

Computadores e Programação

2009–2010 1º semestre

Aula 16/10/2009

Helmut Wolters

Representação de dados no computador

Homem: os mãos têm 10 dedos \Rightarrow
sistema decimal

Computador: memória com elementos tipo
ligado/desligado:
sistema binário

Representação de dados no computador

Homem: os mãos têm 10 dedos \Rightarrow
sistema decimal

Computador: memória com elementos tipo
ligado/desligado:
sistema binário

Números inteiros

são **exactos** por definição, i.e. não têm erros de arredondamento ou de qualquer outra fonte.

Representação de dados num computador

Transformar entre sistema decimal e sistema binário:

$$x = 145 \text{ decimal}$$

Representação de dados num computador

Transformar entre sistema decimal e sistema binário:

$$\begin{aligned}x &= 145 \text{ decimal} \\ &= 1 \cdot 10^2 + 4 \cdot 10^1 + 5 \cdot 10^0\end{aligned}$$

Representação de dados num computador

Transformar entre sistema decimal e sistema binário:

$$\begin{aligned}x &= 145 \text{ decimal} \\ &= 1 \cdot 10^2 + 4 \cdot 10^1 + 5 \cdot 10^0\end{aligned}$$

n	2^n	
0	1	
1	2	
2	4	
3	8	
4	16	
5	32	
6	64	
7	128	
8	256	
\vdots	\vdots	

Representação de dados num computador

Transformar entre sistema decimal e sistema binário:

$$\begin{aligned}x &= 145 \text{ decimal} \\ &= 1 \cdot 10^2 + 4 \cdot 10^1 + 5 \cdot 10^0 \\ &= 128 + 17\end{aligned}$$

n	2^n	
0	1	
1	2	
2	4	
3	8	
4	16	
5	32	
6	64	
7	128	
8	256	
\vdots	\vdots	

Representação de dados num computador

Transformar entre sistema decimal e sistema binário:

$$\begin{aligned}x &= 145 \text{ decimal} \\ &= 1 \cdot 10^2 + 4 \cdot 10^1 + 5 \cdot 10^0 \\ &= 128 + 17\end{aligned}$$

n	2^n	
0	1	
1	2	
2	4	
3	8	
4	16	
5	32	
6	64	
7	128	1
8	256	0
\vdots	\vdots	0

Representação de dados num computador

Transformar entre sistema decimal e sistema binário:

$$\begin{aligned}x &= 145 \text{ decimal} \\ &= 1 \cdot 10^2 + 4 \cdot 10^1 + 5 \cdot 10^0 \\ &= 128 + 17 \\ &= 128 + 16 + 1\end{aligned}$$

n	2^n	
0	1	
1	2	
2	4	
3	8	
4	16	
5	32	
6	64	
7	128	1
8	256	0
\vdots	\vdots	0

Representação de dados num computador

Transformar entre sistema decimal e sistema binário:

$$\begin{aligned}x &= 145 \text{ decimal} \\ &= 1 \cdot 10^2 + 4 \cdot 10^1 + 5 \cdot 10^0 \\ &= 128 + 17 \\ &= 128 + 16 + 1\end{aligned}$$

n	2^n	
0	1	
1	2	
2	4	
3	8	
4	16	1
5	32	0
6	64	0
7	128	1
8	256	0
\vdots	\vdots	0

Representação de dados num computador

Transformar entre sistema decimal e sistema binário:

$$\begin{aligned}x &= 145 \text{ decimal} \\ &= 1 \cdot 10^2 + 4 \cdot 10^1 + 5 \cdot 10^0 \\ &= 128 + 17 \\ &= 128 + 16 + 1\end{aligned}$$

n	2^n	
0	1	1
1	2	0
2	4	0
3	8	0
4	16	1
5	32	0
6	64	0
7	128	1
8	256	0
\vdots	\vdots	0

Representação de dados num computador

Transformar entre sistema decimal e sistema binário:

$$\begin{aligned}x &= 145 \text{ decimal} \\ &= 1 \cdot 10^2 + 4 \cdot 10^1 + 5 \cdot 10^0 \\ &= 128 + 17 \\ &= 128 + 16 + 1 \\ &= 1 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 \\ &\quad + 0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 \\ &= 10010001 \text{ binário}\end{aligned}$$

n	2^n	
0	1	1
1	2	0
2	4	0
3	8	0
4	16	1
5	32	0
6	64	0
7	128	1
8	256	0
\vdots	\vdots	0

Representação de dados num computador

Tipicamente, um número inteiro num computador é representado

- no sistema binário

Representação de dados num computador

Tipicamente, um número inteiro num computador é representado

- no sistema binário
- com 32 algarismos,

Representação de dados num computador

Tipicamente, um número inteiro num computador é representado

- no sistema binário
- com 32 algarismos,
- destes um é utilizado para marcar o sinal.

Representação de dados num computador

Tipicamente, um número inteiro num computador é representado

- no sistema binário
- com 32 algarismos,
- destes um é utilizado para marcar o sinal.
- \Rightarrow domínio de $x \in [-2^{31}; 2^{31} - 1]$.

Representação de dados num computador

Tipicamente, um número inteiro num computador é representado

- no sistema binário
- com 32 algarismos,
- destes um é utilizado para marcar o sinal.
- \Rightarrow domínio de $x \in [-2^{31}; 2^{31} - 1]$.
- entre $-2\ 147\ 483\ 648$
e $2\ 147\ 483\ 647$

Representação de dados num computador

Números reais

No caso de números reais a situação é muito mais complicado.

Em princípio, podemos continuar a ideia, a juntar potências negativas:

$$\begin{aligned}x &= 2,75 \\ &= 2 \cdot 10^0 + 7 \cdot 10^{-1} + 5 \cdot 10^{-2} \\ &= 2 + 0,5 + 0,25 \\ &= 1 \cdot 2^1 + 0 \cdot 2^0 \\ &= +1 \cdot 2^{-1} + 1 \cdot 2^{-2} \\ &= 10,11 \quad \text{binário}\end{aligned}$$

n	2^n	
\vdots	\vdots	0
-3	0,125	0
-2	0,25	1
-1	0,5	1
		,
0	1	0
1	2	1
2	4	0
3	8	0
\vdots	\vdots	0

Representação de dados num computador

$$e = 2,7183\dots \textit{ decimal}$$

$$= 2 \cdot 10^0 + 7 \cdot 10^{-1} + 1 \cdot 10^{-2} + 8 \cdot 10^{-3} + 3 \cdot 10^{-4} \\ \pm 5 \cdot 10^{-5} \quad (0,00005)$$

$$= 2 + \frac{1}{2} + \frac{1}{8} + \frac{1}{16} + \frac{1}{64} + \frac{1}{128} + \frac{1}{256} + \frac{1}{512} \\ + \frac{1}{1024} + \frac{1}{2048} \pm \frac{1}{16384} \quad (\simeq 0,00006)$$

$$= 1 \cdot 2^1 + 0 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} \\ + 1 \cdot 2^{-4} + 0 \cdot 2^{-5} + 1 \cdot 2^{-6} + 1 \cdot 2^{-7} + 1 \cdot 2^{-8} \\ + 1 \cdot 2^{-9} + 1 \cdot 2^{-10} + 1 \cdot 2^{-11} \pm 2^{-12}$$

$$= 10,10110111111100\dots \textit{ binário}$$

Representação de dados num computador

$$\begin{aligned}y &= 1,1 \text{ decimal} \\ &= 1 + \frac{1}{16} + \frac{1}{32} + \frac{1}{256} + \frac{1}{512} + \frac{1}{4096} \pm \frac{1}{8192} \\ &= 1 \cdot 2^0 + 0 \cdot 2^{-1} + 0 \cdot 2^{-2} + 0 \cdot 2^{-3} + 1 \cdot 2^{-4} \\ &\quad + 1 \cdot 2^{-5} + 0 \cdot 2^{-6} + 0 \cdot 2^{-7} + 1 \cdot 2^{-8} + 1 \cdot 2^{-9} \\ &\quad + 0 \cdot 2^{-10} + 0 \cdot 2^{-11} + 1 \cdot 2^{-12} \pm 2^{-13} \\ &= 1,000110011001\dots \text{ binário aproximado} \\ &= 1,00011(0011) \text{ binário correcto}\end{aligned}$$

⇒ erros de arredondamento !

Representação de dados num computador

Tipicamente, um número real num computador é representado no sistema binário numa **notação semi-logarítmica**, i.e.

$$1234,5678 = 1,2345678 \cdot 10^5$$

$$\textit{mantissa} = 1,2345678000000$$

$$\textit{expoente} = 5$$

Na verdade, tudo isto no sistema binário.

Representação de dados num computador

IEEE Standard

(Institute of Electrical and Electronics Engineers)

IEEE 32-bit: 0101010101010101010101010101010101

sinal(bit 1) - expoente (bit 2-9) - mantissa (bit 10-32):

$$\text{valor} = (-1)^{\text{sinal}} \times 2^{(\text{expoente}-127)} \times (1 + \text{mantissa})$$

Representação de dados num computador

IEEE Standard

(Institute of Electrical and Electronics Engineers)

IEEE 32-bit: 01

sinal(bit 1) - expoente (bit 2-9) - mantissa (bit 10-32):

$$\text{valor} = (-1)^{\text{sinal}} \times 2^{(\text{expoente}-127)} \times (1 + \text{mantissa})$$

Exemplo: 0100000001000000000000000000000000000000000000

$$\text{sinal} = 0$$

$$\text{expoente} = 10000000_{bin} = 128$$

$$\text{mantissa} = 0,10000000000000000000000000000000_{bin} = 0,5$$

$$\text{valor} = (-1)^0 \times 2^{(128-127)} \times (1 + 0,5) = 3,000000$$

Representação de dados num computador

IEEE Standard

(Institute of Electrical and Electronics Engineers)

IEEE 32-bit: 0101010101010101010101010101010101

sinal(bit 1) - expoente (bit 2-9) - mantissa (bit 10-32):

$$\text{valor} = (-1)^{\text{sinal}} \times 2^{(\text{expoente}-127)} \times (1 + \text{mantissa})$$

Domínio: $\simeq \pm 10^{\pm 38}$ com $\simeq 7$ dígitos decimais

IEEE 64-bit:

sinal(bit 1) - expoente (bit 2-12) - mantissa (bit 13-64):

$$\text{valor} = (-1)^{\text{sinal}} \times 2^{(\text{expoente}-1023)} \times (1 + \text{mantissa})$$

Domínio: $\simeq \pm 10^{\pm 308}$ com $\simeq 14$ dígitos decimais

Caracteres — ASCII

American Standard Code for Information Interchange 1967

- 7 bits por character = $2^7 = 128$ combinações
- 00xxxxx: 0-31: códigos de controlo
- yyxxxxx: 32-127: dígitos, letras, símbolos de pontuação
- muito limitado, só permite guardar letras do alfabeto inglês, sem acentos
- mas até hoje o único standard que funciona em praticamente todos os sistemas
- na prática ocupa 8 bits (1 byte) com o 8º bit sempre 0
0xxxxxxx

Caracteres — Iso-Latin-1

Extensão para 8 bits para caracteres das línguas da Europa Ocidental

- 0xxxxxxx: ASCII
- 1xxxxxxx: mais $2^8=128$ combinações, para áâãäæ ...
- há outras extensões concorrentes não compatíveis (Latin-2, ..., Grego, Árabe, Hebraico, Windows codepages, ...)
- nem pensar nos símbolos Chineses, Japonêses, Sânscrito, ...

Caracteres — UTF-8

Unicode Transformation Format — desde 1991

- 1 até 4 bytes
- 0xxxxxxx: ASCII - 1 byte
- 110xxxxx 10xxxxxx:
2 bytes, mais $2^{11}=2048$ combinações
- 1110xxxx 10xxxxxx 10xxxxxx:
3 bytes, mais $2^{16}=65.536$ combinações
- 11110xxx 10xxxxxx 10xxxxxx 10xxxxxx:
4 bytes, mais $2^{21}=2.097.152$ combinações
- é compatível com ASCII
- está a ser implementado a nível internacional, pouco a pouco, em todos os sistemas