Aula 28: Alocação Dinâmica (Parte 2)

Introdução a Programação

Túlio Toffolo & Puca Huachi
http://www.toffolo.com.br

BCC201 – 2019/2
Departamento de Computação – UFOP

Aulas anteriores

- Estruturas de memórias heterogêneas
- Alocação dinâmica
- Arquivos de texto

Aula de hoje

1. Exercícios (aulas práticas)
2. Alocação dinâmica (breve revisão)
3. Alocação dinâmica: erros comuns
4. Matrizes dinâmicas
5. Exercícios
6. Próximas aulas

Aula de hoje

Aula de hoje

Alocação dinâmica

Comando `malloc`:

- Faz parte da biblioteca `<stdlib.h>`.
- Aloca dinamicamente um bloco consecutivo de *bytes* na memória e retorna o endereço deste bloco.
- Isto permite escrever programas mais flexíveis.
- Exemplo de uso: alocar um vetor de tamanho definido pelo usuário...

Alocação dinâmica

Uso do método `malloc` para criar um `double`:

```
1  // aloca memória de forma dinâmica
2  double *nro = malloc(sizeof(double));
3
4  // altera o conteúdo da memória apontada por nro para 3.5
5  *nro = 3.5;
6
7  printf("Endereço de memória: %p\n", nro);
8  printf("Valor na memória: %lf\n", *nro);
```

- Este código imprimirá, por exemplo (arquitetura 64 bits):

```
1  Endereço de memória: 0x7feaf4400690
2  Valor na memória: 3.500000
```

Alocação dinâmica

Uso do método `malloc` para criar um bloco com 3 doubles:

```
1  // aloca memória de forma dinâmica e inicializa com valor 10.5
2  double *nro = malloc(3 * sizeof(double));
3
4  // alterando valores
5  nro[0] = 1.1;
6  nro[1] = 1.5;
7  nro[2] = 2.2;
8
9  for (int i = 0; i < 3; i++)
10     printf("%.1lf ", nro[i]);
```

- Este código imprimirá:

```
1  1.1 1.5 2.2
```

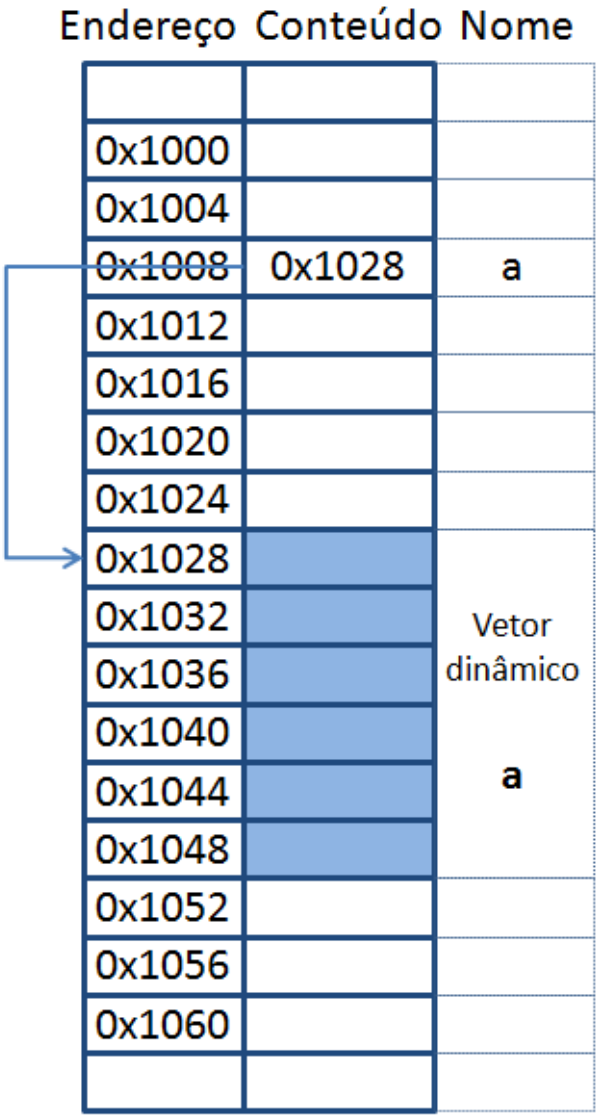
Exemplo de alocação dinâmica

Endereço	Conteúdo	Nome
0x1000		
0x1004		
0x1008	0x0000	a
0x1012		
0x1016		
0x1020		
0x1024		
0x1028		
0x1032		
0x1036		
0x1040		
0x1044		
0x1048		
0x1052		
0x1056		
0x1060		

```
1  int main() {
2   → int *a = NULL;
3      a = malloc(6 * sizeof(int));
4      for (int i = 0; i < 6; i++)
5          a[i] = i;
6      imprimeVetor3(a, 6);
7      ...
8  }
```

- Cria o ponteiro `a` com valor inicial `NULL` (0).

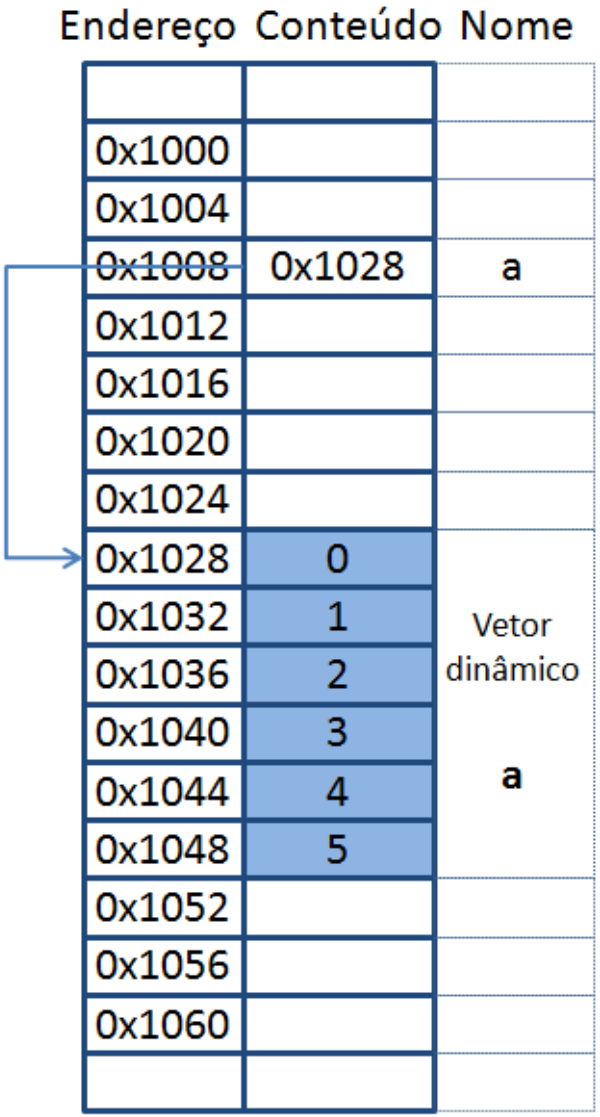
Exemplo de alocação dinâmica




```
1  int main() {
2      int *a = NULL;
3   →  a = malloc(6 * sizeof(int));
4      for (int i = 0; i < 6; i++)
5          a[i] = i;
6      imprimeVetor3(a, 6);
7      ...
8  }
```

- Aloca um bloco de memória com 6 inteiros (usando o comando `malloc`)
- Armazena o endereço de memória no ponteiro `a`.

Exemplo de alocação dinâmica




```
1  int main() {
2      int *a = NULL;
3      a = malloc(6 * sizeof(int));
4   →  for (int i = 0; i < 6; i++)
5   →      a[i] = i;
6      imprimeVetor3(a, 6);
7      ...
8  }
```

- Altera o valor de `a[0]`, `a[1]`, ..., `a[5]`

Muito importante: liberar a memória

- A memória alocada de forma estática pelo compilador é liberada automaticamente.
- Quando fazemos alocação dinâmica, **liberar a memória se torna nossa responsabilidade**.
- Em C usamos o procedimento `free`
- Exemplo (1):

```
1  int *a = malloc(sizeof(int));
2  ...
3  free(a);
```

Muito importante: liberar a memória

- A memória alocada de forma estática pelo compilador é liberada automaticamente.
- Quando fazemos alocação dinâmica, **liberar a memória se torna nossa responsabilidade**.
- Em C usamos o operador `free`
- Exemplo (2):

```
1  int *a = malloc(100 * sizeof(int));
2  ...
3  free(a);
```

Aula de hoje

Alocação dinâmica: erros comuns

Qual o erro no código abaixo?

```c
int main()
{
    int n;
    printf("Qual o tamanho do vetor? ");
    scanf("%d", &n);

    int *vetor = malloc(n * sizeof(int));

    usaVetor(vetor, n); // função que faz algum uso do vetor...

    printf("Qual o tamanho do segundo vetor? ");
    scanf("%d", &n);
    vetor = malloc(n * sizeof(int));

    usaVetor(vetor, n); // função que faz o segundo uso do vetor...

    free(vetor);
    return 0;
}
```

Qual o erro no código abaixo?

```c
int *criaPreencheVetor(int tamanho)
{
    int *vetor = malloc(tamanho * sizeof(int));
    for (int i = 0; i < tamanho; i++) {
        vetor[i] = i;
    }

    return vetor;
}

int main()
{
    int n;
    printf("Qual o tamanho do vetor? ");
    scanf("%d", &n);

    int *vetor = malloc(n * sizeof(int));
    vetor = criaPreencheVetor(n);

    free(vetor);
    return 0;
}
```

Aula de hoje

Matrizes dinâmicas

Matrizes dinâmicas são, na verdade, vetores de vetores!

- Criamos um vetor de ponteiros
- Criamos, para cada ponteiro, um vetor!

Matrizes dinâmicas

Exemplo:

```
1  // função que cria uma matriz de tamanho n x m
2  int ** criaMatriz(int n, int m)
3  {
4      int **matriz;
5      matriz = malloc(n * sizeof(int*));
6      for (int i = 0; i < n; i++) {
7          matriz[i] = malloc(m * sizeof(int));
8      }
9      return matriz;
10 }
```

Mas... como liberar a memória alocada para a matriz?

Matrizes dinâmicas

Como liberar a memória alocada por uma matriz?

- Lembre-se: um `free` para cada `mallloc`!

Exemplo:

```
1  void liberaMatriz(int **matriz, int n, int m)
2  {
3      for (int i = 0; i < n; i++)
4          free(matriz[i]);
5      free(matriz);
6  }
```

- Obs: note que a variável `m` não é necessária!

Erros comuns

O que está errado no código abaixo?

```
1  void liberaMatriz(int **matriz, int n, int m)
2   {
3       free(matriz);
4       for (int i = 0; i < n; i++)
5           free(matriz[i]);
6   }
```

Note que neste contexto a **a ordem** faz toda a diferença!

- Após liberar a memória apontada por `matriz`, não podemos mais acessar `matriz[i]`.

Aula de hoje

Exercícios

Exercício 1

Defina uma função que retorne a transposta de uma matriz `mat` com `lins`$\times$`cols` elementos. Utilize o protótipo a seguir:

```
1  int ** transposta(int **mat, int lins, int cols);
```

Dica: dada uma matriz A de tamanho $l \times c$, lembre-se que sua transposta T tem tamanho $c \times l$ e cada célula $T_{i,j} = A_{j,i}$.

Aula de hoje

Próxima aula

- Alocação dinâmica (parte 3)
- Arquivos binários
- Revisão

Perguntas?