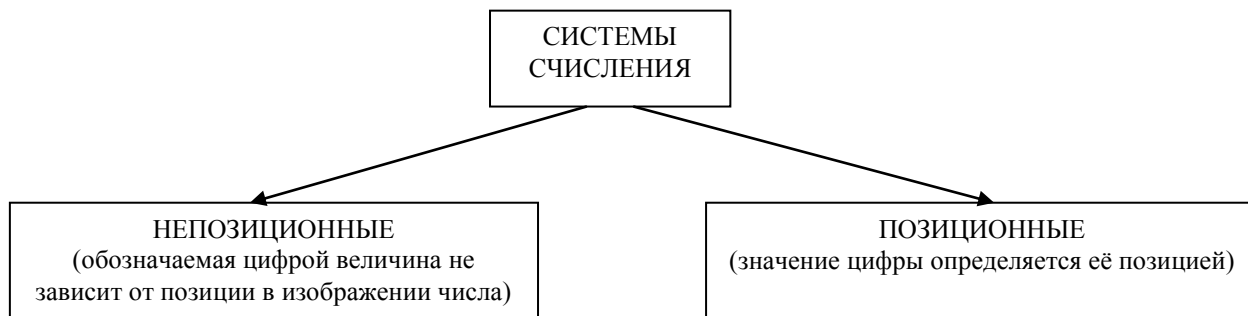


ПРЕДСТАВЛЕНИЕ ДАННЫХ В ЭВМ

Системы счисления числовых данных

Система счисления – совокупность приемов и правил наименования и обозначения чисел, позволяющих установить взаимно однозначное соответствие между любым числом и его представлением в виде конечного числа символов (цифр) из определённого алфавита.



Алфавит позиционной системы счисления – упорядоченный набор символов (цифр), используемый для представления любых чисел в заданной позиционной системе счисления: $\{d^{(0)}, d^{(1)}, \dots, d^{(p-1)}\}$.

Основание позиционной системы счисления – количество различных цифр, используемых для изображения чисел в данной системе счисления: $p > 1$.

Возможно существование бесчисленного множества разнообразных позиционных систем счисления. Системы счисления, наиболее часто используемые при работе с вычислительной техникой, представлены в следующей таблице.

Основание системы счисления	Название системы счисления	Алфавит системы счисления
2	Двоичная	$\{0, 1\}$
3	Троичная	$\{0, 1, 2\}$
8	Восьмеричная	$\{0, 1, 2, 3, 4, 5, 6, 7\}$
10	Десятичная	$\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
16	Шестнадцатеричная	$\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$

Представление натуральных чисел (и нуля) в произвольной p -ичной системе счисления:

$$X_{(n)_p} = X_{n-1} \cdot p^{n-1} + \dots + X_1 \cdot p^1 + X_0 \cdot p^0$$

$$X_i \in \{d^{(0)}, d^{(1)}, \dots, d^{(p-1)}\}$$

n – количество разрядов (разрядность) представления целого числа.

Например:

$$\begin{aligned}
 123_{10} &= 1 \cdot 10^2 + 2 \cdot 10^1 + 3 \cdot 10^0 = \\
 &= 7B_{16} = 7 \cdot 16^1 + 11 \cdot 16^0 = \\
 &= 1111011_2 = 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0
 \end{aligned}$$

Позиции, где располагаются цифры, соответствующие большим степеням порядка p , обычно называются *старшими разрядами*, а соответствующие меньшим степеням порядка p , – *младшими разрядами*.

*) На практике часто заменяют более компактным шестнадцатеричным представлением громоздкую запись в двоичной системе счисления. Существует правило соотношения: каждой шестнадцатеричной цифре однозначно соответствует своё четырёхразрядное двоичное число:

0 ₁₆	0000 ₂	4 ₁₆	0100 ₂	8 ₁₆	1000 ₂	C ₁₆	1100 ₂
1 ₁₆	0001 ₂	5 ₁₆	0101 ₂	9 ₁₆	1001 ₂	D ₁₆	1101 ₂
2 ₁₆	0010 ₂	6 ₁₆	0110 ₂	A ₁₆	1010 ₂	E ₁₆	1110 ₂
3 ₁₆	0011 ₂	7 ₁₆	0111 ₂	B ₁₆	1011 ₂	F ₁₆	1111 ₂

Представление положительных вещественных (действительных) чисел в произвольной p -ичной системе счисления:

$$X_{\langle n \rangle} \cdot X_{\langle m \rangle} p = X_{n-1} \cdot p^{n-1} + \dots + X_1 \cdot p^1 + X_0 \cdot p^0 + X_{-1} \cdot p^{-1} + X_{-2} \cdot p^{-2} + \dots + X_{-m} \cdot p^{-m}$$

$$X_i \in \{d^{(0)}, d^{(1)}, \dots, d^{(p-1)}\}$$

n – разрядность представления целой части числа (количество целых разрядов),

m – разрядность представления дробной части числа (количество дробных разрядов).

Так как при подобной записи позиция десятичной точки, разделяющей целую и дробную части числа, однозначно фиксируется используемой разрядностью представления (n разрядов для целой части числа и m разрядов для дробной части числа), то такое представление вещественных чисел получило название представления **с фиксированной десятичной точкой**.

Например:

$$\begin{aligned} 12.375_{10} &= 1 \cdot 10^1 + 2 \cdot 10^0 + 3 \cdot 10^{-1} + 7 \cdot 10^{-2} + 5 \cdot 10^{-3} = \\ &= 1100.011_2 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} + 1 \cdot 2^{-3} \end{aligned}$$

Однако, в каждой позиционной системе счисления можно точно представить только некоторое подмножество рациональных чисел. Рациональное число, точно представимое в одной из систем счисления, возможно, может быть только приближенно записано в других системах счисления.

Например:

$$\frac{7}{3} = 2.1_3 \approx 10.010101 \dots_2 \approx 2.3333 \dots_2$$

Поэтому при осуществлении арифметических операций с вещественными числами, представленных в одной из позиционных систем счисления, неизбежно возникают *ошибки округления*, наличие которых обязательно нужно учитывать при работе на реальных ЭВМ.

Двоичное представление данных в ЭВМ

1 бит – количество информации, необходимое для определения различий двух равновероятных событий.

В двоичной системе счисления **бит** (*bit, binary digit*) представляет собой один двоичный разряд. Может принимать два значения, обозначаемые обычно как "0" и "1".

Байт (*byte, binary term*) – совокупность битов, одновременно обрабатываемая компьютером. В современных вычислительных системах байт считается равным восьми битам, в этом случае он может принимать одно из 2^8 различных значений (состояний).

Машинное слово – количество битов или байтов, соответствующее разрядности регистров процессора и/или разрядности шины данных ЭВМ. В отличие от бита и байта машинное слово является машиннозависимой и платформозависимой величиной. Для современных 32-битных процессоров x86 исторически машинным словом считается 16 бит, реально – 32 бита (типы данных WORD и DWORD).

Обрабатываемые и хранимые в памяти ЭВМ данные являются аппаратной комбинацией символов 0 и 1. С точки зрения двоичной реализации на ЭВМ имеют место только три основных типа данных:

- **целочисленные данные;**
- **вещественные данные в форме с плавающей десятичной точкой;**
- **данные в форме алфавитно-цифровых символов.**

Правильная интерпретация ячейки памяти каждого типа позволяет корректно извлекать хранимую в ней информацию и обрабатывать ее соответствующим образом.

Двоичное представление целочисленных данных

Беззнаковые целочисленные величины представляются в своем двоичном коде.

Для знаковых целочисленных данных старший бит резервируется под знак (0 – "+", 1 – "-"). В оставшихся $n-1$ битах записывается либо прямой либо добавочный двоичный код числа. Прямой код соответствует обыкновенному двоичному коду. Добавочный код формируется для отрицательных чисел по правилу: инверсия всех битов модуля (абсолютной величины) числа + 1.

Например (для однобайтовых величин):

$$+0_{10} \rightarrow 00000000_2$$

$$-0_{10} \rightarrow 11111111_2 + 1_2 \rightarrow 1 \mid 00000000_2$$

$$+1_{10} \rightarrow 00000001_2$$

$$-1_{10} \rightarrow 11111110_2 + 1_2 \rightarrow 11111111_2$$

$$+2_{10} \rightarrow 00000010_2$$

$$-2_{10} \rightarrow 11111101_2 + 1_2 \rightarrow 11111110_2$$

$+3_{10} \rightarrow 00000011_2$
 $-3_{10} \rightarrow 11111100_2 + 1_2 \rightarrow 11111101_2$

$+50_{10} \rightarrow 00110010_2$
 $-50_{10} \rightarrow 11001101_2 + 1_2 \rightarrow 11001110_2$

$+56_{10} \rightarrow 00111000_2$
 $-56_{10} \rightarrow 11000111_2 + 1_2 \rightarrow 11001000_2$

$+100_{10} \rightarrow 01100100_2$
 $-100_{10} \rightarrow 10011011_2 + 1_2 \rightarrow 10011100_2$

$+127_{10} \rightarrow 01111111_2$
 $-127_{10} \rightarrow 10000000_2 + 1_2 \rightarrow 10000001_2$

$+128_{10} \rightarrow 10000000_2$
 $-128_{10} \rightarrow 01111111_2 + 1_2 \rightarrow 10000000_2$

$+200_{10} \rightarrow 11001000_2$

$+255_{10} \rightarrow 11111111_2$

Примеры арифметических действия без переполнения

Беззнаковые числа:

$+100_{10} \rightarrow 01100100_2$
+
 $+50_{10} \rightarrow 00110010_2$

 $10010110_2 \rightarrow 150_{10}$

Знаковые числа:

$+100_{10} \rightarrow 01100100_2$
+
 $-50_{10} \rightarrow 11001110_2$

 $1|00110010_2 \rightarrow 50_{10}$

Примеры арифметических действия с переполнением

Беззнаковые числа:

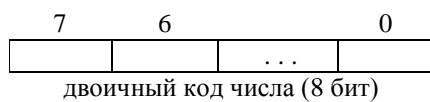
$+100_{10} \rightarrow 01100100_2$
+
 $+200_{10} \rightarrow 11001000_2$

 $1|00101100_2 \rightarrow 44_{10}$

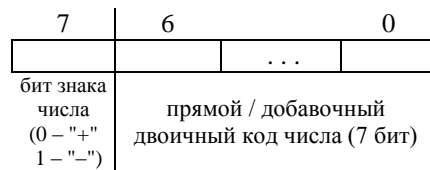
$+100_{10} \rightarrow 01100100_2$
+
 $+50_{10} \rightarrow 00110010_2$

 $10010110_2 \rightarrow -106_{10}$

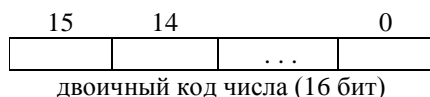
unsigned char :



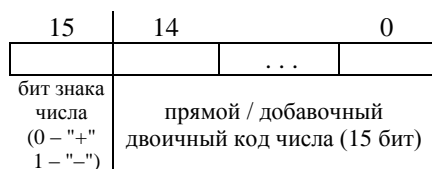
char :



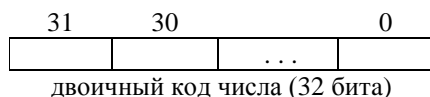
unsigned short :



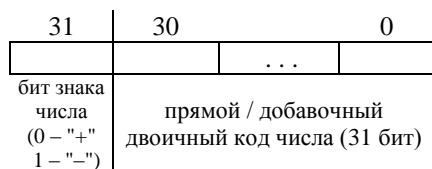
short :



unsigned int , unsigned long :



int , long :



Двоичное представление вещественных данных

Любое вещественное число N , представленное в системе счисления с основанием p , можно записать в виде экспоненциального представления :

$$N = \pm M \times p^{\pm k},$$

где:

$$M = X_{\langle n \rangle} . X_{\langle m \rangle} = X_{n-1} \dots X_1 X_0 . X_{-1} X_{-2} \dots X_{-m} \text{ – мантисса числа,}$$

k – целочисленный порядок числа.

Например:

$$123.45 = 123.45 \times 10^0 = 1234.5 \times 10^{-1} = 12345 \times 10^{-2} = 123450 \times 10^{-3} = \\ = 12.345 \times 10^1 = 1.2345 \times 10^2 = 0.12345 \times 10^3 = 0.012345 \times 10^4$$

$$1011.01 = 1011.01 \times 10^0 = 10110.1 \times 10^{-1} = 101101 \times 10^{-2} = 1011010 \times 10^{-3} = \\ = 101.101 \times 10^1 = 10.1101 \times 10^2 = 1.01101 \times 10^3 = 0.101101 \times 10^4 = 0.0101101 \times 10^5$$

Такое представление вещественных чисел называется *представлением с плавающей (float) десятичной точкой (десятичной запятой)*. Величина порядка определяет, на сколько разрядов необходимо осуществлять сдвиг относительно этой десятичной точки (десятичной запятой).

Нормализованная форма записи – такая форма записи вещественного числа с плавающей десятичной точкой (запятой), при которой эта плавающая десятичная точка (запятая) располагается в мантиссе перед первой значащей цифрой.

$$|M| = 0.X_{\langle m \rangle} = 0.X_{-1}X_{-2} \dots X_{-m}, \quad X_{-1} \neq 0$$

Для двоичной системы счисления при нормализованной форме записи:

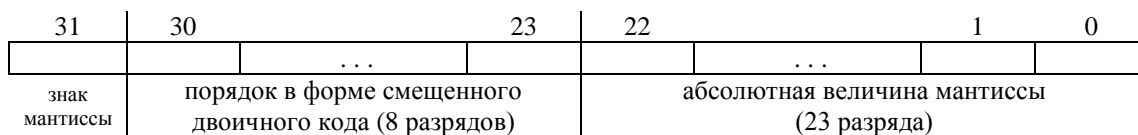
$$|M| = 0.X_{\langle m \rangle} = 0.1X_{-2} \dots X_{-m}$$

и можно хранить только $m-1$ разрядов мантиссы (самый старший разряд всегда равен 1). Значение абсолютной величины мантиссы при этом всегда находится в диапазоне

$$0.5 \leq |M| \leq 1.$$

Порядок нормализованного вещественного числа задается в виде k -разрядного смещенного кода Грея, что позволяет производить операции над порядками как над беззнаковыми числами.

float :



double :

63	62	52	51	1	0
		
знак мантиссы	порядок в форме смещенного двоичного кода (11 разрядов)		абсолютная величина мантиссы (52 разряда)		

long double :

79	78	64	63	1	0
		
знак мантиссы	порядок в форме смещенного двоичного кода (15 разрядов)		абсолютная величина мантиссы (64 разряда)		

Двоичное представление символьной информации

Символьная (алфавитно-цифровая) информация хранится и обрабатывается в ЭВМ в форме цифрового кода, т.е. каждому символу ставится в соответствие отдельное бинарное слово-код.

Среди наборов символов исторически наибольшее распространение получили знаки кода ASCII (*American Standard Code for Information Interchange* – американский стандартный код обмена информацией). Базовый код ASCII – это семиразрядный код, обеспечивающий $2^7 = 128$ различных битовых комбинаций. Базовая таблица ASCII-кодов символов стандартна для всех IBM-совместимых компьютеров. Расширенная таблица ASCII-кодов относится к символам от 128 до 255 и может различаться на системах разного типа. Таким образом, алфавитно-цифровые символы хранятся в памяти ЭВМ в виде восьмиразрядных байтов.

В последнее время все большее распространение получает универсальная система кодирования текстовых данных UNICODE. В данной системе символы кодируются не восьмиразрядными, а 16-разрядными двоичными числами, что позволяет обеспечивать $2^{16} = 65536$ уникальных кодов символов.

Рассмотрим подробнее таблицу кодирования символов ASCII.

Символ	BIN	HEX	DEC
Управляющие	0000 0000	00	0
спец.
символы	0001 1111	1F	31
' ' (SPACEBAR)	0010 0000	20	32
'!'	0010 0001	21	33
.
'/'	0010 1111	2F	47
'0'	0011 0000	30	48
'1'	0011 0001	31	49
.
'9'	0011 1001	39	57
':'	0011 1010	3A	58
.
'@'	0100 0000	40	64
'A'	0100 0001	41	65
'B'	0100 0010	42	66
.
'Z'	0101 1010	5A	90
'['	0101 1011	5B	91
.
'a'	0110 0001	61	97
'b'	0110 0010	62	98
.
'z'	0111 1010	7A	122
'{'	0111 1011	7B	123
.
Символы расшир.	1000 0000	80	128
таблицы
ASCII-кодов	1111 1111	FF	255

В расширенной таблице ASCII-кодов располагаются коды специальных знаков, символов псевдографики и коды национальных символов. Например, для букв русского алфавита широко используется кодировка Windows-1251, а также КОИ-7 и КОИ-8 (коды обмена информацией 7-битный и 8-битный).