

Variáveis e Operadores

Algoritmos e Estrutura de Dados I

Instituto de Engenharia – UFMT

Agenda

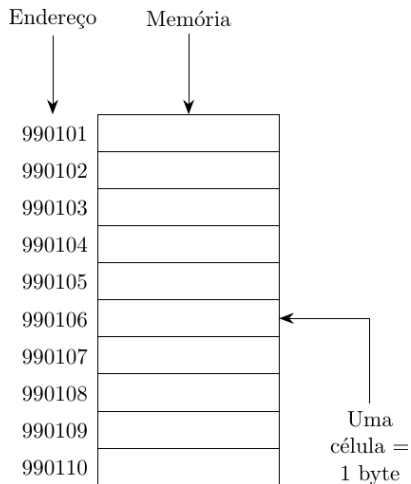
Objetivos

- Entender como funciona a memória do computador.
- Aprender a usar variáveis para armazenar dados na memória.
- Entender a representação para os tipos de variáveis.

Memória

- A memória pode ser entendida como uma sequência de células nas quais se pode armazenar um valor correspondente a um dado.
- Uma célula de memória armazena um único valor (conteúdo), que é representado por uma sequência de bits.
- Um **bit** nada mais é que um dos valores 0 ou 1. A uma sequência de 8 bits damos o nome de **byte**.

Memória



Representação da memória.

Memória

- Um programa nada mais faz que manipular valores que estão armazenados nas células de memória.
- O efeito final do programa é dado pelo conteúdo final das células de memória que ele manipula.

Exemplo algoritmo que calcula a média de dois valores:

- 1 Leia um numero e armazene no endereço 990101.
- 2 Leia outro numero e armazene no endereço 990102.
- 3 Some os valores nos endereços 990101 e 990102 e armazene no endereço 990103.
- 4 Divida o valor armazenado no endereço 990103 por dois e armazene o resultado no endereço 990104.
- 5 Informe o valor armazenado no endereço 990104.

Memória

- Essa forma complicada utiliza um nível de complexidade desnecessário.
- O uso de endereços fixos de memória torna-se impraticável para algoritmos longos.
- Por esse motivo, as linguagens de programação criaram o conceito de variáveis, para abstrair a complexidade do endereçamento das células de memória.

Variáveis

- Variáveis são abstrações da noção de célula de memória.
- São espaços de memória reservados que guardam valores durante a execução de um programa.
- Cada variável é caracterizada pelo seu rótulo e seu conteúdo.

Variáveis

Exemplo algoritmo que calcula a média de dois valores:

- 1 Leia um número e armazene na variável x .
- 2 Leia outro número e armazene na variável y .
- 3 Some os valores das variáveis x e y e armazene na variável z .
- 4 Divida o valor de z por dois e armazene o resultado na variável w .
- 5 Informe o valor da variável w .

Tipos de Variáveis

- O tipo de uma variável informa a quantidade de memória, em bytes, que a variável ocupará e a forma como um valor deverá ser armazenado.
- Há cinco tipos básicos de dados em C:

TIPO	BIT	BYTES	ESCALAS
char	8	1	-128 a 127
int	16	2	-32768 a 32767
float	32	4	3.4E-38 a 3.4E+38
double	64	8	1.7E-308 a 1.7E+308
void	0	0	Nenhum valor

Representação Binária

- As variáveis que armazenam números inteiros são declaradas como sendo do tipo int.
- Computadores utilizam o sistema binária
- No sistema decimal, a base de contagem é o número 10: alfabeto de 10 símbolos: 0,1,2,3,4,5,6,7,8,9.
- No sistema binário, a base de contagem é o número 2: alfabeto de 2 símbolos: 0,1.

Representação Binária

- O significado de cada dígito em determinada posição é o valor da casa multiplicado pelo valor do dígito e a quantidade representada é a soma de todos os produtos.
- A equação a seguir transforma um número em base B para decimal

$$x = a_n B^n + a_{n-1} B^{n-1} + \dots + a_0 B^0,$$

- onde cada a_i é um número inteiro não negativo e n é um valor que representa a posição mais significativo do número.

Representação Binária

Exemplos na base decimal:

- $(3547)_{10} = 3 \times 10^3 + 5 \times 10^2 + 4 \times 10^1 + 7 \times 10^0$.
- $(2371)_{10} = ?$

Representação Binária

Exemplos na base binária:

- $(1011)_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 11.$
- $(1101)_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 13$
- $(1010)_2 = ?$
- $(0111)_2 = ?$

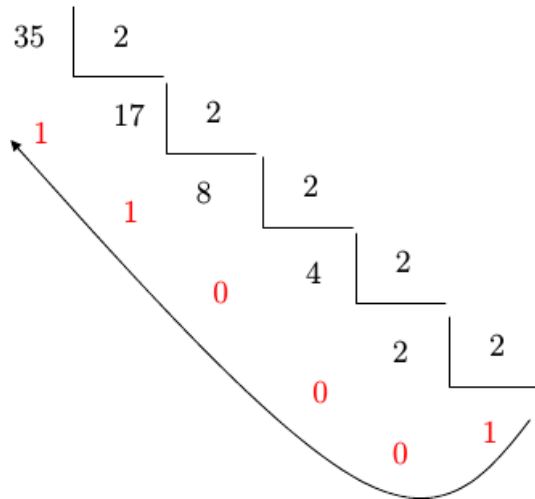
Representação Binária

- Com 1 bit podemos representar dois valores: 0 e 1.
- Com 2 bits podemos representar 4 valores: 00,01,10,11.
- Com 4 bits podemos representar 16 valores distintos.
- Com n bits podemos representar 2^n valores distintos.

Conversão Binário para Decimal

- Números binários são representados em grupos de 4 em 4.
- $(1001\ 1011)_2 = 1 \times 2^7 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^1 + 1 \times 2^0$.
- $(1001\ 1011)_2 = 128 + 16 + 8 + 2 + 1 = 155$.
- $(1011\ 1011)_2 = ?$

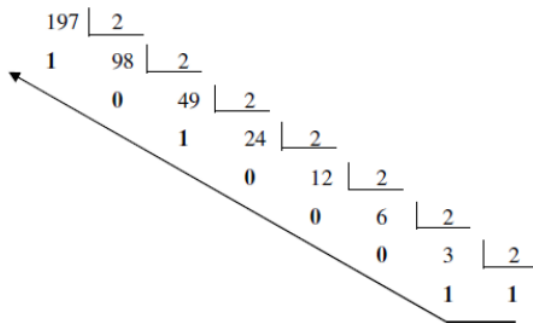
Conversão Decimal para Binário



- $(0010\ 0011)_2 = 1 \times 2^5 + 1 \times 2^1 + 1 \times 2^0$
- $(0010\ 0011)_2 = 32 + 2 + 1 = (35)_{10}$

Conversão Decimal para Binário

Exemplo: 197



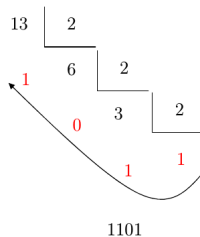
- $(1100\ 0101)_2 = 1 \times 2^7 + 1 \times 2^6 + 1 \times 2^2 + 1 \times 2^0$
- $(1100\ 0101)_2 = 128 + 64 + 4 + 1 = (197)_{10}$

Números fracionários

- É trivial para um computador atual tratar e operar com números inteiros.
- Entretanto, em muitas aplicações do dia a dia é necessário realizar operações com quantidades fracionárias, ou representar valores muito grandes ou muito pequenos.
- As variáveis que armazenam números fracionários são declaradas como sendo do tipo float.

Conversão decimal para ponto flutuante

Convertendo $(13.75)_{10}$



0.75

$$0.75 \times 2 = 1.50$$

$$0.50 \times 2 = 1.00$$

$$0.00 \times 2 = 0.00$$

$$(13.75)_{10} = (1101.110)_2$$

Conversão ponto flutuante para decimal

Convertendo $(1010\ 1110)_2$

- $(1010\ 1110)_2 = 1 \times 2^3 + 1 \times 2^1 + 1 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3}$
- $(1010\ 1110)_2 = 8 + 2 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8}$
- $(1010\ 1110)_2 = 10 + 0.5 + 0.25 + 0.125 = 10.875$

Caracteres

- As variáveis que armazenam um caractere são declaradas como sendo do tipo char.
- Uma variável do tipo char é armazenada internamente como um inteiro.
- Existe uma correspondência entre símbolos e números, dada pela tabela ASCII (*American Standard Code for Information Interchange*).

Caracteres

- Esta tabela utiliza os números de 0 até 127 para letras do alfabeto inglês e para os sinais de pontuação mais comuns.
- Por exemplo, na entrada 65 da tabela ASCII encontramos a letra 'A'.
- Assim, uma variável do tipo char cujo conteúdo seja 65 representaria também a letra 'A'.
- Infelizmente não existe consenso de como tratar os caracteres numerados a partir de 128, onde estariam os caracteres acentuados.

Fim

Fim

Variáveis e Operadores

Algoritmos e Estrutura de Dados I

Instituto de Engenharia – UFMT