | **55** | **56** | **57** | **58** | **59** | **60** | **61** | **62** | **63** |
| **Символ** | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | : | ; | < | = | > | ? |
| **Код** | **64** | **65** | **66** | **67** | **68** | **69** | **70** | **71** | **72** | **73** | **74** | **75** | **76** | **77** | **78** | **79** |
| **Символ** | @ | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O |
| **Код** | **80** | **81** | **82** | **83** | **84** | **85** | **86** | **87** | **88** | **89** | **90** | **91** | **92** | **93** | **94** | **95** |
| **Символ** | P | Q | R | S | T | U | V | w | X | Y | Z | [ | \ | ] | ^ | \_ |

Любой ASCII-символ можно ввести с клавиатуры через код. Нужно зажать клавишу Alt и, не опуская ее, на дополнительной цифровой клавиатуре (справа) набрать код (четырьмя цифрами). Например, Alt+0167 = §, Alt+0169 = ©. Так можно вводить символы, которых нет на клавиатуре.

Однако в мире очень много языков с различными символами, все они в эту таблицу не уместятся. Проблему решили следующим образом.

Первая половина таблицы (0-127) стала универсальной, международной. В ней содержатся английские буквы, цифры, знаки препинания, арифметические действия, пробел, а также *непечатаемые символы* – «удаление», «переход на новую строку», «выход» и др.

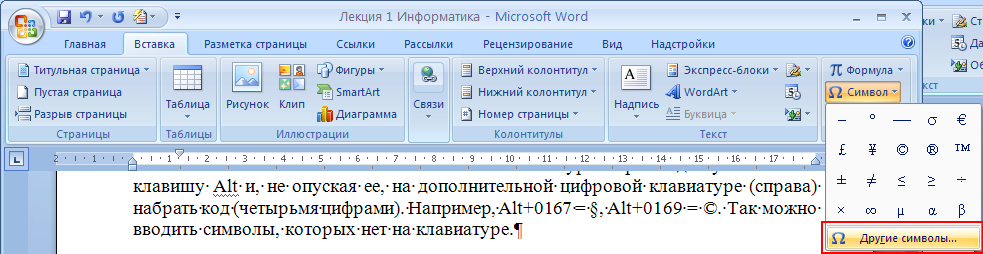
Вторая половина таблицы – национальная. Каждый сам решает, что и как в нее записывать. Например, существуют варианты кодировки кириллицы – Windows-1251 (американская), КОИ8 (российская, сейчас выходит из употребления). Для кодирования расширенной латиницы (с точками, ударениями над буквами) используется Windows-1252.

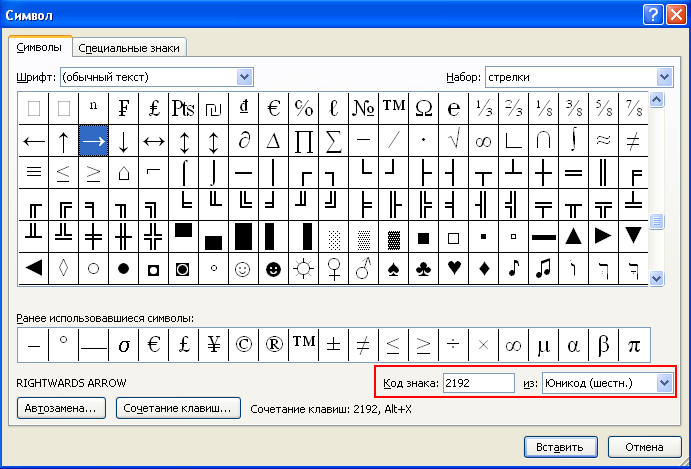
К сожалению, это приводит к тому, что в тексте нельзя использовать одновременно несколько разных языков, кроме английского. Из-за неправильной кодировки вместо текста на экране будут отображаться «зюки» и «кракозябры».

Поэтому была создана таблица **Unicode** (читается «юникод»).

В ней каждый символ кодируется двумя байтами, всего в нее можно записать 216 = 65 536 символов. Этого оказалось достаточно для всех современных языков Земли (по крайней мере, использующихся в компьютерах). В таблице даже осталось немного места «про запас», на будущее.

Все символы Unicode можно найти в MS Word.





В Word символ можно набрать, сначала введя его код, а потом нажав Alt+X. Так можно поставить ударение: 0301, Alt+X = ́, 0300, Alt+X = ̀. Ударение смещается на предыдущую букву (например, замо́к).

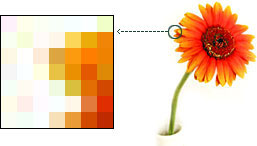
**Кодирование графики**

Существует два основных способа представления рисунков в компьютере:

* растровый
* векторный.



В *растровой* графике каждый рисунок представляется в виде точек (пикселей). Каждая точка имеет свой цвет. Так лучше всего хранить фотографии, художественные картины и другие реалистичные изображения с плавными переходами и большим количеством цветов.



В *векторной* графике используются линии и кривые, которые компьютер обрабатывает как математические формулы.

Например, формула окружности:

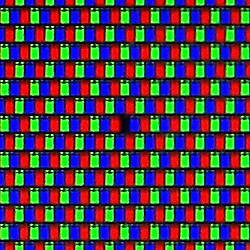
(x – x0)2 + (y – y0)2= R2

(x0, y0) – координаты центра, R – радиус.

Поэтому, когда в векторной графике рисуется круг, компьютер просто запоминает эти числа, цвет линии и заливки. Не нужно хранить каждый пиксель окружности, и можно увеличить ее на сколько угодно без потери качества.

В любом случае нужно закодировать **цвет**.

Основной способ кодировки цвета – **RGB** (red, green, blue – красный, синий зеленый). Все остальные цвета смешиваются из них. Так представляются точки на экране монитора. Под лупой он выглядит так:



Каждый цвет записывается одним байтом, т.е. всего каждый пиксель занимает три байта. Каждый цвет кодируется числом от 0 до 255.

Примеры:

|  |  |  |
| --- | --- | --- |
| RGB(0,0,0) | черный | ни один цвет не горит на экране |
| RGB(255,255,255) | белый | светятся все цвета |
| RGB(127,127,127) | серый | все цвета поровну, но неярко |
| RGB(255,0,0) | красный | только красный |
| RGB(255,255,0) | желтый | красный и зеленый |
| RGB(78,169,214) |  | сложный оттенок синего |

Всего так можно закодировать очень много цветов – 224 = 16,5млн. Это примерно соответствует числу оттенков, которые способен различить человеческий глаз.

Поскольку в каждой картинке пикселей очень много (тысячи и миллионы), растровые рисунки занимают много места, и для них используется *сжатие*. Без сжатия одна фотография занимала бы столько же места, как 1000 страниц текста (без оформления).

Пример. Для печати нужно не менее 120 пикселей в 1 см. Тогда в фото 10х15см без сжатия:

10 \* 120 = 1200 пикселей

15 \* 120 = 1800 пикселей

Всего 1200 \* 1800 = 2 160 000 пикселей

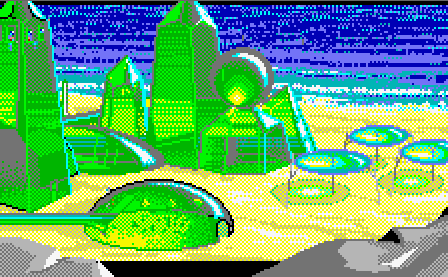
2 160 000 \* 3 = 6 480 000 байт

Замечание. Раньше, когда компьютеры еще не могли работать с такими большими объемами, обходились гораздо меньшим количеством цветов: 16, потом 256. Каждый цвет, как и символ, кодировался определенным числом, записанным в таблице – **палитре**.

Системная палитра:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Код* | *R* | *G* | *B* | *Название* | *Цвет* |
| 0 | 0 | 0 | 0 | Черный |  |
| 1 | 255 | 255 | 255 | Белый |  |
| 2 | 255 | 0 | 0 | Красный |  |
| 3 | 0 | 255 | 0 | Зеленый |  |
| 4 | 0 | 0 | 255 | Синий |  |
| 5 | 255 | 255 | 0 | Желтый |  |
| 6 | 0 | 255 | 255 | Голубой |  |
| 7 | 255 | 0 | 255 | Розовый (фуксин) |  |
| 8 | 128 | 0 | 0 | Бордовый |  |
| 9 | 0 | 128 | 0 | Темно-зеленый |  |
| 10 | 0 | 0 | 128 | Темно-синий |  |
| 11 | 128 | 128 | 0 | Оливковый |  |
| 12 | 0 | 128 | 128 | Морской волны |  |
| 13 | 128 | 0 | 128 | Фиолетовый |  |
| 14 | 128 | 128 | 128 | Серый |  |
| 15 | 255 | 128 | 128 | Серебристый |  |

Пример рисунка из 16 цветов:



Можно создать и свою собственную палитру, с цветами для конкретного рисунка. При этом даже с 16 цветами он будет выглядеть не очень страшно.

|  |  |
| --- | --- |
| http://upload.wikimedia.org/wikipedia/commons/6/69/Truecolor.png | 16,7 млн. цветов |
| http://upload.wikimedia.org/wikipedia/commons/f/ff/8_bit.png | 256 цветов |
| http://upload.wikimedia.org/wikipedia/commons/0/0d/4_bit.png | 16 цветов |

**Аудио** и **видео** кодируется множеством разных способов. Для них существуют **кодеки**. В кодеке как раз записывается, как именно нужно кодировать или раскодировать определенный формат аудио и видео.