Estruturas de Dados

Algoritmos e Programação de Computadores

Guilherme N. Ramos
gnramos@unb.br

2015/2

Representação de Dados

Por que todo programa manipula dados [por definição]?

Tipos de dados: númericos, simbólicos e lógicos.

- *O tipo define o que o programa pode fazer com o dado.*

Como representar os dados [na memória] do computador?

Representação de Dados

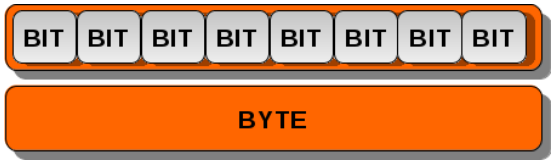
bit (*binary digit*)

Representa um estado binário:

"ligado" é representado pelo símbolo 1.

"desligado" é representado pelo símbolo 0.

A memória é um conjunto ordenado de *bits* que podem conter instruções ou dados.



`10100100111100010111000111010101001010 11`

- A representação do dado é uma só: *binária*!
- A interpretação dos bits define a informação.

`0xBF400000`

Sinal e Magnitude -4145152_{10}

Complemento de 1 -8486911_{10}

Complemento de 2 -1086324736_{10}

Ponto Flutuante (32) -0.75_{10}

`0x41200000`

Inteiro 1092616192_{10}

Ponto Flutuante (32) 10.0_{10}

ASCII A

Sistemas Numéricos

Bits, podem representar números pelo sistema numérico posicional[1]. Por exemplo, 123_{10}:

$$100 + 20 + 3 = 1 \cdot 10^2 + 2 \cdot 10^1 + 3 \cdot 10^0$$

O valor depende de cada algarismo (base numérica) e de sua posição, e pode ser facilmente obtido com a seguinte fórmula:

$$a_n a_{n-1} \cdots a_2 a_1 a_0 = a_n \cdot b^n + a_{n-1} \cdot b^{n-1} + \cdots + a_2 \cdot b^2 + a_1 \cdot b^1 + a_0$$

[1] 49 em algarismos romanos?

Sistemas Numéricos

Bases:

Hexadecimal $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$
Decimal $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
Octal $\{0, 1, 2, 3, 4, 5, 6, 7\}$
Binária $\{0, 1\}$

$$7B_{16} = 123_{10} = 173_8 = 1111011_2$$

$$75_{10} = (\quad)_2 = (\quad)_8 = (\quad)_{16}$$

Números Reais

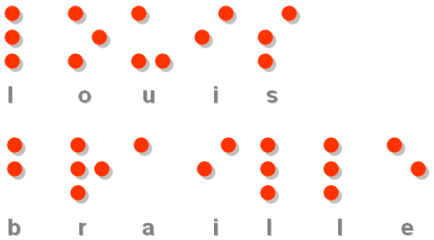
Reais - IEEE 754

$$(-1)^{sinal} \cdot (1 + \text{mantissa}) \cdot 2^{expoente - offset}$$

$$\begin{aligned}
&(-1)^1 \cdot 1,1 \cdot 2^{126-127} \\
&= -1,1 \cdot 2^{-1} \\
&= -0,11 \\
&= -(1 \cdot 2^{-1} + 1 \cdot 2^{-2}) \\
&= -(0,5 + 0,25) \\
&= -0,75
\end{aligned}$$

Símbolos

A *codificação de caracteres* é a associação de bits a símbolos.



Por necessidade de diálogos entre os diferentes computadores, foram criados diversos códigos objetivando a padronização.

Ponteiros

Cada variável declarada ocupa um espaço na memória, conforme seu tipo, e nome da váriável é apenas uma forma "amigável" de lidar com o endereço deste espaço.

← Ponteiro →

Tipo de dado que armazena um *endereço de memória*, possibilitando leitura e escrita deste endereço.

Atenção

Há uma diferença conceitual entre endereço e conteúdo. O endereço indica a localização na memória (onde está armazenado), o conteúdo indica o valor dos bits (o que está armazenado).

Ponteiros

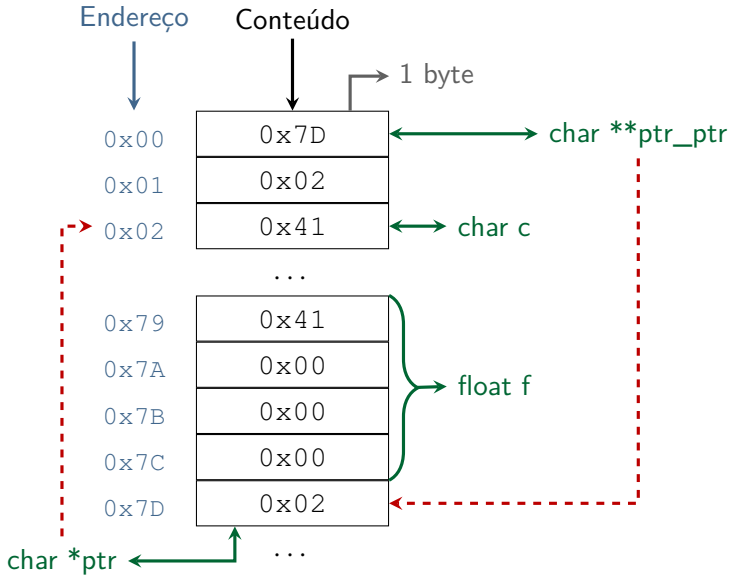
Em linguagem C, um ponteiro é declarado da seguinte forma:

```
tipo* identificador;
```

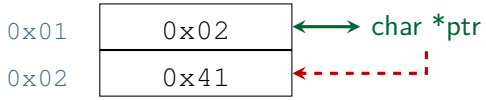
Por exemplo:

```
1 int*   ptr_int;      // ponteiro para inteiro
2 float* ptr_float;    // ponteiro para real
3 char*  ptr_char;     // ponteiro para caractere
4
5 int**  ptr_ptr_int; // ponteiro para (ponteiro para inteiro)
```


Ponteiros



Ponteiros



0-ponteiro.c

```
1  char c = 'A';
2  char* ptr = &c; /* Armazena o endereço de c */
3
4  /* O conteúdo de c é: */
5  printf("   c = %c\n", c);
6  /* O conteúdo de ptr é: */
7  printf(" ptr = %p\n", ptr);
8  /* O conteúdo do endereço apontado por ptr é: */
9  printf("*ptr = %c\n", *ptr);
10 /* O endereço de ptr é: */
11 printf("&ptr = %p\n", &ptr);
```