Кодирование информации

1. Двоичное кодирование
2. Кодирование чисел и символов
3. Кодирование рисунков
4. Кодирование звука

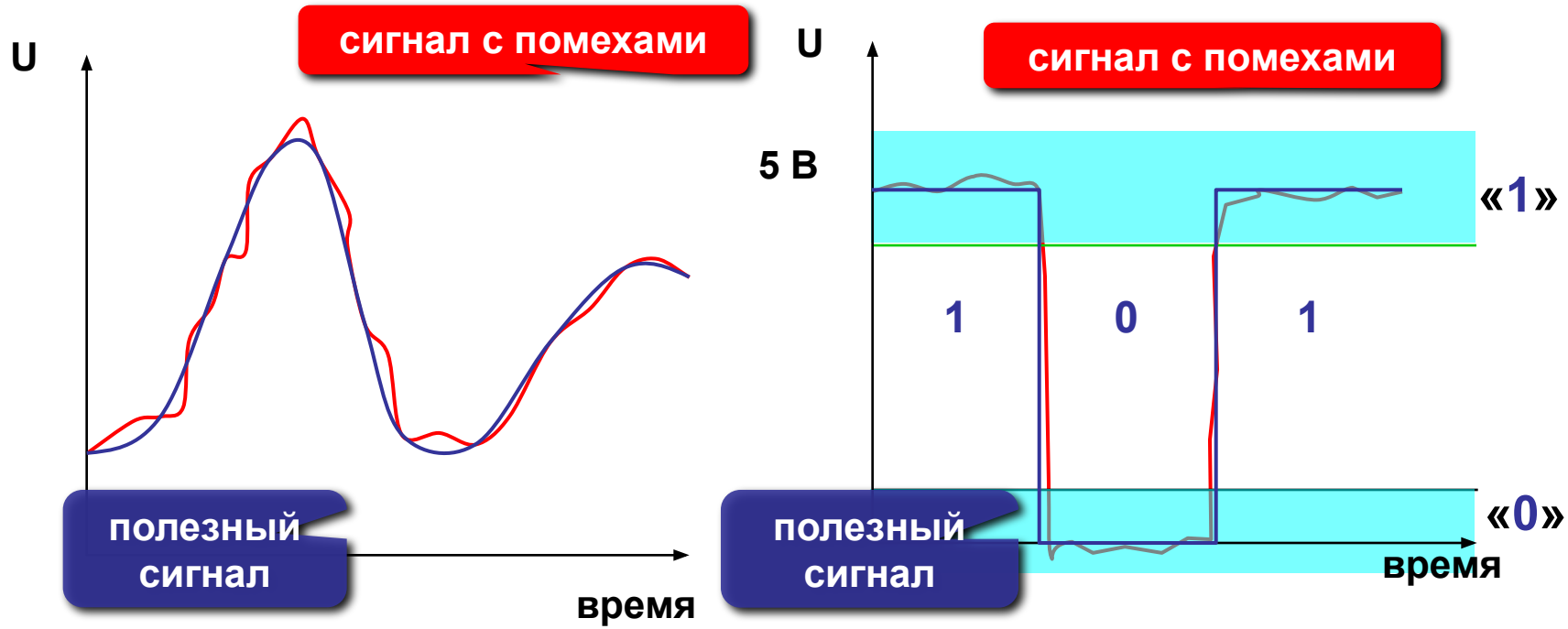
Кодирование информации

Тема 1. Двоичное кодирование

Двоичное кодирование

Двоичное кодирование – это кодирование всех видов информации с помощью двух знаков (обычно 0 и 1).

Передача электрических сигналов:



Двоичное кодирование

- в такой форме можно закодировать **все виды** информации
- нужны только устройства с **двумя состояниями**
- практически **нет ошибок** при передаче
- **компьютеру легче** обрабатывать данные

- **человеку сложно** воспринимать двоичные коды

Можно ли использовать не «0» и «1», а другие символы, например, «А» и «Б»?

Кодирование информации

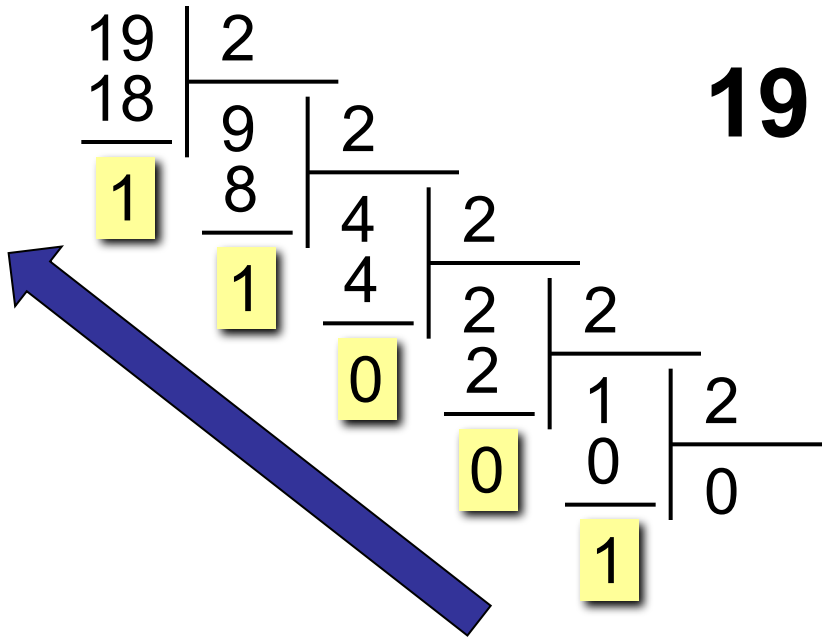
Тема 2. Кодирование чисел и символов

Кодирование чисел (двоичная система)

Алфавит: 0, 1

Основание (количество цифр): **2**

10 → 2



$19 = 10011_2$

система счисления

2 → 10

4 3 2 1 0 разряды

$$10011_2 = 1 \cdot 2^4 + \cancel{0 \cdot 2^3} + \cancel{0 \cdot 2^2} + 1 \cdot 2^1 + 1 \cdot 2^0$$

$$= 16 + 2 + 1 = 19$$

Кодирование символов

Текстовый файл

- на экране (символы)
- в памяти – двоичные коды

1000001_2	1000010_2	1000011_2	1000100_2
65	66	67	68

В файле хранятся не изображения символов, а их числовые коды в двоичной системе!

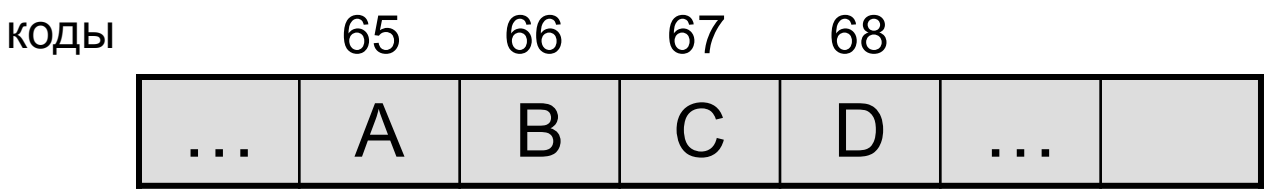
А где же хранятся изображения?

Кодирование символов

1. **Сколько символов** надо использовать одновременно? **256** или 65536 (UNICODE)
2. **Сколько места** надо выделить **на символ**:

256 = 2^8 ⇒ **8 бит на символ**

3. Выбрать **256 любых символов** (или 65536) - **алфавит**.
4. Каждому символу – **уникальный код 0..255** (или 0..65535). Таблица символов:



5. Коды – в **двоичную систему**.

Кодировка 1 байт на символ

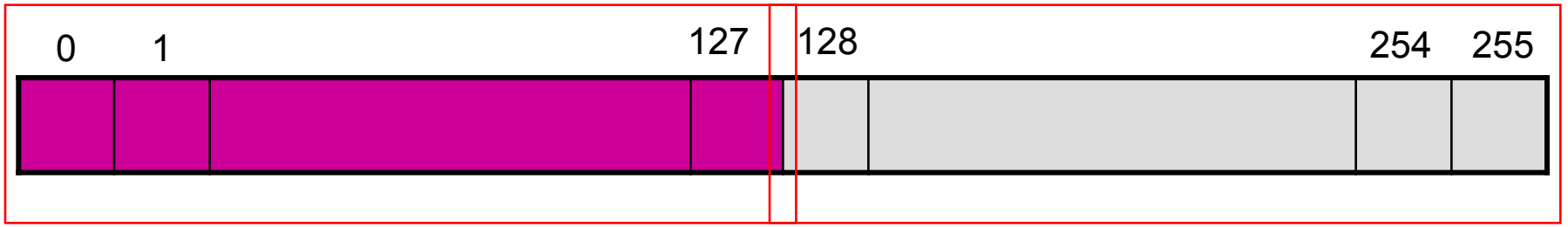


таблица ASCII (международная) **кодовая страница**

ASCII = *American Standard Code for Information Interchange*

0-31 управляющие символы:
7 – звонок, 10 – новая строка, 13 – возврат каретки, 27 – Esc.
32 пробел

знаки препинания: `. , : ; ! ?`

специальные знаки: `+ - * / () {} []`

48-57 цифры **0..9**
65-90 заглавные латинские буквы **A-Z**
97-122 строчные латинские буквы **a-z**

Кодовая страница (расширенная таблица ASCII)

для русского языка:

CP-866 для системы MS DOS
CP-1251 для системы *Windows*
КОИ8-R для системы UNIX (Интернет)

Кодировка UNICODE (2 байта на символ)

- ***Windows*, *MS Office*, …**
- **16 бит на символ**
- **65536 или 2^{16} символов в одной таблице**

- можно одновременно использовать символы разных языков

- размер файла увеличивается **в 2 раза**

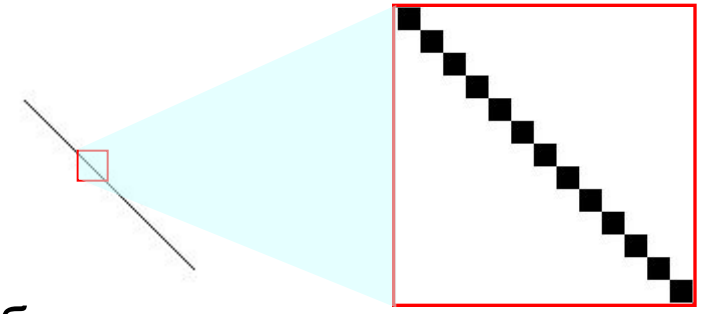
Кодирование информации

Тема 3. Кодирование рисунков

Два типа кодирования рисунков

- **растровое кодирование**

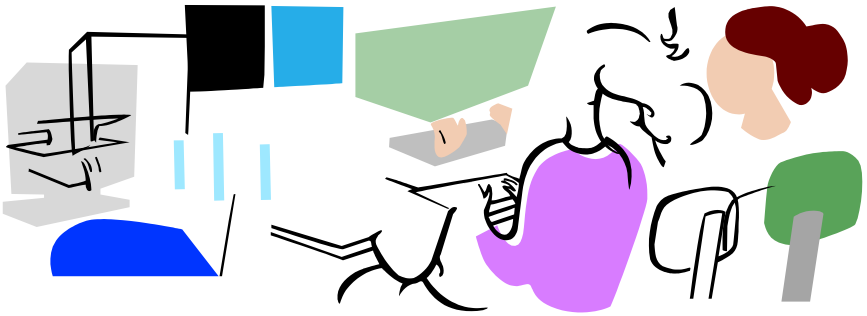
точечный рисунок, состоит из **пикселей**



фотографии, размытые изображения

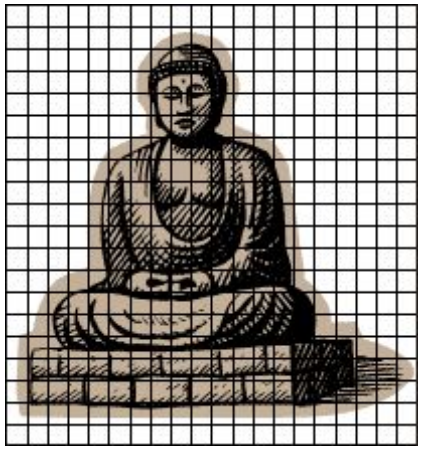
- **векторное кодирование**

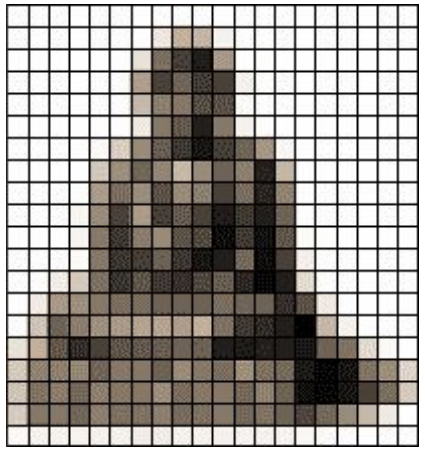
рисунок, состоит из **отдельных геометрических фигур**



чертежи, схемы, карты

Растровое кодирование





Шаг 1. Дискретизация: разбивка на *пиксели*.

Пиксель – это наименьший элемент рисунка, для которого можно независимо установить цвет.

Шаг 2. Для каждого пикселя определяется **единый цвет**.

Есть потеря информации!

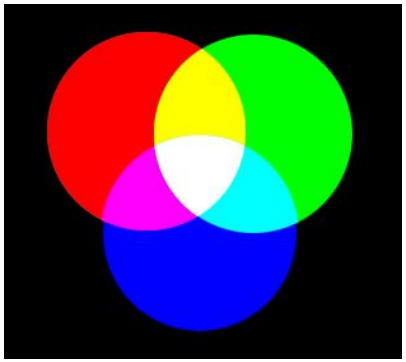
- **почему?**
- **как ее уменьшить?**

Разрешение: число пикселей на дюйм, *pixels per inch (ppi)*

экран 96 ppi, **печать 300-600** ppi, **типография 1200** ppi

Растровое кодирование *(True Color)*

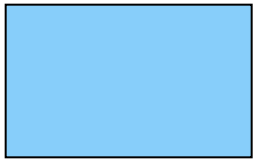
Шаг 3. От цвета – к числам: модель RGB



цвет = R + G + B

	red	***green***	***blue***
	красный	зеленый	синий
	0..255	0..255	0..255

R = 218
G = 164
B = 32



R = 135
G = 206
B = 250

Шаг 4. Числа – в двоичную систему.

Сколько разных цветов можно кодировать?

256·256·256 = **16 777 216** ***True Color***

Сколько памяти нужно для хранения цвета 1 пикселя?

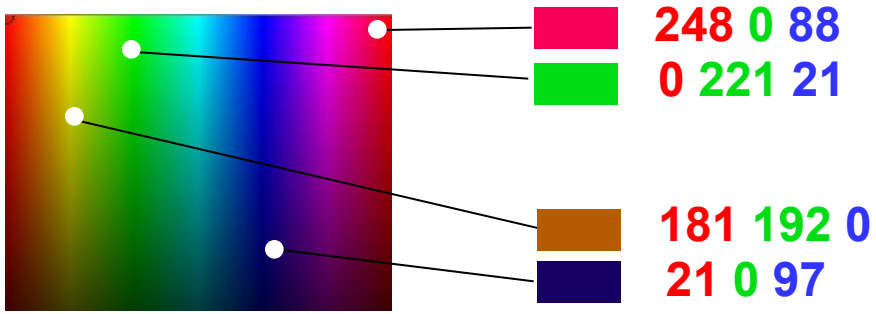
R: 256=2^8 вариантов, нужно 8 бит = 1 байт

R G B: всего 3 байта

Растровое кодирование с палитрой

Шаг 1. Выбрать количество цветов: 2, 4, … **256.**

Шаг 2. Выбрать 256 цветов из палитры:



Шаг 3. Составить палитру (каждому цвету – номер 0..255)

палитра хранится в начале файла

0	1		254	255
248 0 88	0 221 21	…	181 192 0	21 0 97

Шаг 4. Код пикселя = номеру его цвета в палитре

2	45	65	14	…	12	23

Растровое кодирование с палитрой

Файл с палитрой:

Сколько занимает палитра и основная часть?

Один цвет в палитре: 3 байта (RGB)

256 = 2^8 цветов:

палитра 256·3 = 768 байт
рисунок 8 бит на пиксель

16 цветов:

палитра 16·3 = 48 байт
рисунок 4 бита на пиксель

2 цвета:

палитра 2·3 = 6 байт
рисунок 1 бит на пиксель

Форматы файлов (растровые рисунки)

Формат	True Color	Палитра	Прозрачность
BMP	+	+	
JPG	+		
GIF		+	+
PNG	+	+	+

Растровые рисунки

- лучший способ для хранения **фотографий** и изображений без четких границ
- **спецэффекты** (тени, ореолы, и т.д.)

- есть **потеря информации** (почему?)
- при изменении размеров рисунка он **искажается**
- **размер файла** не зависит от сложности рисунка (а от чего зависит?)

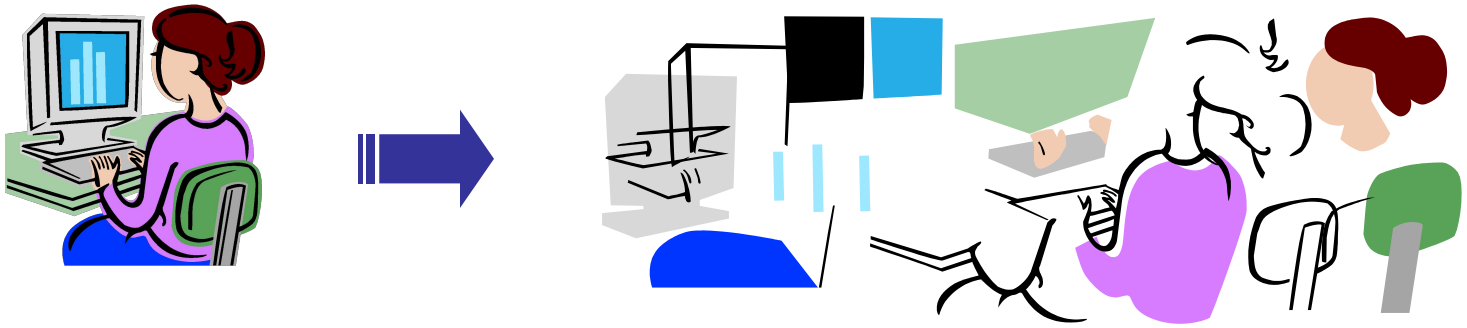
Векторные рисунки

Строятся из геометрических фигур:

- отрезки, ломаные, прямоугольники
- окружности, эллипсы, дуги
- сглаженные линии (кривые Безье)

Для каждой фигуры в памяти хранятся:

- размеры и координаты на рисунке
- цвет и стиль границы
- цвет и стиль заливки (для замкнутых фигур)



Форматы файлов:

- **WMF** *(Windows Metafile)*
- **CDR** *(CorelDraw)*
- **AI** *(Adobe Illustrator)*
- **FH** *(FreeHand)*

Векторные рисунки

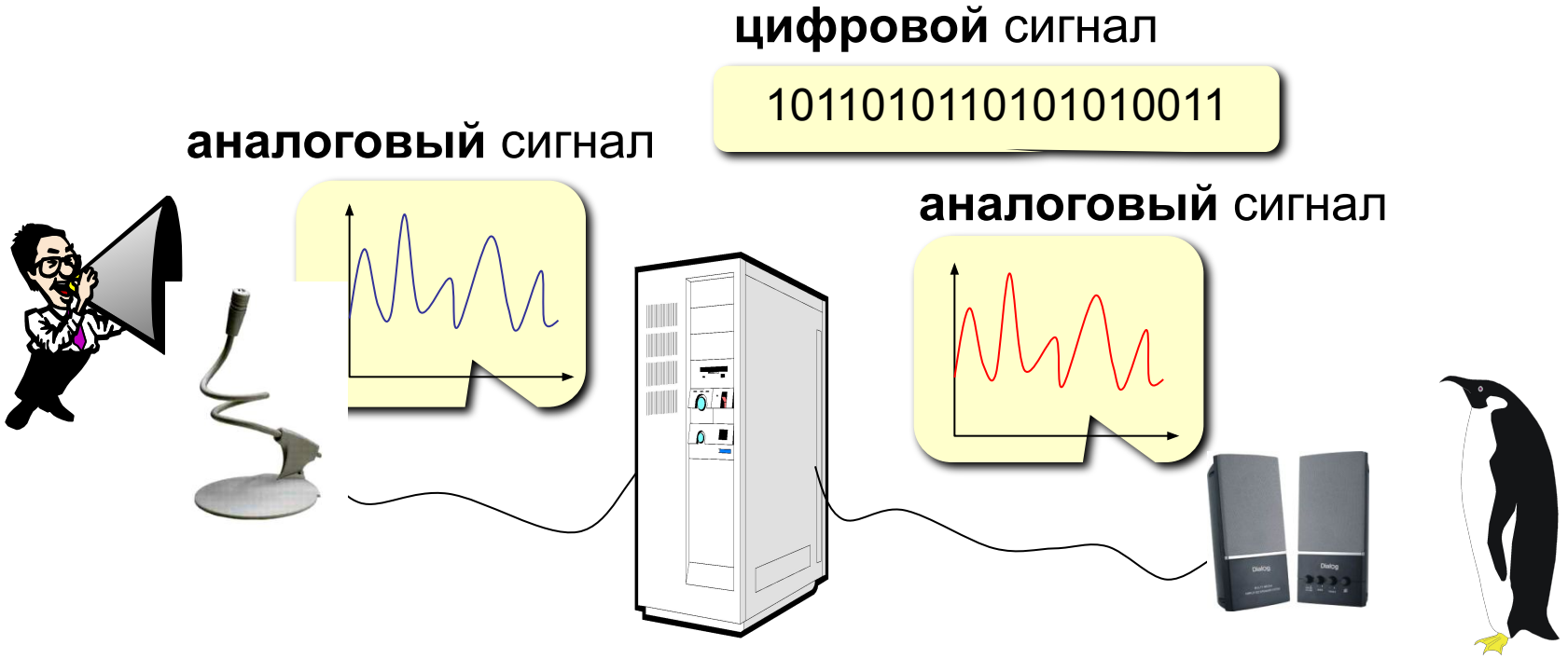
- лучший способ для хранения **чертежей, схем, карт;**
- при кодировании **нет потери информации**;
- при изменении размера **нет искажений**;
- меньше **размер файла**, зависит от сложности рисунка;

- неэффективно использовать для **фотографий** и размытых изображений

Кодирование информации

Тема 4. Кодирование звука

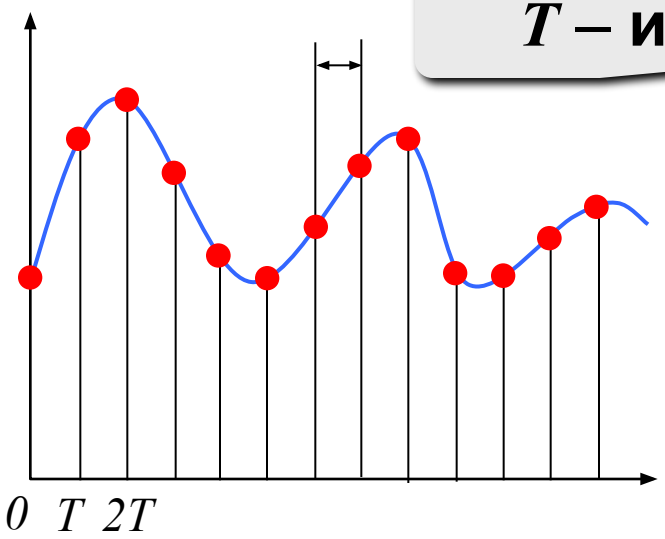
Оцифровка (перевод в цифровую форму)



- Какой объем информации в аналоговом сигнале?
- Можно ли хранить его в памяти реального устройства?
- Будет ли сигнал на выходе тот же самый?
- Почему есть потеря информации?

Дискретизация по времени

хранятся только значения сигнала в моменты 0, T, $2T$, …



T – интервал дискретизации

Частота дискретизации:

$$f = \frac{1}{T}$$

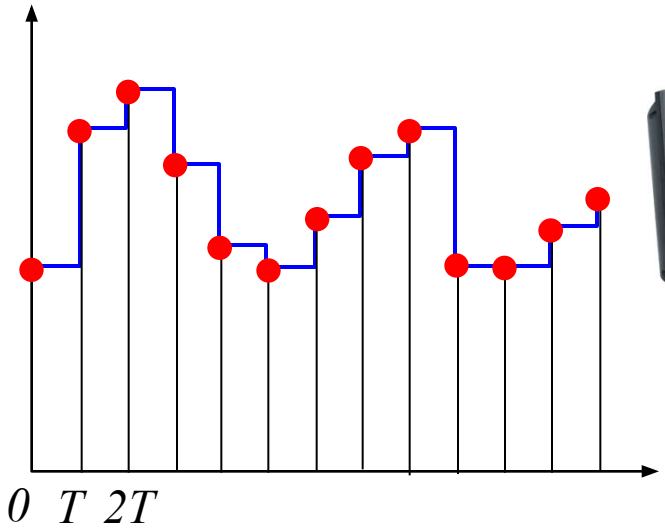
8 кГц, 11 кГц,
22 кГц, 44 кГц (CD)

22 кГц $T = \frac{1}{22000} \approx 0{,}00005$ с

Человек слышит 16 Гц … 20 кГц

Что компьютер может выдать на выход?

Как улучшить качество? Что при этом ухудшится?



Дискретизация по уровню

Сколько бит нужно, чтобы хранить число 0,7?

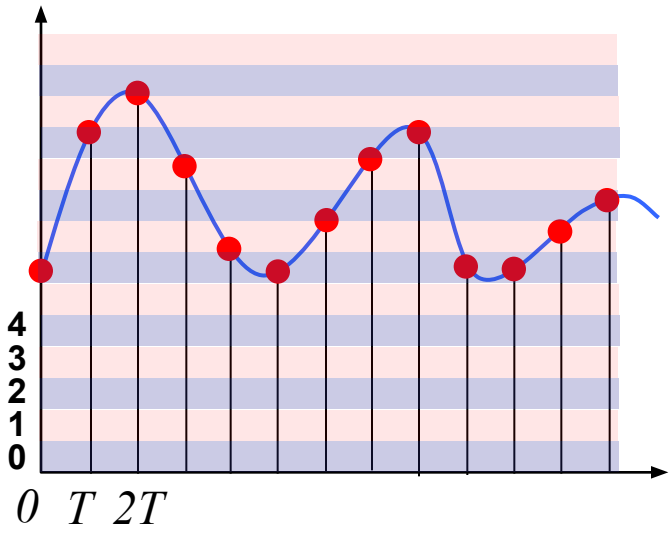
У всех точек в одной полосе одинаковый код!

8 бит = 256 уровней

16 бит = 65536 уровней

32 бита = 2^{32} уровней

64 бита = 2^{64} уровней



При оцифровке потерю информации дает дискретизация как по времени, так и по уровню!

Оцифровка – итог

- можно закодировать **любой звук** (в т.ч. голос, свист, шорох, …)

- есть **потеря информации**
- большой **объем файлов**

  44 кГц, 16 бит: 88 Кб/с, 5,3 Мб/мин

Форматы файлов:

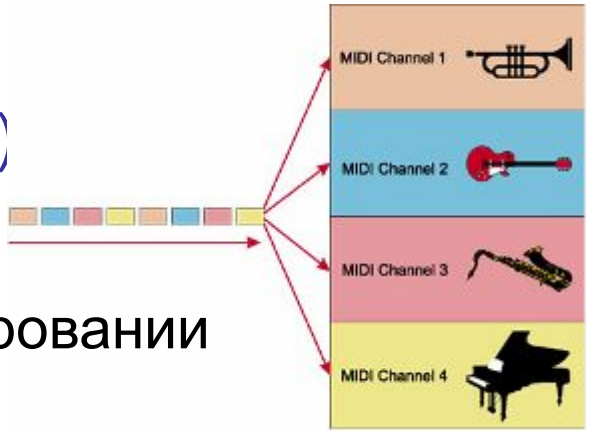
- **WAV** (*Waveform audio format),* часто без сжатия (размер!)
- **MP3** (*MPEG-1 Audio Layer 3*, сжатие с потерями)
- **WMA** (*Windows Media Audio,* потоковый звук, сжатие)

Инструментальное кодирование

MIDI (*Musical Instrument Digital Interface)*, файлы ***.MID**

в файле:

- нота (высота, длительность)
- музыкальный инструмент
- параметры звука (громкость, тембр)
- может быть несколько каналов



- **нет потери информации** при кодировании инструментальной музыки
- маленький **размер файлов**

- невозможно закодировать нестандартный звук, голос

MIDI-клавиатура: