Aula 27: Alocação Dinâmica

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

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`:

```c
// aloca memória de forma dinâmica
double *nro = malloc(sizeof(double));

// altera o conteúdo da memória apontada por nro para 3.5
*nro = 3.5;

printf("Endereço de memória: %p\n", nro);
printf("Valor na memória: %lf\n", *nro);
```

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

```
Endereço de memória: 0x7feaf4400690
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
```

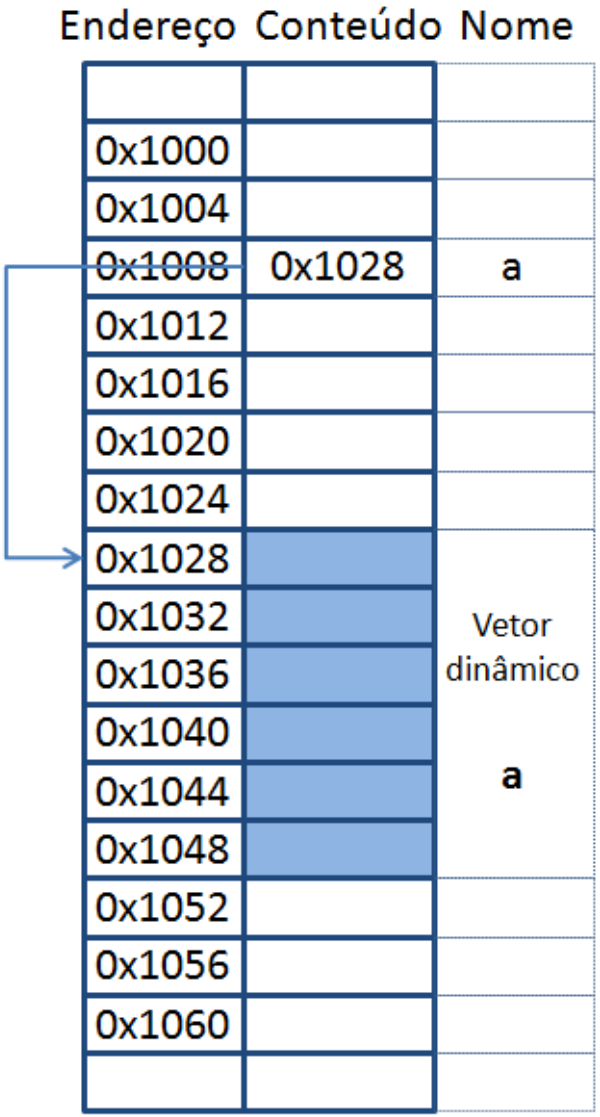
Exemplo de alocação dinâmica

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).

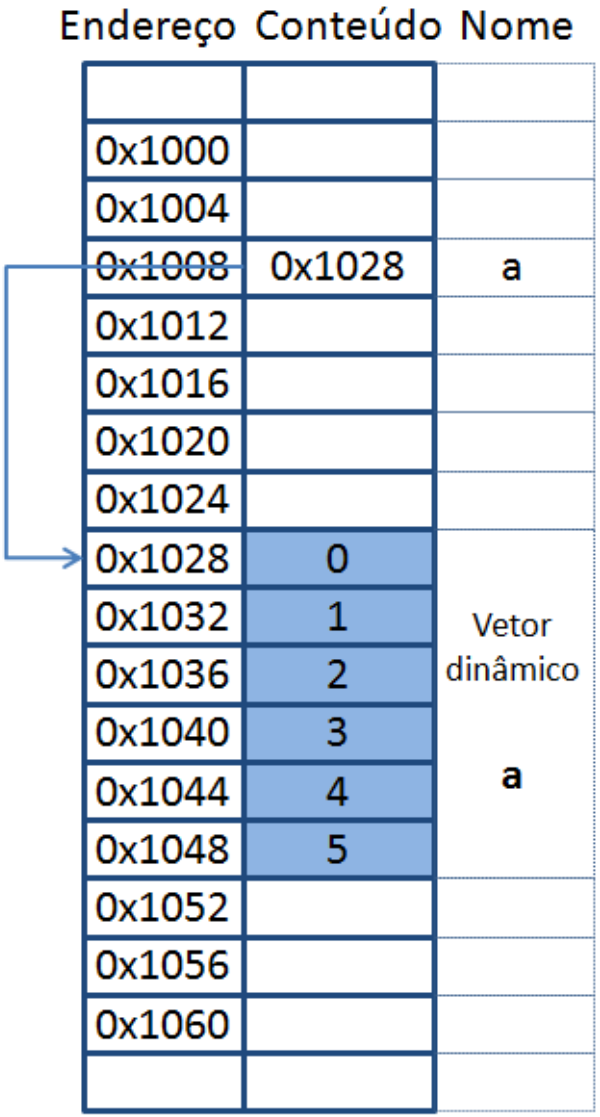
Exemplo de alocação dinâmica




```
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 procedimento `free`
- Exemplo (2):

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

Exemplo:

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

```c
#include <stdlib.h>
#include <stdio.h>

void leVetor(int *v, int n) {
    for (int i = 0; i < n; i++)
        scanf("%d", &v[i]);
}

int *maior(int *v, int n) {
    int *maior = v;
    for (int i = 1; i < n; i++)
        if (v[i] > *maior)
            maior = v + i;
    return maior;
}

int main() {
    int n, *v;
    scanf("%d", &n);
    v = malloc(n * sizeof(int));
    leVetor(v, n);
    int *valor = maior(v, n);
    printf("Maior = %d\n", *valor);
    free(v);
    return 0;
}
```

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

Próximas aula

- Alocação dinâmica (parte 2)
- Alocação dinâmica (parte 3)
- Arquivos binários
- Revisão sobre alocação dinâmica
- Revisão para Prova 03

Aula de hoje

Exercícios

Exercício 1

Crie uma função que:

1. recebe um vetor `v` e seu tamanho `n` por parâmetro;
2. cria um novo vetor por alocação dinâmica, preenchendo-o com o conteúdo de `v` em ordem inversa;
3. retorna este novo vetor.

Dica: utilize o protótipo a seguir:

```
1  int *inverso(int *v, int n);
```

- Crie um exemplo de utilização desta função no método `main()`.
- Não se esqueça de liberar a memória (usando `free`)!

Perguntas?