

Aula 1 – Sistemas de numeração

TECNOLOGIA em **Gestão de Dados**

Programação I

Prof. Dr. Wener Sampaio



ReUni
DIGITAL

CEAD
Centro de Educação
Aberta e a Distância

UFPI
UNIVERSIDADE
FEDERAL DO PIAUÍ

MEC

Sistemas de numeração

- É um sistema em que um conjunto de números são representados por numerais de uma forma consistente.
- Que número é esse 11?
- Alfabeto
 - {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
 - Tamanho = 10

Sistemas de numeração

- Sistema não posicional
 - O valor de um símbolo não se altera com a posição do mesmo na palavra numérica.
 - Ex: Sistema de numeração romano
 - Alfabeto = {I, V, X, L, C, D, M}
 - XXXI

Sistemas de numeração

- Sistema posicional

- O valor de um símbolo se altera com a posição do mesmo na palavra numérica.
- Símbolos mais à esquerda possuem maior valor
- Ex.: Sistema de numeração decimal
 - 4
 - 45
 - 431

Bases numéricas

- Conversão para decimal

$$d = \sum_{i=0}^{n-1} (x_i \times B^i)$$

$$d = x_{n-1} \times B^{n-1} + x_{n-2} \times B^{n-2} + \dots + x_1 \times B^1 + x_0 \times B^0$$

- d: é o decimal equivalente
- B: base de origem
- n: quantidade de dígitos
- x_i : dígito na posição i

Bases numéricas

- Exemplo: 432_{10}

- 4 centenas
- 3 dezenas
- 2 unidades

$$\begin{aligned} 432 &= 4 \times 10^2 + 3 \times 10^1 + 2 \times 10^0 \\ &= 4 \times 100 + 3 \times 10 + 2 \times 1 \\ &= 400 + 30 + 2 \\ &= 432 \end{aligned}$$

Base 2 - binários

- Base utilizada em sistemas digitais.
- Circuitos e dispositivos de um computador precisam de um modelo padrão de comunicação.
- Hardware: dois estados
 - Ausência ou presença de corrente;
 - V ou F;
 - Sim ou Não.
- Todas as informações de um computador são formados exclusivamente por 0 ou 1 (bits).

Base 2 - binários

- Quantidade de valores representados: 2^n

p0	Valor
0	0
1	1

p1	p0	Valor
0	0	00
0	1	01
1	0	10
1	1	11

p2	p1	p0	Valor
0	0	0	000
0	0	1	001
0	1	0	010
0	1	1	011
1	0	0	100
1	0	1	101
1	1	0	110
1	1	1	111

- n é a quantidade de bits da palavra binária.

Base 8 - octais

- Derivou do sistema binário;
- Foi utilizado para compactar números binários em linguagem de máquina;
- Alfabeto: {0, 1, 2, 3, 4, 5, 6, 7};
- 1 octal $\square \square$ 3 bits.

Base 16 - hexadecimais

- Derivou do sistema binário;
- Utilizado para compactar binários.
- Fácil visualização;
- Alfabeto: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F};
- 1 hexadecimal $\square\square$ 4 bits.

Conversões entre sistemas

- De qualquer base para decimal:

$$d = \sum_{i=0}^{n-1} (x_i \times B^i)$$

- 49152_{10}
- $1100\ 0000\ 0000\ 0000_2$
- 140000_8
- $C000_{16}$

Conversões entre sistemas

- De decimal para qualquer outra base: (divisões sucessivas)
 1. Dividir o **DECIMAL** pela base desejada;
 2. Registrar quociente e resto;
 3. Dividir o quociente pela base;
 4. Ir ao passo 2, até obter quociente igual a 0 (zero);
 5. Formar o número na base desejada usando os restos na ordem inversa em que foram obtidos.

Conversões entre sistemas

- De decimal para binário:

Dividendo /2	Quociente	Resto
56	28	0
28	14	0
14	7	0
7	3	1
3	1	1
1	0	1

• $56_{10} = 111000_2$

Dividendo /2	Quociente	Resto
197	98	1
98	49	0
49	24	1
24	12	0
12	6	0
6	3	0
3	1	1
1	0	1

$197_{10} = 11000101_2$

Conversões entre sistemas

- De decimal para binário:
 - Método da decomposição em somatório de potências de 2
 - Indicado para pequenos valores e próximos de potências de 2.
 - Exemplo: 523

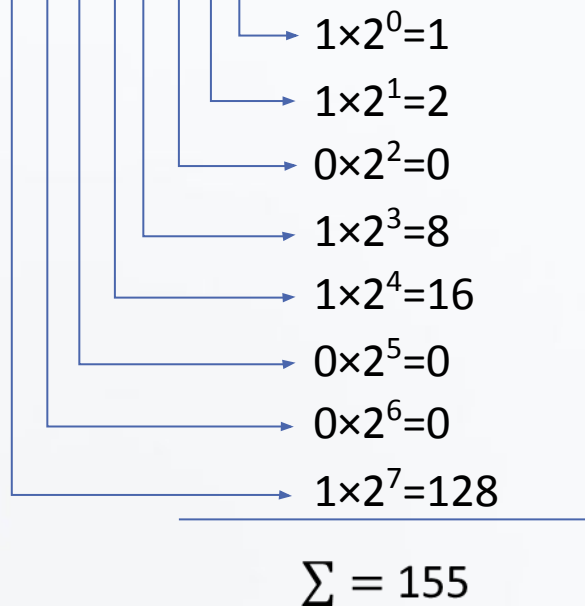
$$523 = 512 + 11 = 512 + 8 + 3 = 512 + 8 + 2 + 1 = 2^9 + 2^3 + 2^1 + 2^0$$

1	0	0	0	0	0	1	0	1	1
9	8	7	6	5	4	3	2	1	0

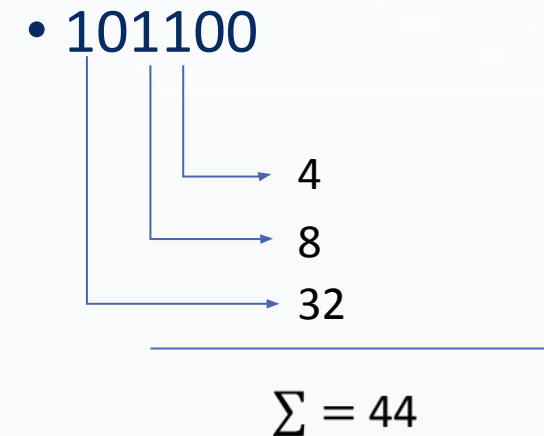
Conversões entre sistemas

- De binário para decimal:

- Forma geral: $d = \sum_{i=0}^{n-1} (x_i \times B^i)$
- 10011011



- Forma prática: substituir apenas os 1s por potências de 2.



Conversões entre sistemas

- De decimal para octal:

Dividendo /8	Quociente	Resto
217	27	1
27	3	3
3	0	3

- $217_{10} == 331_8$

- De octal para decimal: $d = \sum_{i=0}^{n-1} (x_i \times B^i)$
- $71263_8 =$

$$3 \times 8^0 = 3$$

$$6 \times 8^1 = 48$$

$$2 \times 8^2 = 128$$

$$1 \times 8^3 = 512$$

$$7 \times 2^4 = 28672$$

$$\Sigma = 29363_{10}$$

Conversões entre sistemas

- De binário para octal:

- Octal \square base 8 $\square 8 = 2^3$
- Cada grupo de 3 bits formam um octal.
- $11001_2 \square 011\ 001 \square 31_8$

- De octal para binário:

- Cada octal forma grupo de 3 bits.
- $136_8 \square 001\ 011\ 110 \square 1011110_2$

Conversões entre sistemas

- De decimal para hexadecimal:

Dividendo /16	Quociente	Resto
2736	171	0
171	10	11 = B
10	0	10 = A

- $2736_{10} = \text{AB0}_{16}$

- De hexadecimal para decimal: $d = \sum_{i=0}^{n-1} (x_i \times B^i)$
 - C01F_{16}

$$15 \times 16^0 = 15$$

$$1 \times 16^1 = 16$$

$$0 \times 16^2 = 0$$

$$12 \times 16^3 = 49152$$

$$\Sigma = 49183$$

- $\text{C01F}_{16} = 49183_{10}$

Conversões entre sistemas

- De binário para hexadecimal:

- Hexadecimal \square base 16 \square
 $16 = 2^4$

- Cada grupo de 4 bits formam um hexadecimal.

- 10111001111100111_2
1 0111 0011 1110 0111
1 7 3 E 7
 $10111001111100111_2 = 173E7_{16}$

- De hexadecimal para binário:

- Cada hexadecimal gera 4 bits.

- $A56B_{16}$
1010 0101 0110 1011
 $A56B_{16} = 1010 0101 0110 1011_2$

Conversões entre sistemas

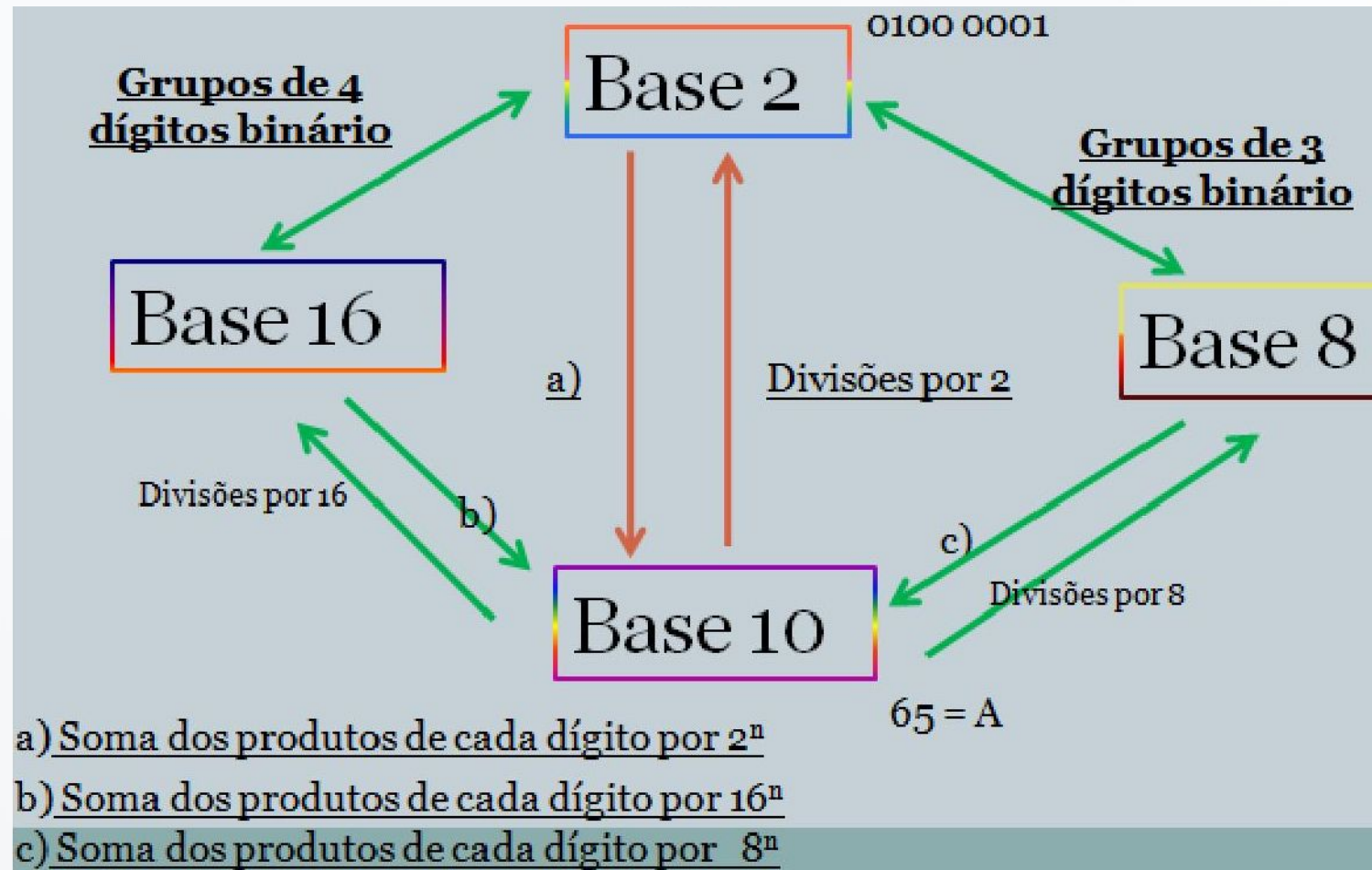
- Entre octal e hexadecimal:
 - Não existe conversão direta.
 - Deve-se utilizar uma base intermediária.
 - Modo fácil: usar base 2
 - Modo difícil: usar base 10 (ou outra base)

- Exemplos:

- $C0F_{16} \square B_8$
 - $C0F_{16} \square B_2 \square B_8$
 - $C0F_{16} \square 1100\ 0000\ 1111_2$
 - $1100\ 0000\ 1111_2 \square B_8$
 - $110\ 000\ 001\ 111_2 \square 6017_8$
- $741_8 \square B_{16}$
 - $741_8 \square B_2 \square B_{16}$
 - $741_8 \square 111\ 100\ 001_2$
 - $111\ 100\ 001_2 \square B_{16}$
 - $1\ 1110\ 0001_2 \square 1E1_{16}$

Conversões entre sistemas

- Resumo:



Exercícios:

- Converter:

- $53_8 \square B_2$
- $2B_{16} \square B_2$
- $1100101_2 \square B_{10}$
- $CAD_{16} \square B_8$
- $1030_{10} \square B_5$
- $103_4 \square B_6$

Vídeo

