

Двоичное кодирование текстовой информации

Вспомним, что в одном байте можно хранить десятичные числа от 0 до 255.

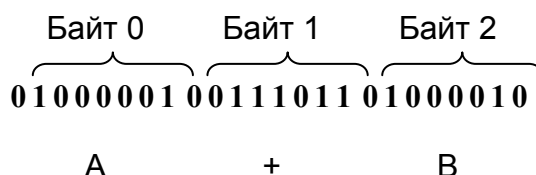
Этого количества вполне достаточно, чтобы дать уникальное восьмибитовое обозначение каждой заглавной и строчной букве двух алфавитов, всем цифрам, знакам препинания, некоторым другим необходимым символам, а также служебным кодам для передачи информации.

Для каждого символа можно указать соответствующий код – *закодировать* информацию. Соответствие между набором букв и числами называется **кодировкой символов**.

Например:

Символ	Код
A	0100 0001
B	0100 0010
+	0011 1011

Тогда в памяти машины выражение «**A+B**» запишется таким образом:



*Таблица, в которой каждой букве, цифре, служебному символу присвоен какой-либо код (от 0 до 255), называется **кодовой таблицей символов**.*

В системе **Unicode** для кодирования символа используются **16 бит**. Это позволяет закодировать **65 536** различных символов.

Двоичное кодирование числовой информации

При хранении информации в памяти машины каждый бит хранится в одном разряде памяти, а разряды объединяются в ячейки памяти фиксированного размера.

Ячейки размером 8, 16, 32 разряда носят названия: **байт, слово, двойное слово**.

Не всякое число может быть записано в одном байте.

Например, $1000_{10} = 11\ 1110\ 1000_2$.

Кроме того, для отрицательных чисел нужен знак *минус*. Таким образом, необходимо договориться о кодировании.

ПК производит действия над числами, размещенными в байтах, словах и двойных словах.

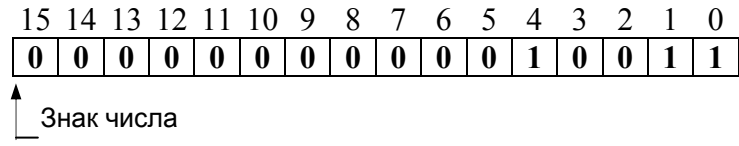
Бит не может быть пустым. Он всегда имеет значение 0 или 1. Байт, в котором записано число 0, будет состоять из нулевых битов: **00000000**.

Понятно, что максимальным числом без знака, записанным с помощью восьми битов будет **11111111**. В десятичной системе это число **255**. Значит, вместе с **нулём** в байте можно записать **256** различных десятичных чисел.

Если в одном байте представляются и положительные и отрицательные числа, то один бит отводится под знак числа. Знак «+» обычно кодируется **нулём**, а знак «-» – **единицей** в старшем разряде. В этом случае диапазон представляемых чисел будет от **-128** до **+127**.

Целые числа без знака в *двухбайтовом формате* (2 байта= 16 бит) могут принимать значения от **0** до $(2^{16}-1)$, а со знаком – от (-2^{15}) до $(+2^{15}-1)$, т.е. от -32768 до +32767.

Например, десятичное число +19 (10011_2) может быть записано в памяти машины так:



Для представления чисел с плавающей точкой используется **нормализованная форма**:

$$n = m \times q^p,$$

где **m** – **мантисса** числа, **q** – **основание** системы счисления, **p** – **порядок** числа.

$$0,1 \leq |m| < 1.$$

Например: $648.104 = 0.648104 \cdot 10^3$;

При записи числа с плавающей точкой отводятся разряды для хранения мантиссы, порядка, знака числа и знака порядка.

Пусть $n=4,5_{10} = 100,1_2 = 0,1001 \cdot 2^3$.

Мы перевели десятичное число 4,5 в двоичное. В нормализованной форме порядок числа равен трём в десятичной системе, что соответствует числу 11 в двоичной системе.

Итак, мантисса числа равна в двоичной системе 0.1001, порядок равен 11. Дробные числа дополняются незначащими нулями справа, а целые дополняются незначащими нулями слева. В ячейке это будет выглядеть так:

