Informática Aplicada à Química

Sistemas de Numeração

Representação de Dados

Representando Dados

- Bit
- Byte
- Palavra

Bit (b)

- Abreviação de binary digit (dígito binário).
  - Dois valores possíveis: 0 e 1.
  - Nunca pode estar vazio.
- Unidade básica para armazenar dados:
  - 0 significa desligado; 1 significa ligado.

Byte (B)

- Um grupo de 8 bits.
  - Cada byte tem 256 (2^8) valores possíveis.
- Para texto, armazena um caractere:
  - Pode ser letra, dígito ou caractere especial.
- Dispositivos de memória e armazenamento são medidos em número de bytes.

Palavra (word)

- O número de bits que a CPU processa como uma unidade.
  - Tipicamente, um número inteiro de bytes.
  - Quanto maior a palavra, mais potente é o computador.
  - Computadores pessoais tipicamente têm 32 ou 64 bits de extensão de palavras.

Representação de Dados

- Os computadores entendem duas coisas: ligado e desligado.
- Dados são representados na forma binária:
  - Sistema numérico binário (base 2).
  - Contém somente 2 dígitos: 0 e 1.
    - Corresponde a dois estados: ligado e desligado.

EQUIVALENTES BINÁRIOS DOS NÚMEROS DECIMAIS DE 0 A 15

Decimal	Binário
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

Representação de Dados

- **Hexadecimal (base *16*)**
  - **Algarismos: números *0*, *1*, *2*, *3*, *4*, *5*, *6*, *7*, *8* e *9* letras *A*, *B*, *C*, *D*, *E* e *F***
  - **Empregado na representação de números grandes, e.g. endereços de memória**

Decimal	Hexadecimal	Binário
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

Potências de 2, 10 e 16

n	2^n	10^n	16^n
0	1	1	1
1	2	10	16
2	4	100	256
3	8	1.000	4.096
4	16	10.000	65.536
5	32	100.000	1.048.576
6	64	1.000.000	16.777.216
7	128	10.000.000	268.435.456
8	256	100.000.000	4.294.967.296
9	512	1.000.000.000	68.719.476.736
10	1.024	10.000.000.000	1.099.511.627.776
11	2.048	100.000.000.000	17.592.186.044.416
12	4.096	1.000.000.000.000	281.474.976.710.656
13	8.192	10.000.000.000.000	4.503.599.627.370.500
14	16.384	100.000.000.000.000	72.057.594.037.927.900
15	32.768	1.000.000.000.000.000	1.152.921.504.606.850.000

Revisão de Bases Numéricas

Base Decimal

• Na base decimal são usados 10 dígitos (ou algarismos) diferentes {0,1,2,3,4,5,6,7,8,9}, formando números em base 10.

• Cada algarismo N de um número possui um valor que depende de sua posição. (N * Bpos)

Exemplo:

1999

• O dígito mais a direita (9) representa a quantidade de unidades, pois está na posição 0. O dígito 9 mais a esquerda ,pode ser interpretado como sendo 9 * 100

O valor completo do número pode ser calculado como sendo

$$1*10^3 + 9*10^2 + 9*10^1 + 9*10^0 =$$

$$1*1000 + 9*100 + 9*10 + 9*1 =$$

$$1000 + 900 + 90 + 9 =$$

$$1999$$

Revisão de Bases Numéricas

Base Hexadecimal

- Na base hexadecimal dispomos de 16 algarismos {0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F}, onde os dígitos de 0 a 9 são idênticos as decimais, e os dígitos de A a F, correspondem aos valores decimais de 10 a 15, sucessivamente.

Exemplo:

$$10AC_{16}$$

O valor completo do número pode ser convertido a base 10

$$1*16^3 + 0*16^2 + 10*16^1 + 12*16^0$$

$$1*4096 + 0*256 + 10*16 + 12*1$$

$$4096 + 0 + 160 + 12$$

$$4268_{10}$$

Revisão de Bases Numéricas

Base Binária

• A base binária ou base 2 é a maneira usual de representação de números em computadores eletrônicos.

• Nesta forma de representação temos apenas dois algarismos disponíveis {0,1} que correspondem aos sinais elétricos ligado e desligado.

Exemplo:

$$10011_2$$

O valor completo do número pode ser convertido a base 10

$$1*2^4 + 0*2^3 + 0*2^2 + 1*2^1 + 1*2^0$$

$$1*16 + 0*8 + 0*4 +1*2 +1*1$$

$$16+0+0+2+1$$

$$19_{10}$$

Conversão de Bases Numéricas

- Para a conversão de um número inteiro em uma base qualquer para a base decimal usa-se a seguinte fórmula:

$$N=a_{n-1} \text{ x } b^{n-1} + a_{n-2} \text{ x } b^{n-2} + \ldots + a_1 \text{ x } b^1 + a_0 \text{ x } b^0$$

Onde:

- a : Indica cada algarismo do número
- b : Indica a base de numeração
- n : Indica a posição do algarismo, contando a partir da direita

Conversão de Bases Numéricas

- Para a conversão de um número real em uma base qualquer para a base decimal usa-se a seguinte fórmula:

$$N=a_i \times b^i + \ldots + a_2 \times b^2 + a_1 \times b^1 + a_0 \times b^0 + a_{-1} \times b^{-1} + a_{-2} \times b^{-2} + \ldots + a_{-i} \times b^{-i}$$

(parte inteira, antes da vírgula) (parte fracionária, depois da vírgula)

Onde:

- a : Indica cada algarismo do número
- b : Indica a base de numeração
- i : Indica a posição do algarismo

Conversão de Bases Numéricas

Exemplos:

$$1011_2 \text{ (na base 2)}$$

$$1x2^3 + 0x2^2 + 1x2^1 + 1x2^0 = 11_{10}$$

$$1001{,}01_2$$

$$1x2^3 + 0x2^2 + 0x2^1 + 1x2^0 + 0x2^{-1} + 1x2^{-2} = 8 + 0 + 0 + 1 + 0 + 0{,}25 = 9{,}25_{10}$$

Conversão de Bases Numéricas

Ao converter proceda assim:

$$1011_2 = 1x2^0 + 1x2^1 + 0x2^2 + 1x2^3 = 11_{10}$$

$$1001_2 = 1x2^0 + 0x2^1 + 0x2^2 + 1x2^3 = 9_{10}$$

Conversão de Bases Numéricas

Método mais rápido para conversão de binário para decimal:

128	64	32	16	8	4	2	1		*na base 10*
0	0	0	0	1	0	1	0	=	8 + 2 = 10
0	0	0	1	1	0	0	0	=	16 + 8 = 24
1	1	0	0	0	0	0	0	=	128 + 64 = 192
1	0	1	1	1	0	0	1	=	128 + 32 + 16 + 8 + 1 = 185

Conversão de Bases Numéricas

Converter os seguintes números da base 2 para a base 10:

Conversão de Bases Numéricas

Converter os seguintes números da base 2 para a base 10:

1110 =

Conversão de Bases Numéricas

Converter os seguintes números da base 2 para a base 10:

1110 = 14

Conversão de Bases Numéricas

Converter os seguintes números da base 2 para a base 10:

1110	**=**	**14**
1001	**=**	

Conversão de Bases Numéricas

Converter os seguintes números da base 2 para a base 10:

1110	**=**	**14**
1001	**=**	**9**

Conversão de Bases Numéricas

Converter os seguintes números da base 2 para a base 10:

1110	=	14
1001	=	9
11010	=	

Conversão de Bases Numéricas

Converter os seguintes números da base 2 para a base 10:

1110	=	14
1001	=	9
11010	=	26

Conversão de Bases Numéricas

Converter os seguintes números da base 2 para a base 10:

1110	**=**	**14**
1001	**=**	**9**
11010	**=**	**26**
101001	**=**	

Conversão de Bases Numéricas

Converter os seguintes números da base 2 para a base 10:

1110	=	14
1001	=	9
11010	=	26
101001	=	41

Conversão de Base 10 para qualquer base

- **Este procedimento consiste em dividir o número representado na base 10 sucessivamente pela nova base em que se deseja representá-lo, até que o quociente da divisão seja menor que a base em questão.**

- **Em seguida toma-se o último quociente e os restos das sucessivas divisões em ordem inversa e obtém-se, assim, a representação do número na nova base.**

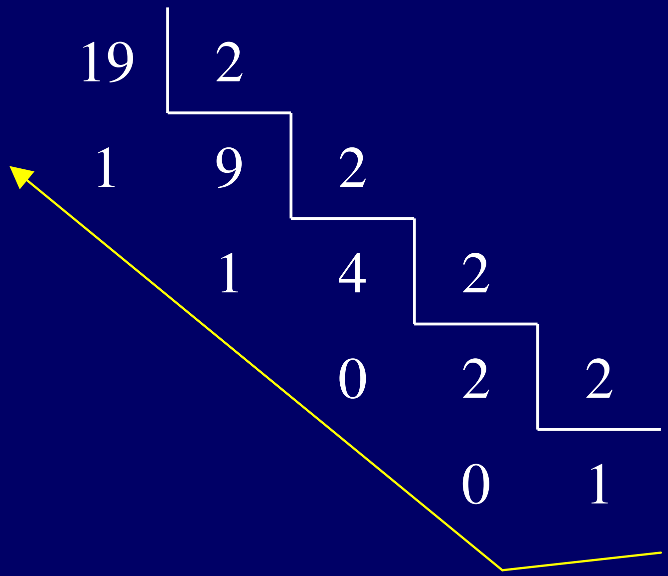
Exemplo:

Converter 26_{10} para base 2:

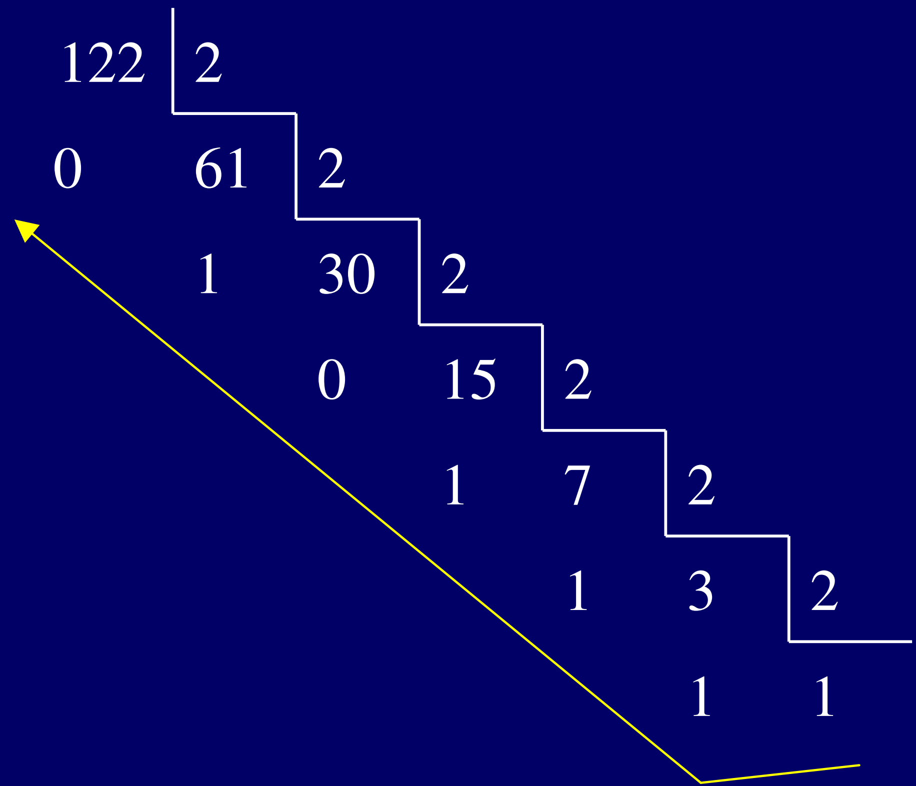
26	2			
0	13	2		
	1	6	2	
		0	3	2
			1	1

Portanto $26_{10} = 11010_2$

Conversão de Base 10 para qualquer base



$$19_{10} = 10011_2$$



$$122_{10} = 1111010_2$$

Conversão de Bases Numéricas

Converter os seguintes números da base 10 para a base 2:

Conversão de Bases Numéricas

Converter os seguintes números da base 10 para a base 2:

14 =

Conversão de Bases Numéricas

Converter os seguintes números da base 10 para a base 2:

14 = 1110

Conversão de Bases Numéricas

Converter os seguintes números da base 10 para a base 2:

14 = 1110

9 =

Conversão de Bases Numéricas

Converter os seguintes números da base 10 para a base 2:

14 = 1110

9 = 1001

Conversão de Bases Numéricas

Converter os seguintes números da base 10 para a base 2:

14	=	1110
9	=	1001
26	=	

Conversão de Bases Numéricas

Converter os seguintes números da base 10 para a base 2:

14	=	1110
9	=	1001
26	=	11010

Conversão de Bases Numéricas

Converter os seguintes números da base 10 para a base 2:

14	=	1110
9	=	1001
26	=	11010
41	=	

Conversão de Bases Numéricas

Converter os seguintes números da base 10 para a base 2:

14	=	1110
9	=	1001
26	=	11010
41	=	101001

Conversão de Bases Numéricas

Converter os seguintes números da base 10 para a base 2:

14	=	1110
9	=	1001
26	=	11010
41	=	101001
255	=	

Conversão de Bases Numéricas

Converter os seguintes números da base 10 para a base 2:

14	=	1110
9	=	1001
26	=	11010
41	=	101001
255	=	11111111

Conversão de Bases Numéricas

Converter os seguintes números da base 10 para a base 2:

14	=	1110
9	=	1001
26	=	11010
41	=	101001
255	=	11111111
72	=	

Conversão de Bases Numéricas

Converter os seguintes números da base 10 para a base 2:

14	=	1110
9	=	1001
26	=	11010
41	=	101001
255	=	11111111
72	=	1001000

Conversão de Base 10 para qualquer base

Se o número for fracionário, a conversão se fará em duas etapas distintas: primeiro a parte inteira e depois a parte fracionária. O algoritmo para a parte fracionária consiste de uma série de multiplicações sucessivas do número fracionário a ser convertido pela base; a parte inteira do resultado da primeira multiplicação será o valor da primeira casa fracionária e a parte fracionária será de novo multiplicada pela base; e assim por diante, até o resultado dar zero ou até encontrarmos o número de casas decimais desejado.

exemplo: 19,125

19	2			
1	9	2		
	1	4	2	
		0	2	2
			0	1

parte inteira $19_{10} = 10011_2$

parte fracionária:

$0{,}125 \times 2 = 0{,}25$

$0{,}25 \times 2 = 0{,}50$

$0{,}50 \times 2 = 1{,}00$ ← a fração zerou!

ordenando os valores inteiros: $,001_2$

Resposta: $10011{,}001_2$

Revisão de Bases Numéricas

Adição binária

$$\begin{array}{r} 11001 \\ +\ \ 10011 \\ \hline 101100 \end{array} \rightarrow \begin{array}{r} 25 \\ +\ \ 19 \\ \hline 44 \end{array}$$

$$\begin{array}{r} 1100 \\ +\ \ 111 \\ \hline 10011 \end{array} \rightarrow \begin{array}{r} 12 \\ +\ \ 7 \\ \hline 19 \end{array}$$

Revisão de Bases Numéricas

Subtração binária

$$\begin{array}{r} 11101 \\ -\quad 111 \\ \hline 10110 \end{array} \rightarrow \begin{array}{r} 29 \\ -\quad 7 \\ \hline 22 \end{array}$$

Capacidades de Armazenamento

- Kilobyte: 1024 (2^{10}) bytes.
  - Capacidade de memória dos computadores pessoais mais antigos.
- Megabyte: aproximadamente, um milhão (2^{20}) de bytes.
  - Memória de computadores pessoais.
  - Dispositivos de armazenamento portáteis (disquetes, CD-ROMs).
- Gigabyte: aproximadamente, um bilhão (2^{30}) de bytes.
  - Dispositivos de armazenamento (discos rígidos).
  - Memória de mainframes e servidores de rede.
- Terabyte: aproximadamente, um trilhão (2^{40}) de bytes.
  - Dispositivos de armazenamento para sistemas muito grandes.

Capacidades de Armazenamento

Prefixos em uso na computação coloquial

Nome	Abrev	Fator	tam SI
quilo	K	2^{10} ="1024"	10^{3} ="1000"
mega	M	2^{20} ="1 048 576"	10^{6} ="1 000 000"
giga	G	2^{30} ="1 073 741 824"	10^{9} ="1 000 000 000"
tera	T	2^{40} ="1 099 511 627 776"	10^{12} ="1 000 000 000 000"
peta	P	2^{50} ="1 125 899 906 842 624"	10^{15} ="1 000 000 000 000 000"
exa	E	2^{60} ="1 152 921 504 606 846 976"	10^{18} ="1 000 000 000 000 000 000"
zetta	Z	2^{70} ="1 180 591 620 717 411 303 424"	10^{21} ="1 000 000 000 000 000 000 000"
yotta	Y	2^{80} ="1 208 925 819 614 629 174 706 176"	10^{24} ="1 000 000 000 000 000 000 000 000"

Estes são idênticos aos prefixos SI, exceto pelo "K", que corresponde ao "k" no SI (K representa Kelvin no SI).

Prefixos IEC padrão

Em 2000, a *International Electrotechnical Commission* (IEC) publicou a segunda edição da *Amendment 2 to "IEC 60027-2*: Letras e símbolos a serem usados na tecnologia elétrica - Parte 2: Telecomunicações e eletrônica ". Este padrão, que tinha sido aprovado em 1998, introduziu os prefixos *kibi-*, *mebi-*, *gibi-*, *tebi-*, *pebi-*, *exbi-*, *zebi-* e *yobi-* para serem usados para especificar múltiplos binários de uma quantidade. Os nomes vêm das versões simplificadas dos prefixos originais do SI; *bi* é a simplificação de "binário". Esclarece ainda que, do ponto da vista do IEC, os prefixos do SI têm somente seu significado na base-10 e nunca têm um significado na base-2.

Novo IEC padrão de prefixos

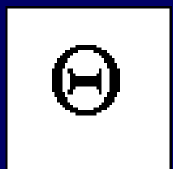
Nome	Abrev	Fator
kibi	Ki	2^{10} ="1024"
mebi	Mi	2^{20} ="1 048 576"
gibi	Gi	2^{30} ="1 073 741 824"
tebi	Ti	2^{40} ="1 099 511 627 776"
pebi	Pi	2^{50} ="1 125 899 906 842 624"
exbi	Ei	2^{60} ="1 152 921 504 606 846 976"
zebi	Zi	2^{70} ="1 180 591 620 717 411 303 424"
yobi	Ui	2^{80} ="1 208 925 819 614 629 174 706 176"

Códigos de Representação de Dados

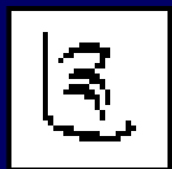
- ASCII (*American Standard Code for Interchange Information*)
  - Mais usado em microcomputadores
  - Representação de *256* caracteres diferentes (e.g. em um teclado alfanumérico) ⇨ codificação em *8* bits
    - *128* símbolos universais
    - *128* símbolos adicionais, passíveis de variações de país para país
  - Exemplo: Letra *A*
    - Representação: $41_{16} = 0100\ 0001_{2}$

Códigos de Representação de Dados

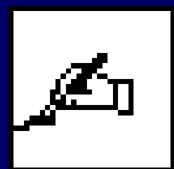
- **Unicode**
  - **Representação de *65536* caracteres diferentes ⇨ codificação em *16* bits**
  - **Modelado sobre o conjunto de caracteres *ASCII***
  - **Possibilita a codificação da maioria dos caracteres correntemente em uso**
  - **Usa scripts para a definição de caracteres em um idioma específico**



Grego



Tibetano



Dingbats

Katakana