Conversão entre bases numéricas*

Prof. Daniel Corrêa LOBATO

Revisão: novembro, 2004

Resumo

Toda a informação armazenada em um dispositivo computacional deve estar representada em base binária. Isso significa que o computador é incapaz de efetuar $10_{(10)}+5_{(10)}$. Entretanto, se os mesmos dados estiverem armazenados em base binária, a operação $1010_{(2)} + 101_{(2)}$ se torna trivial, e resposta $1111_{(2)}$ flui como mágica. Neste módulo vamos discutir conversão entre as bases numéricas 2, 8, 10 e 16 e, no módulo seguinte, as operações aritméticas elementares nessas bases.

1 Objetivos

- Apresentar as bases numéricas 2, 8, 10 e 16
- Apresentar os algoritmos de conversão entre as bases

2 Representação da informação

Os números decimais, que estamos habituados a usar, usa um sistema de numeração posicional; ou seja, a posição ocupada por cada algarismo em um número altera seu valor em potência de 10 (na base 10) para cada casa à esquerda. Assim, o número 423 na base 10 pode ser decomposto na forma representada na tabela 1.

	centena 10^2	dezena 10^1	unidade 10^0	
algarismo	4	2	3	
valor	4×10^2	2×10^1	3×10^0	$400 + 20 + 3 = 423$

Tabela 1: Representação do número 423 em base 10

A *base* de um sistema de numeração é a quantidade de algarismos que podem ser usados na representação dos números. A base 10, usada por nós humanos, tem dez algarismos, de 0 a 9. Entretanto, ela não é a única: nós compramos um cento de laranjas (base 100), marcamos o tempo usando minutos e segundos (base 60), e compramos uma dúzia de ovos (base 12).

Os computadores usam base 2 (binário), e os programadores, por conveniência, usam uma base potência de 2, como 2^4 (base 16) ou base 2^3 (base 8).

Uma base b qualquer disporá de b algarismos, variando entre 0 e $(b-1)$.

*Material desenvolvido para o segundo bimestre da disciplina *Computação e Automação de Sistemas* do curso de Ciência da Computação das Faculdades COC

Intuitivamente, sabemos que o maior número que podemos representar, com n algarismos, na base b, será o número composto n vezes pelo maior algarismo disponível naquela base (ou seja, $b-1$). Por exemplo, o maior número que pode ser representado na base 10 usando 3 algarismos será 999 (ou seja, $10^3 - 1 = 999$).

Generalizando, podemos ver que o maior número inteiro N que pode ser representado, em uma dada base b, com n algarismos (n "casas"), será $N = b^n - 1$. Assim, o maior número de 2 algarismos na base 16 será $FF_{(16)}$ que, na base 10, equivale a $255_{(10)} = 16^2 - 1$.

2.1 Representação binária

Os computadores modernos usam, hoje, apenas o sistema binário (veja tabela 2. Isto significa que tudo o que é armazenado em um computador o é usando, apenas, duas grandezas: 0 e 1. Havendo apenas dois algarismos, portanto dígitos binários, o elemento mínimo de informação nos computadores foi apelidado de *bit* (uma contração do inglês ***bi****nary digi***t**).

Representação binária	Potência	Representação decimal
1	2^0	1
10	2^1	2
100	2^2	4
1000	2^3	8
10000	2^4	16
100000	2^5	32
1000000	2^6	64
10000000	2^7	128
100000000	2^8	256
1000000000	2^9	512
10000000000	2^{10}	1024

Tabela 2: Potências de dois

Um número representado em binário apresenta muitos bits. Com muitos bits, a chance de o programador ou usuário digitar um incorreto é maior. Para facilitar a visualização e manipulação das grandezas processadas em computadores são usualmente adotadas as representações octal (base 8) e principalmente hexadecimal (base 16). Essa representação é apenas para nós, humanos mortais: os computadores trabalham muito bem, obrigado, na base 2.

2.2 Representação em octal e hexadecimal

Nós, humanos, tendemos a representar as quantidades usando sistemas de numeração potência de dois, com o objetivo de reduzir o número de algarismos do número que desejamos manipular. As duas bases mais usadas são a octal (base 8) e a hexadecimal (base 16). No sistema octal (base 8), cada três bits são representados por apenas um algarismo octal (de 0 a 7). No sistema hexadecimal (base 16), cada quatro bits são representados por apenas um algarismo hexadecimal (de 0 a F). Veja a relação entre os valores nas diferentes bases na tabela 3.

Base 2	Base 8	Base 10	Base 16
0000	0000	0000	0000
0001	0001	0001	0001
0010	0002	0002	0002
0011	0003	0003	0003
0100	0004	0004	0004
0101	0005	0005	0005
0110	0006	0006	0006
0111	0007	0007	0007
1000	0010	0008	0008
1001	0011	0009	0009
1010	0012	0010	000A
1011	0013	0011	000B
1100	0014	0012	000C
1101	0015	0013	000D
1110	0016	0014	000E
1111	0017	0015	000F

Tabela 3: Correspondência entre as bases 2, 8, 10 e 16

3 Conversão entre bases

Estando diante do número $101101011_{(2)}$, como saber qual o correspondente em base 10 ou base 16[1]?

Quando a conversão é entre bases que são potência entre si, fica fácil

3.1 Conversão entre bases 2, 8 e 16

Vamos exemplificar com a conversão entre a base 2 e a base 8. Como $2^3 = 8$, se separarmos os bits de um número binário em grupos de três bits (começando sempre da direita para a esquerda!) e convertermos cada grupo de três bits para seu equivalente em octal, teremos a representação do número em octal. Por exemplo:

$$\begin{aligned} 10111100011_{(2)} \\ 010\ 111\ 100\ 011_{(2)} \\ 2\ 7\ 4\ 3_{(8)} \end{aligned}$$

Assim, $10111100011_{(2)}$ é o mesmo que $2743_{(8)}$. Para converter para a base 16 ($2^4 = 16$), o processo é o mesmo, só que agora os bits são agrupados de quatro em quatro. Veja:

$$\begin{aligned} 10111100011_{(2)} \\ 0101\ 1110\ 0011_{(2)} \\ 5\ E\ 3_{(16)} \end{aligned}$$

E para converter entre as bases 16 e 8? O mais fácil é converter da base 16 para a base 2 (um algarismo por vez) e, depois, agrupar de três em três bits e, por fim, converter para os algarismos octais. Veja o exemplo:

[1]Só por curiosidade: $363_{(10)}$ e $16B_{(16)}$

$$\begin{array}{r}
5\ E\ 3_{(16)} \\
0101\ 1110\ 0011_{(2)} \\
10111100011_{(2)} \\
010\ 111\ 100\ 011_{(2)} \\
2\ 7\ 4\ 3_{(8)}
\end{array}$$

3.2 Conversão entre base b e base 10

Um número, em base b, qualquer b, pode ser representado na forma $N = a_n \times b^n + \ldots + a_2 \times b^2 + a_1 \times b^1 + a_0 \times b^0$. Com base nessa informação você pode converter números em uma base b qualquer para números em base 10. Basta seguir o seguinte exemplo:

$$\begin{aligned}
101101_{(2)} &= 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\
&= 32 + 0 + 8 + 4 + 0 + 1 \\
&= 45_{(10)}
\end{aligned}$$

E isso vale para qualquer base mesmo. Veja:

$$\begin{aligned}
4F5_{(16)} &= 4 \times 16^2 + 15 \times 16^1 + 5 \times 16^0 \\
&= 4 \times 256 + 15 \times 16 + 5 \\
&= 1024 + 240 + 5 \\
&= 1269_{(10)}
\end{aligned}$$

3.3 Conversão entre base 10 e base b

O número decimal será dividido sucessivas vezes pela base b; o resto de cada divisão ocupará sucessivamente as posições de ordem 0, 1, 2 e assim por diante até que o resto da última divisão (que resulta em qüociente zero) ocupe a posição de mais alta ordem. Se você, eventualmente, continuar a dividir o qüociente zero pela base b, vai apenas obter zeros de ordem mais alta, o que não mudará o número resultante na base b, pois serão zeros à esquerda. Veja o exemplo da conversão do número $19_{(10)}$ para a base 2.

$$\begin{array}{rrrrrrrrr}
 & 19 & \vert\; 2 & & & & & & \\
a_0 = & \mathbf{1} & \vert\; 9 & \vert\; 2 & & & & & \\
 & & a_1 = \mathbf{1} & \vert\; 4 & \vert\; 2 & & & & \\
 & & & a_2 = \mathbf{0} & \vert\; 2 & \vert\; 2 & & & \\
 & & & & a_3 = \mathbf{0} & \vert\; 1 & \vert\; 2 & & \\
 & & & & & a_4 = \mathbf{1} & \vert\; 0 & \vert\; 2 & \\
 & & & & & & a_5 = 0 & \vert\; 0 & \ldots
\end{array}$$

Assim, $19_{(10)} = 10011_{(2)}$. De fato, pois

$$\begin{aligned}
10011_{(2)} &= 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \\
&= 16 + 0 + 0 + 2 + 1 \\
&= 19_{(10)}
\end{aligned}$$

4 Exercícios

Efetue as seguintes conversões de base, da base indicada para as demais (2, 8, 10 e 16)

- $11010101_{(2)}$
- $638_{(10)}$
- $77110_{(8)}$
- $BACABA_{(16)}$
- $CA1C0_{(16)}$
- $100111010110_{(2)}$
- $615243_{(8)}$
- $11010101_{(10)}$