

1. ARRAY

Um array é uma estrutura de dados que armazena uma coleção de elementos, todos do mesmo tipo. Os elementos são organizados de forma contígua na memória e são acessíveis por índices numéricos. Arrays são usados para armazenar dados sequenciais, como listas de números, caracteres ou outros tipos de dados.

- **Unidimensional:** Um array unidimensional (ou vetor) é uma lista linear de elementos. Por exemplo, um array de inteiros pode ser representado como [1, 2, 3, 4, 5].
- **Bidimensional:** Um array bidimensional (ou matriz) é uma grade ou tabela de elementos. Pode ser visualizado como um array de arrays. Por exemplo, uma matriz de inteiros.

2. ARMAZENAMENTO EM MEMÓRIA DE UM ARRAY

2.1 ARRAY UNIDIMENSIONAL

Os elementos de um array unidimensional são armazenados em locais de memória consecutivos. A localização de cada elemento é calculada com base no índice do elemento e no endereço base do array. Se `base` é o endereço inicial do array e `i` é o índice, a localização de `A[i]` é `base + i * sizeof(tipo)`.

2.2 ARRAY BIDIMENSIONAL

Existem duas formas comuns de armazenar um array bidimensional na memória:

- **Row-major order:** As linhas são armazenadas sequencialmente. A localização de `A[i][j]` pode ser calculada como `base + (i * num_colunas + j) * sizeof(tipo)`.
- **Column-major order:** As colunas são armazenadas sequencialmente. A localização de `A[i][j]` pode ser calculada como `base + (j * num_linhas + i) * sizeof(tipo)`.

3. STRING E SUAS FUNÇÕES DE MANIPULAÇÃO

Uma string é uma sequência de caracteres usada para representar texto. Em muitas linguagens de programação, as strings são arrays de caracteres terminados por um caractere nulo (`\0`).

3.1 MANIPULAÇÕES

- **Concatenar:** Combinar duas strings em uma. Exemplo: `strcat(destino, origem)` em C.
- **Comparar:** Comparar duas strings lexicograficamente. Exemplo: `strcmp(str1, str2)` em C.
- **Copiar:** Copiar uma string para outra. Exemplo: `strcpy(destino, origem)` em C.
- **Comprimento:** Determinar o comprimento de uma string. Exemplo: `strlen(string)` em C.
- **Substr:** Extrair uma substring de uma string maior. Em Python, por exemplo, `string[start:end]`.
- **Buscar:** Encontrar a posição de uma substring dentro de outra string. Exemplo: `strstr(haystack, needle)` em C.