Numeral Systems

dr Szymon Murawski

October 29, 2019

1 Numeral systems

Over the years many different numeral systems have been developed, to represent numbers. Almost every human nowadays uses decimal system, that has roots in ten fingers in our hands. Computers however use binary systems, the only systems that is allowed by the physical architecture of computer. During human-computer communication numbers have to be translated from one system to another.

1.1 Positional number systems

In a positional number system with base b, a number is represented by a string of digits d_i, with weight b_i associated with position of a digit. It can be expressed as:

$$(d_{m-1}d_{m-2}\dots d_1 d_0.d_{-1}d_{-2}\dots d_{-n})_b$$

Dot marks the *radix* - a points separating integer and fractional part of a number. In the number above there are m digits to the left of radix (forming integer part), n digits to the right (forming fractional part). The value D of such a digit can be calculated by:

$$D = \sum_{i=-n}^{m-1} d_i b^i \tag{1}$$

For example in base-ten ($b = 10$) system (decimal):

$$542 = 5 * 10^2 + 4 * 10^1 + 2 * 10^0$$

1.2 Standard number systems

In common use are the following number systems:

- **Decimal** - Commonly used by humans
- **Binary** - Used in computers. Reason for using binary system is the ease of implementation - anything that has at least two states can be used to store information, be it low/high voltage, on/off gate, high/low magnetoresistance etc.
- **Octal** - Used as a shorthand for binary systems, commonly used in early computers. Nowadays only used in UNIX file permission system. Numbers written in octal typically start with 0, or 0o (Zero and lowercase o).
- **Hexadecimal** - Commonly used nowadays as a human-readable format of binary numbers. Numbers written in hexadecimal base start with 0x

Table 1: Standard number systems

Name	**Base**	**Digits**	**Examples**
Binary	2	0,1	1001_2
Octal	8	0,1,2,3,4,5,6,7	32_8
Decimal	10	0,1,2,3,4,5,6,7,8,9	49_{10}
Hexadecimal	16	0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F	$4C2_{16}$

Table 2: Numbers expressed in different systems

Binary	**Octal**	**Decimal**	**Hexadecimal**
0	0	0	0
1	1	1	1
10	2	2	2
11	3	3	3
100	4	4	4
101	5	5	5
110	6	6	6
111	7	7	7
1000	10	8	8
1001	11	9	9
1010	12	10	A
1011	13	11	B
1100	14	12	C
1101	15	13	D
1110	16	14	E
1111	17	15	F

2 Conversion between systems

2.1 Any system do decimal

For calculating the value in decimal system of number expressed in any system we can directly use equation 1, as in the following examples:

$$
\begin{aligned}
1A2C_{16} &= 1 \times 16^3 + 10 \times 16^2 + 2 \times 16^1 + 12 \times 16^0 = 6700_{10} \\
428_8 &= 5 \times 8^2 + 2 \times 8^1 + 8 \times *^0 = 280_{10} \\
101101_2 &= 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 45
\end{aligned}
$$

2.2 Decimal to any system

While performing integer division x/y, we obtain the quotient q. By using modulo operation $x\%y$ we obtain the remainder r, hence we can write

$$x = yq + r \tag{2}$$

To convert number N_a expressed in base a to base b we can make use of equation 2 and use the **repeated division algorithm**:

1. Take $n = N$
2. Take the remainder $n\%b$ as the next digit in representation b
3. Set $n = n/b$ (only the integer part!)
4. Go to step 2 if quotient n/b is greater than zero
5. Reverse the order of the representation

For example in we want to convert 2576_{10} into octal base:

$$
\begin{aligned}
2576/8 &= 322 & \quad & remainder\, 0 \rightarrow 0 \\
322/8 &= 40 & & remainder\, 2 \rightarrow 2 \\
40/8 &= 5 & & remainder\, 0 \rightarrow 0 \\
5/8 &= 0 & & remainder\, 5 \rightarrow 5
\end{aligned}
$$

So we obtain

$$2576_{10} = 5020_8$$

Some more examples:

$$
\begin{aligned}
68_{10} &= 104_8 = 44_{16} = 1000100_2 \\
123_{10} &= 173_8 = 7B_{16} = 1111011_2 \\
8271_{10} &= 20117_8 = 204F_{16} = 10000001001111_2
\end{aligned}
$$

2.3 2^k systems

Conversion between 2^k systems (binary, octal, hexadecimal) have an interesting property, that can simplify the process of conversion. Since these number systems possess base 2^k, all numbers within these systems can be uniquely represented by k binary bits. Instead of calculating the value of the number in old or new system, we can then group the bits (according to the base) and use conversion table (like table 2) to write new number. Some examples:

$$\begin{aligned}
11010110_2 &= \overbrace{011}^{3}\,\overbrace{010}^{2}\,\overbrace{110}^{2} = 326_8 \\
&= \overbrace{1101}^{D}\,\overbrace{0110}^{6} = D6_16 \\
11010010_2 &= \overbrace{011}^{3}\,\overbrace{010}^{2}\,\overbrace{010}^{2} = 322_8 \\
&= \overbrace{1101}^{D}\,\overbrace{0010}^{2} = D2_{16} \\
10111011_2 &= \overbrace{010}^{2}\,\overbrace{111}^{7}\,\overbrace{011}^{3} = 273_8 \\
&= \overbrace{1011}^{B}\,\overbrace{1011}^{B} = BB_{16}
\end{aligned}$$

Notice, that we start grouping bits from the right side, and if there are not enough bits for a group, we add zeroes to the left side. This method can also be used to convert octal to hexadecimal number - first write the number in decimal system, then group bits again in the new base.

3 Fractions

If the number has a digit with weight with negative power coefficient, that number is a fraction. Integer and fractional parts of a number are separated by a dot (.) or comma (,), depending on the language used. We will use a single dot as the separator. Fraction can either be finite 0.1426, or infinite ($1/3 = 0.333333(3)$). Based on this we say that a number is either rational or irrational. Be aware, that in different bases different numbers are irrational!

3.1 Converting to decimal

While converting fractional number in any base to decimal, we can once again use equation 1, as in the following examples:

$$\begin{aligned}
0.763_8 &= 7 \times 8^{-1} + 6 \times 8^{-2} + 3 \times 8^{-3} = 0.9746_{10} \\
1A.F1_{16} &= 1 \times 16^1 + 10 \times 16^0 + 15 \times 16^{-1} + 1 \times 16^{-2} = 26.94140625 \\
101.011_2 &= 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} = 5.375_{10}
\end{aligned}$$

3.2 Converting from decimal

To convert fractional number N from decimal system to any base b, we can use **multiplication algorithm**:

1. Set $n = N$
2. Multiply $n \times b$
3. Take integer part of $n \times b$ as the next digit in representation b
4. Set n equal to fractional part of $n \times b$
5. Go to step 2, unless fractional part is equal to zero, or if desired accuracy is met

Second termination condition in the above algorithm is required to deal with irrational numbers

For example, the process of converting 0.372_{10} into binary:

$$\begin{gathered}
0.372_{10} \times 2 = 0.744 \rightarrow 0 \\
0.744_{10} \times 2 = 1.488 \rightarrow 1 \\
0.448_{10} \times 2 = 0.976 \rightarrow 0 \\
0.976_{10} \times 2 = 1.952 \rightarrow 1 \\
0.952_{10} \times 2 = 1.904 \rightarrow 1 \\
0.904_{10} \times 2 = 1.808 \rightarrow 1 \\
\ldots
\end{gathered}$$

In the end we obtain: $0.372_{10} = 0.01011\ldots_2$

Examples:

$$\begin{aligned}
15.231_{10} &= 17.1662132071_8 = F.3B22D0E56_{16} = 1111.00111011_2 \\
521.5_{10} &= 1011.4_8 = 209.8_{16} = 1000001001.1_2 \\
83.05 &= 123.0314631463_8 = 53.0CCCCCCCC_{16} = 1010011.0000110011_2
\end{aligned}$$

3.3 Converting between 2^k systems

Algorithm is the same as in explained in chapter 2.3 - bits are grouped according to target base and translated using character table. Examples:

$$\begin{aligned}
1000.010011_2 &= \overbrace{001}^{1}\,\overbrace{000}^{0}\,.\,\overbrace{010}^{2}\,\overbrace{011}^{3} = 10.23_8 \\
&= \overbrace{1000}^{8}.\overbrace{0100}^{4}\overbrace{1100}^{C} = 8.4C_{16} \\
111010.001001_2 &= \overbrace{111}^{7}\,\overbrace{010}^{2}\,.\,\overbrace{001}^{1}\,\overbrace{001}^{1} = 72.11_8 \\
&= \overbrace{0011}^{3}\overbrace{1010}^{A}.\overbrace{0010}^{2}\overbrace{0100}^{4} = 3A.24_{16}
\end{aligned}$$