

Projeto e Desenvolvimento de Algoritmos

Variáveis

Adriano Cruz e Jonas Knopman

Índice

- Objetivos
- Introdução
- Modelo de Memória
- Armazenamento de Dados Numéricos
 - **Dados Inteiros**
 - **Dados Reais**
- Armazenamento de Dados Literais
- Armazenamento de Dados Lógicos
- Variáveis

Objetivos

- Apresentar o modelo de memória do computador
- Mostrar como os diversos tipos de dados são armazenados na memória do computador.
- Apresentar o conceito de variável e sua utilidade no desenvolvimento de algoritmos.

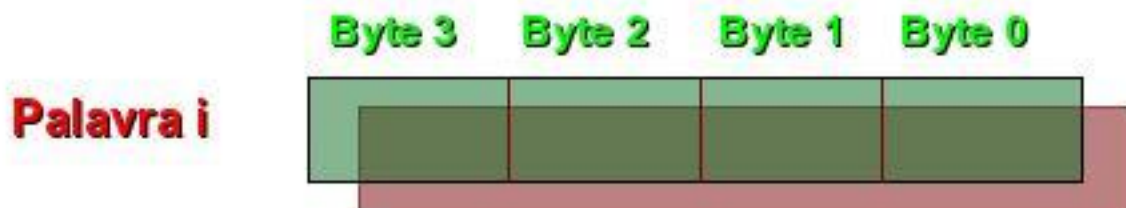
Memória

- A memória é um conjunto ordenado de células, cada um identificada por um endereço.
- A unidade mínima de informação na memória é o bit que consegue armazenar ou o valor 0 ou 1.
- Um conjunto de 8 bits forma um byte.
- Um conjunto de bytes, usualmente 4 (32 bits) forma uma palavra de memória.



Palavras e bytes

- Usualmente uma palavra de memória é composta de 4 bytes.



- A maioria das memórias são endereçadas por byte, portanto o endereço permite manipular o conteúdo de um determinado byte

Memória e endereçamento

- Memória com $2^n - 1$ palavras de 4 bytes.
- Então $n + 2$ é o número de bits do endereço.

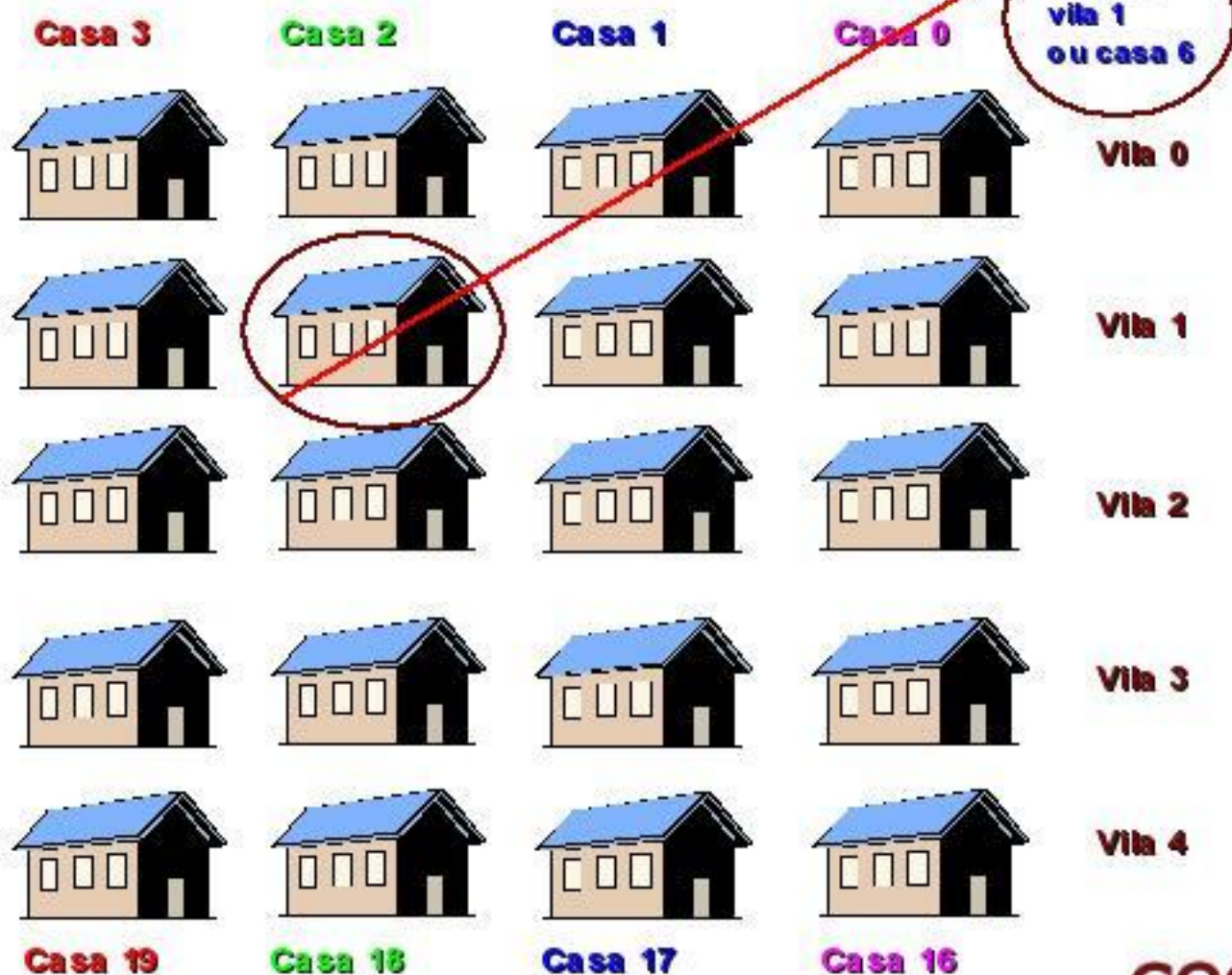


Memória e endereços

- Podemos pensar cada byte como uma casa.
- Uma palavra seria uma vila de casas.
- A memória uma rua de vilas.
- Suponha uma rua (memória) com 5 vilas (palavras), cada vila com 4 casas (bytes).



Memória e endereços



Memória e endereços

- Temos endereços de casas indo de casa 0 até casa 19.
- Temos endereços de vilas indo de vila 0 até vila 4.
- Posso dar meu endereço como casa 6 ou casa 2 da vila 1.
- Observar que 6 dividido por 4 (casas por vila) é igual a 1 com resto 2.
- *Em computação os endereços de memória começam sempre em 0.*

Exemplo

- Memória de 128 Mega (2^{20}) bytes
- Memória com $128 \times 2^{20} = 2^7 \times 2^{20} =$
 $128 \times 1.048.576$ bytes
- O endereço deve ter 27 bits
 - 7 bits devido aos $128 = 2^7$
 - 20 bits devido aos 2^{20}
- Se cada palavra de memória tem 4 bytes
temos então 32 Mega palavras

Bits e dados

- Cada tipo de dado requer uma quantidade de bits para armazenar o valor.
- Esta quantidade é variável e depende da linguagem e do computador.
- Atualmente computadores típicos possuem memórias com palavras de 32 bits, ou 4 bytes.

Armazenamento de Inteiros

- O conjunto dos números inteiros ($\mathbb{Z} = \{\dots, -2, -1, 0, +1, +2, \dots\}$) contém uma quantidade infinita de elementos.
- Como o número de bits disponíveis na memória para armazenar os números é finito não é possível representar todos os números deste conjunto.

Armazenamento de Inteiros

- Considere uma palavra de 32 bits.
- Um bit é reservado para o sinal.
- O bit mais significativo está à esquerda e o menos significativo à direita;
- Um número inteiro i pode variar entre

$$-2^{31} \leq i \leq 2^{31}-1$$

$$-2.147.483.648 \leq i \leq 2.147.483.647$$

- Como não temos $+0$ e -0 , há um número negativo a mais que os positivos.



Armazenamento de Inteiros

- Observar que não podemos armazenar números maiores que 2.147.483.647 e menores que -2.147.483.648.
- Isto pode ser contornado, em alguns casos, empregando-se os números reais que veremos em seguida.

Armazenamento de Reais

- Os números reais também são armazenados em 32 bits
- Estes números são, as vezes, chamados de números em ponto flutuante devido a forma como eles são processados.
- O método é parecido com a notação científica que algumas calculadoras empregam.



Notação Científica

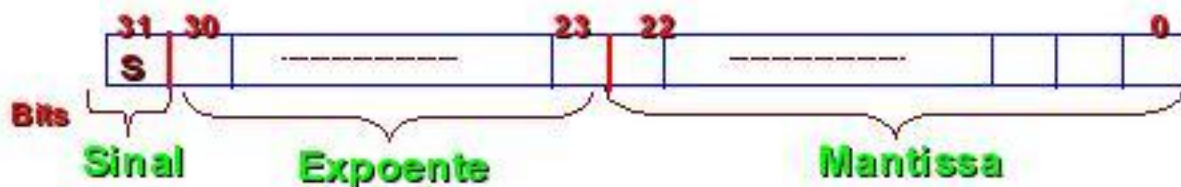
- Os números em notação científica nas calculadoras são expressos por um número real **normalizado** multiplicado por um número elevado a uma potência.
- Ex. $1,5 \text{ E}+8 = 1,5 \times 10^8$
- Um número normalizado implica que antes da vírgula somente deve aparecer um algarismo.
- Ex. $(5,0 \text{ E}+8) + (6,0 \text{ E}+8) = 11,0 \text{ E}+8$
- Resultado final = $1,10 \text{ E}+9$
- **Errata do Vídeo:** O correto é dizer que em $1,5 \text{ E}+8$ há 7 zeros antes da vírgula

Armazenamento de Reais

- O conjunto dos números reais contém uma quantidade infinita de elementos.
- Como o número de bits para armazenar os números é finito não é possível representar todos os números deste conjunto.

Reais em bits

- Considere uma palavra de 32 bits.
- Um bit é reservado para o sinal.
- Oito bits são reservados para o expoente.
- A base do expoente depende do computador, normalmente 2.
- Vinte e três bits são usados para o número real, chamado de mantissa.



Quais números reais?

- Computadores tem dificuldade de armazenar os números reais muito grandes e os muito pequenos.
- O subconjunto dos números reais disponíveis pode ser representado como



- O zero está incluído no conjunto.
- Para uma palavra de 32 bits temos $R_{MAX}=3.4E+38$ e $R_{MIN}=3.4E-38$

Armazenamento de caracteres

- O código de caracteres normalmente empregado é o ASCII que precisa de 8 bits ou um byte.
- Como os computadores endereçam a memória por bytes então é possível armazenar um caractere por byte.
- Para armazenar um conjunto de caracteres normalmente se emprega um conjunto de bytes.

Valores Lógicos

- Este tipo de dado somente possui dois valores **verdadeiro** e **falso**.
- Portanto um bit é suficiente para armazenar estes dados.
- Normalmente se usa um byte inteiro para armazenar valores lógicos devido a dificuldade de endereçar bits.

Variáveis

- Diversos tipos de dados são armazenados na memória.
- A memória é endereçada por meio de números.
- Para procurar um determinado dado na memória seria preciso saber o número da palavra (ou byte) onde este dado está armazenado.
- Este método seria complicado e ilegível.
- Mais fácil empregar nomes como fazemos com as ruas de nossa cidade.

Nomes de Variáveis

- Variáveis receberão nomes.
- Cada variável deve receber um nome diferente para poder ser identificada sem problemas.
- Estes nomes deverão ser utilizados sempre que quisermos modificar ou saber o conteúdo de uma posição na memória do computador.

Nomes de Variáveis

- As regras para criação dos nomes das variáveis são as seguintes:
 - Um nome de variável pode conter letras, algarismos e o caracter _ (sublinha);
 - Um nome de variável deve necessariamente começar por uma letra;
 - Um nome de variável não deve conter nenhum símbolo diferente de letra ou algarismo, exceto o símbolo _ (sublinha)
 - Não existe limitação para o número de caracteres do nome;
 - Não será feita diferenciação entre letras maiúsculas e minúsculas.

Dicas

- **Escolher nomes significativos para as variáveis**
 - P. Ex. salario, total, nota, pagamento
- **Nomes significativos ajudam a tornar os algoritmos e os programas auto-explicativos**
- **Nomes de variáveis com mais de uma palavra podem ajudar também**
- **Separe as palavras por sublinhados**
 - P. Ex. total_pagamentos, prova_final

Dicas

- **Não é necessário alongar desnecessariamente os nomes.**
 - P. Ex. `total_de_recebimentos_do_ano`,
`variavel_nota`
- **Evitar nomes que não ajudem o entendimento do algoritmo.**

Exemplos de nomes corretos

- soma
- salario_total
- nota_final
- prova1
- raio
- velocidade_inicial

Exemplos de nomes incorretos

- soma\$ \$ não é permitido
- salario total Espaço em branco não é permitido
- 2prova Não começou por uma letra
- Salario/hora / não é permitido