# Estrutura de Dados

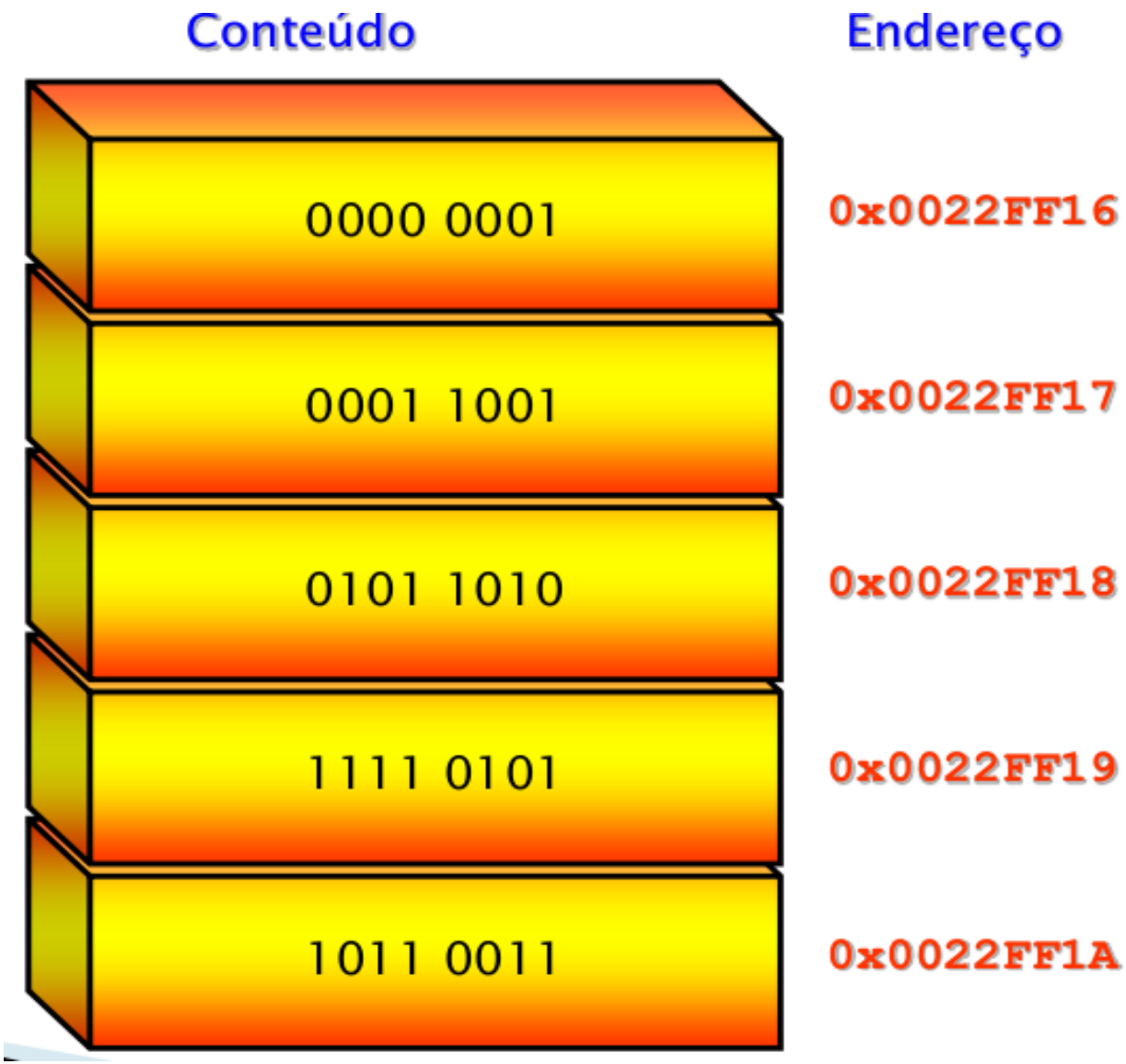
Prof. Rogerio Atem de Carvalho, D. Eng.

Aula 4: Endereçamento & Aritmética de Ponteiros

# Endereçamento

- A memória é uma sequência de bytes.
- Um byte armazena um número inteiro entre 0 e 255.
- Cada byte na memória é identificado por um endereço numérico, independentemente do tipo de conteúdo que armazena.

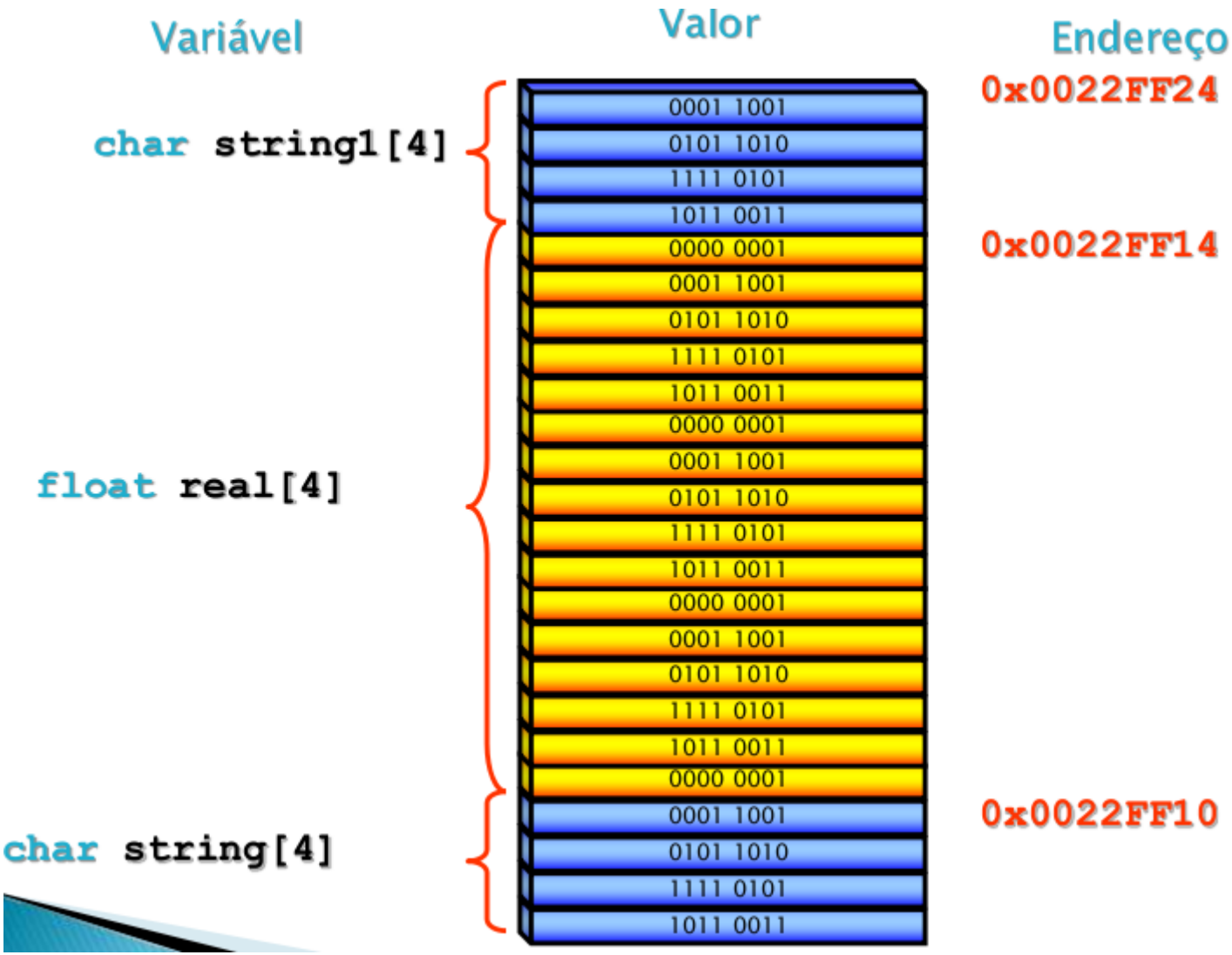
# Endereçamento



# Endereçamento

- Cada objeto que reside na memória ocupa uma determinada quantidade de bytes consecutivos:
  - Caracteres: 1 byte
  - Inteiros: 2 ou 4 bytes
  - Real: 4 bytes
- Cada objeto é referenciado por um endereço de memória, sendo este o primeiro byte que ocupa.

# Endereçamento



# Endereçamento

- Uma declaração de variável vai representar a associação de um nome a um endereço de memória, permitindo acesso a seu conteúdo.
- Ex.:

  int x = 10;

  Nome: x

  Conteúdo: 10

  Endereço: 0xfgd456d2

# Ponteiros

- Um variável tipo ponteiro (ou apontador) armazena um endereço de memória, ou seja, ela “aponta” para um endereço de memória.
- Um ponteiro pode armazenar o valor especial NULL quando não apontar para nenhum endereço.
- NULL é uma constante definida em <stdlib.h>

# Ponteiros

- Ponteiros são declarados empregando '*', que deve ser empregado também para acessar o conteúdo do endereço para o qual apontam
- Ex.:

```
int x;

int *pt_i;   // ponteiro para inteiro

pt_i  = &x; // pt_i aponta para x

*pt_i = 10;// mesmo que x = 10
```

# Ponteiros

- É necessário definir o tipo para o qual o ponteiro está apontando para que seja possível resolver as operações com ponteiros.
- Existem ponteiros para ponteiros, ex.:

  char **p;

# Aritmética de Ponteiros

- Conjunto de operações aritméticas disponíveis para ponteiros.
- A soma de um (p = p + 1;) incrementa o ponteiro do número de bytes relativo ao tamanho do tipo para o qual ele aponta.
- Para executar operações seguras empregar o operador sizeof()

# Aritmética de Ponteiros

- Ponteiros e matrizes
  - O nome de uma matriz é na realidade um ponteiro para seu primeiro elemento.
  - O acesso a um elemento de uma matriz unidimensional (vetor) segue a fórmula

    *(vetor + sizeof(<tipo>) * índice)

    sendo que a chamada a sizeof() é feita automaticamente pelo compilador.
  - Isto equivale a vetor[índice].

# Aritmética de Ponteiros

- Ponteiros e matrizes
  - Ex.:

```
int numeros[5] = {10, 15, 20, 25, 30};
for( i = 0; i < 5; i++)
{
   printf("%i \n", numeros[i]);
   printf("%i \n", *(numeros + i));
};
```

# Aritmética de Ponteiros

- Ponteiros e matrizes
  - Matriz constante de strings, exemplo:

```
char *erros_de_arquivo[] = {
        "Erro de abertura de arquivo \n",
        "Erro de leitura de arquivo \n",
        "Erro de escrita de arquivo \n",
        "Erro de fechamento de arquivo \n"
        };
```

# Próximo Tópico

Próximo Tópico: Alocação Dinâmica de Memória