

Alocação de Memória

Estrutura de Dados

Prof. Msc. Felipe Leivas Teixeira

Versão 1.0

1 Alocação de Memória

2 Alocação de Memória: malloc

3 Alocação de Memória: calloc

4 Alocação de Memória: realloc

5 Alocação de Memória: free

6 Exercícios

Alocação de Memória

Felipe
Teixeira

Alocação de Memória

Alocação
de
Memória:
malloc

Alocação
de
Memória:
calloc

Alocação
de
Memória:
realloc

Alocação
de
Memória:
free

Exercícios

1 Alocação de Memória

2 Alocação de Memória: malloc

3 Alocação de Memória: calloc

4 Alocação de Memória: realloc

5 Alocação de Memória: free

6 Exercícios

Alocação de Memória

Alocação
de Memória

Felipe
Teixeira

Alocação
de Memória

Alocação
de
Memória:
malloc

Alocação
de
Memória:
calloc

Alocação
de
Memória:
realloc

Alocação
de
Memória:
free

Exercícios

- Na alocação dinâmica de memória, os espaços de memória são alocados e liberados durante a execução do programa

Alocação de Memória

Alocação de Memória

Felipe
Teixeira

Alocação de Memória

Alocação
de
Memória:
malloc

Alocação
de
Memória:
calloc

Alocação
de
Memória:
realloc

Alocação
de
Memória:
free

Exercícios

- Na alocação dinâmica de memória, os espaços de memória são alocados e liberados durante a execução do programa
- A alocação de memória permite otimizar a utilização da memória, garantindo que uma área só será alocada e tornada disponível para o programa quando realmente for entrar em uso

Alocação de Memória

Alocação
de Memória

Felipe
Teixeira

Alocação
de Memória

Alocação
de
Memória:
malloc

Alocação
de
Memória:
calloc

Alocação
de
Memória:
realloc

Alocação
de
Memória:
free

Exercícios

- Na alocação dinâmica de memória, o usuário solicita o espaço requerido durante a execução, no momento em que isso se torna necessário

Alocação de Memória

Alocação de Memória

Felipe
Teixeira

Alocação de Memória

Alocação
de
Memória:
malloc

Alocação
de
Memória:
calloc

Alocação
de
Memória:
realloc

Alocação
de
Memória:
free

Exercícios

- Na alocação dinâmica de memória, o usuário solicita o espaço requerido durante a execução, no momento em que isso se torna necessário
- No momento da alocação, devem ser informados tanto a quantidade de memória necessária, quanto o tipo dos valores a serem armazenados

- Na alocação dinâmica de memória, o usuário solicita o espaço requerido durante a execução, no momento em que isso se torna necessário
- No momento da alocação, devem ser informados tanto a quantidade de memória necessária, quanto o tipo dos valores a serem armazenados
- Tão logo o espaço não seja mais necessário, em qualquer ponto da execução do programa, ele deve ser liberado

Alocação
de Memória

Felipe
Teixeira

Alocação
de Memória

Alocação
de
Memória:
malloc

Alocação
de
Memória:
calloc

Alocação
de
Memória:
realloc

Alocação
de
Memória:
free

Exercícios

- O resultado de uma tentativa de alocação de memória deve ser sempre verificado e tratado adequadamente

Alocação de Memória

Felipe
Teixeira

Alocação de Memória

Alocação
de
Memória:
malloc

Alocação
de
Memória:
calloc

Alocação
de
Memória:
realloc

Alocação
de
Memória:
free

Exercícios

- O resultado de uma tentativa de alocação de memória deve ser sempre verificado e tratado adequadamente
- Se uma alocação não é bem-sucedida pela inexistência de espaço para seu atendimento, ocorre um erro durante a execução

Alocação de Memória

Alocação de Memória

Felipe
Teixeira

Alocação de Memória

Alocação
de
Memória:
malloc

Alocação
de
Memória:
calloc

Alocação
de
Memória:
realloc

Alocação
de
Memória:
free

Exercícios

- A alocação dinâmica de memória é feita por meio de ponteiros

- A alocação dinâmica de memória é feita por meio de ponteiros
- Uma vez que a alocação tenha sido realizada, é devolvido ao programa o endereço físico do espaço alocado

Alocação de Memória

Alocação de Memória

Felipe
Teixeira

Alocação de Memória

Alocação
de
Memória:
malloc

Alocação
de
Memória:
calloc

Alocação
de
Memória:
realloc

Alocação
de
Memória:
free

Exercícios

- A alocação dinâmica de memória é feita por meio de ponteiros
- Uma vez que a alocação tenha sido realizada, é devolvido ao programa o endereço físico do espaço alocado
- O endereço da variável dinâmica alocada é armazenado em um ponteiro

- Em C, as funções relativas à alocação dinâmica de memória estão na biblioteca **stdlib.h**

Alocação de Memória

Alocação de Memória

Felipe
Teixeira

Alocação de Memória

Alocação
de
Memória:
malloc

Alocação
de
Memória:
calloc

Alocação
de
Memória:
realloc

Alocação
de
Memória:
free

Exercícios

- Em C, as funções relativas à alocação dinâmica de memória estão na biblioteca **stdlib.h**
- Na linguagem C existem duas funções de alocação dinâmica de memória, são elas:

Alocação de Memória

Alocação de Memória

Felipe
Teixeira

Alocação de Memória

Alocação
de
Memória:
malloc

Alocação
de
Memória:
calloc

Alocação
de
Memória:
realloc

Alocação
de
Memória:
free

Exercícios

- Em C, as funções relativas à alocação dinâmica de memória estão na biblioteca **stdlib.h**
- Na linguagem C existem duas funções de alocação dinâmica de memória, são elas:
 - **malloc**

Alocação de Memória

Alocação de Memória

Felipe
Teixeira

Alocação de Memória

Alocação
de
Memória:
malloc

Alocação
de
Memória:
calloc

Alocação
de
Memória:
realloc

Alocação
de
Memória:
free

Exercícios

- Em C, as funções relativas à alocação dinâmica de memória estão na biblioteca **stdlib.h**
- Na linguagem C existem duas funções de alocação dinâmica de memória, são elas:
 - **malloc**
 - **calloc**

- Em C, as funções relativas à alocação dinâmica de memória estão na biblioteca **stdlib.h**
- Na linguagem C existem duas funções de alocação dinâmica de memória, são elas:
 - **malloc**
 - **calloc**
- Além das duas funções de alocação, existe uma função que realoca memória dinamicamente. Essa função é chamada de **realloc**

- Em C, as funções relativas à alocação dinâmica de memória estão na biblioteca **stdlib.h**
- Na linguagem C existem duas funções de alocação dinâmica de memória, são elas:
 - **malloc**
 - **calloc**
- Além das duas funções de alocação, existe uma função que realoca memória dinamicamente. Essa função é chamada de **realloc**
- As funções de alocação e a função de realocação de memória, devolvem um ponteiro **void ***, ou seja, um ponteiro para qualquer tipo de dado. Devido a isso sempre deve ser feito o **casting** para o tipo desejado

- Em C, as funções relativas à alocação dinâmica de memória estão na biblioteca **stdlib.h**
- Na linguagem C existem duas funções de alocação dinâmica de memória, são elas:
 - **malloc**
 - **calloc**
- Além das duas funções de alocação, existe uma função que realoca memória dinamicamente. Essa função é chamada de **realloc**
- As funções de alocação e a função de realocação de memória, devolvem um ponteiro **void ***, ou seja, um ponteiro para qualquer tipo de dado. Devido a isso sempre deve ser feito o **casting** para o tipo desejado
- Por fim, quando uma memória, alocada dinamicamente, não é mais utilizada, ela pode ser liberada pela função **free**

1 Alocação de Memória

2 Alocação de Memória: malloc

3 Alocação de Memória: calloc

4 Alocação de Memória: realloc

5 Alocação de Memória: free

6 Exercícios

Alocação de Memória: malloc

Alocação
de Memória

Felipe
Teixeira

Alocação
de Memória

Alocação
de
Memória:
malloc

Alocação
de
Memória:
calloc

Alocação
de
Memória:
realloc

Alocação
de
Memória:
free

Exercícios

- A função malloc é usada para criar um bloco de memória com um determinado número de bytes

Alocação de Memória: malloc

Alocação
de Memória

Felipe
Teixeira

Alocação
de Memória

Alocação
de
Memória:
malloc

Alocação
de
Memória:
calloc

Alocação
de
Memória:
realloc

Alocação
de
Memória:
free

Exercícios

- A função malloc é usada para criar um bloco de memória com um determinado número de bytes
- Se o bloco puder ser criado, a função devolve um ponteiro **void *** para o primeiro byte alocado, caso contrário, é devolvido um ponteiro NULL

- A função malloc é usada para criar um bloco de memória com um determinado número de bytes
- Se o bloco puder ser criado, a função devolve um ponteiro **void *** para o primeiro byte alocado, caso contrário, é devolvido um ponteiro NULL
- O formato da função malloc é a seguinte:

void * malloc(<número de bytes do bloco>)

Alocação
de Memória

Felipe
Teixeira

Alocação
de Memória

Alocação
de
Memória:
malloc

Alocação
de
Memória:
calloc

Alocação
de
Memória:
realloc

Alocação
de
Memória:
free

Exercícios

EXEMPLO MALLOC

Alocação de Memória

Felipe
Teixeira

Alocação
de Memória

Alocação
de
Memória:
malloc

Alocação
de
Memória:
calloc

Alocação
de
Memória:
realloc

Alocação
de
Memória:
free

Exercícios

1 Alocação de Memória

2 Alocação de Memória: malloc

3 Alocação de Memória: calloc

4 Alocação de Memória: realloc

5 Alocação de Memória: free

6 Exercícios

Alocação de Memória: calloc

Alocação de Memória

Felipe
Teixeira

Alocação de Memória

Alocação
de
Memória:
malloc

**Alocação
de
Memória:
calloc**

Alocação
de
Memória:
realloc

Alocação
de
Memória:
free

Exercícios

- A função calloc é usada para alocar memória para um vetor

Alocação de Memória: calloc

Alocação
de Memória

Felipe
Teixeira

Alocação
de Memória

Alocação
de
Memória:
malloc

Alocação
de
Memória:
calloc

Alocação
de
Memória:
realloc

Alocação
de
Memória:
free

Exercícios

- A função calloc é usada para alocar memória para um vetor
- O número de elementos do vetor e o tamanho, em bytes, de cada elemento são especificados na chamada da função

Alocação de Memória: calloc

Alocação de Memória

Felipe
Teixeira

Alocação de Memória

Alocação de Memória: malloc

Alocação de Memória: calloc

Alocação de Memória: realloc

Alocação de Memória: free

Exercícios

- A função calloc é usada para alocar memória para um vetor
- O número de elementos do vetor e o tamanho, em bytes, de cada elemento são especificados na chamada da função
- A memória alocada é inicializada com zeros

Alocação de Memória: calloc

Alocação de Memória

Felipe
Teixeira

Alocação de Memória

Alocação de Memória: malloc

Alocação de Memória: calloc

Alocação de Memória: realloc

Alocação de Memória: free

Exercícios

- A função calloc é usada para alocar memória para um vetor
- O número de elementos do vetor e o tamanho, em bytes, de cada elemento são especificados na chamada da função
- A memória alocada é inicializada com zeros
- Se a alocação for bem-sucedida, é devolvido um ponteiro para a área criada, caso contrário, é devolvido um ponteiro nulo

- A função calloc é usada para alocar memória para um vetor
- O número de elementos do vetor e o tamanho, em bytes, de cada elemento são especificados na chamada da função
- A memória alocada é inicializada com zeros
- Se a alocação for bem-sucedida, é devolvido um ponteiro para a área criada, caso contrário, é devolvido um ponteiro nulo
- O formato da função calloc é a seguinte:

void * calloc(<número de elementos>, <número de bytes de cada elemento>)

Alocação
de Memória

Felipe
Teixeira

Alocação
de Memória

Alocação
de
Memória:
malloc

Alocação
de
Memória:
calloc

Alocação
de
Memória:
realloc

Alocação
de
Memória:
free

Exercícios

EXEMPLO CALLOC

1 Alocação de Memória

2 Alocação de Memória: malloc

3 Alocação de Memória: calloc

4 Alocação de Memória: realloc

5 Alocação de Memória: free

6 Exercícios

Alocação de Memória: realloc

Alocação de Memória

Felipe
Teixeira

Alocação de Memória

Alocação
de
Memória:
malloc

Alocação
de
Memória:
calloc

Alocação
de
Memória:
realloc

Alocação
de
Memória:
free

Exercícios

- A função realloc altera o tamanho da memória apontada por um ponteiro para um determinado tamanho de bytes passado como parâmetro

Alocação de Memória: realloc

Alocação de Memória

Felipe
Teixeira

Alocação de Memória

Alocação de Memória: malloc

Alocação de Memória: calloc

Alocação de Memória: realloc

Alocação de Memória: free

Exercícios

- A função realloc altera o tamanho da memória apontada por um ponteiro para um determinado tamanho de bytes passado como parâmetro
- O valor de bytes a serem realocados pode ser maior ou menor que o original.

- A função realloc altera o tamanho da memória apontada por um ponteiro para um determinado tamanho de bytes passado como parâmetro
- O valor de bytes a serem realocados pode ser maior ou menor que o original.
- Um ponteiro para o bloco é devolvido porque realloc pode precisar mover o bloco para aumentar seu tamanho. Se isso ocorrer, o conteúdo do bloco antigo é copiado no novo bloco, e nenhuma informação é perdida

Alocação de Memória: realloc

Alocação
de Memória

Felipe
Teixeira

Alocação
de Memória

Alocação
de
Memória:
malloc

Alocação
de
Memória:
calloc

Alocação
de
Memória:
realloc

Alocação
de
Memória:
free

Exercícios

- A função realloc altera o tamanho da memória apontada por um ponteiro para um determinado tamanho de bytes passado como parâmetro
- O valor de bytes a serem realocados pode ser maior ou menor que o original.
- Um ponteiro para o bloco é devolvido porque realloc pode precisar mover o bloco para aumentar seu tamanho. Se isso ocorrer, o conteúdo do bloco antigo é copiado no novo bloco, e nenhuma informação é perdida
- Se não houver memória suficiente para a realocação, um ponteiro nulo é devolvido e o bloco original é deixado inalterado

Alocação de Memória: realloc

Alocação de Memória

Felipe
Teixeira

Alocação de Memória

Alocação de Memória: malloc

Alocação de Memória: calloc

Alocação de Memória: realloc

Alocação de Memória: free

Exercícios

- A função realloc altera o tamanho da memória apontada por um ponteiro para um determinado tamanho de bytes passado como parâmetro
- O valor de bytes a serem realocados pode ser maior ou menor que o original.
- Um ponteiro para o bloco é devolvido porque realloc pode precisar mover o bloco para aumentar seu tamanho. Se isso ocorrer, o conteúdo do bloco antigo é copiado no novo bloco, e nenhuma informação é perdida
- Se não houver memória suficiente para a realocação, um ponteiro nulo é devolvido e o bloco original é deixado inalterado
- O formato da função realloc é a seguinte:

void * realloc(<ponteiro>, <número de bytes do novo bloco>)

Alocação
de Memória

Felipe
Teixeira

Alocação
de Memória

Alocação
de
Memória:
malloc

Alocação
de
Memória:
calloc

**Alocação
de
Memória:
realloc**

Alocação
de
Memória:
free

Exercícios

EXEMPLO REALLOC

Alocação de Memória

Felipe
Teixeira

Alocação
de Memória

Alocação
de
Memória:
malloc

Alocação
de
Memória:
calloc

Alocação
de
Memória:
realloc

Alocação
de
Memória:
free

Exercícios

1 Alocação de Memória

2 Alocação de Memória: malloc

3 Alocação de Memória: calloc

4 Alocação de Memória: realloc

5 Alocação de Memória: free

6 Exercícios

Alocação de Memória: free

Alocação de Memória

Felipe
Teixeira

Alocação de Memória

Alocação
de
Memória:
malloc

Alocação
de
Memória:
calloc

Alocação
de
Memória:
realloc

Alocação
de
Memória:
free

Exercícios

- Quando alocamos memória dinamicamente é necessário que nós a liberemos quando ela não for mais necessária

Alocação de Memória: free

Alocação
de Memória

Felipe
Teixeira

Alocação
de Memória

Alocação
de
Memória:
malloc

Alocação
de
Memória:
calloc

Alocação
de
Memória:
realloc

Alocação
de
Memória:
free

Exercícios

- Quando alocamos memória dinamicamente é necessário que nós a liberemos quando ela não for mais necessária
- Para isto existe a função free cuja forma é:

`void free(<ponteiro que aponta para uma memória alocada>)`

- Quando alocamos memória dinamicamente é necessário que nós a liberemos quando ela não for mais necessária
- Para isto existe a função free cuja forma é:
void free(<ponteiro que aponta para uma memória alocada>)
- Após a desalocação da memória, convém não deixar ponteiros soltos no seu programa. Portanto, depois de cada free(ptr), atribua NULL a ptr.

Alocação
de Memória

Felipe
Teixeira

Alocação
de Memória

Alocação
de
Memória:
malloc

Alocação
de
Memória:
calloc

Alocação
de
Memória:
realloc

Alocação
de
Memória:
free

Exercícios

EXEMPLO FREE

Alocação de Memória

Felipe
Teixeira

Alocação
de Memória

Alocação
de
Memória:
malloc

Alocação
de
Memória:
calloc

Alocação
de
Memória:
realloc

Alocação
de
Memória:
free

Exercícios

1 Alocação de Memória

2 Alocação de Memória: malloc

3 Alocação de Memória: calloc

4 Alocação de Memória: realloc

5 Alocação de Memória: free

6 Exercícios

- 1** Escreva um programa que solicita ao usuário a quantidade de alunos de uma turma e aloca um vetor de notas (números reais). Depois de ler as notas, imprime a média aritmética. Obs: não deve ocorrer desperdício de memória; e após ser utilizada a memória deve ser devolvida.
- 2** Implemente uma função chamada newcalloc(), que faz exatamente o que a calloc() faz. Faça ela usando as funções malloc() e a memset(), que pertence a biblioteca string.h e recebe três argumentos (o ponteiro, o que queremos colocar em todas as posições do vetor e o número de bytes): memset(ptr, 1, numero * tamanho_em_bytes).
- 3** Escreva um programa que tenha inicialmente um vetor de 10 posições, onde o usuário pode inserir valores neste vetor quantas vezes ele quiser e quando ele decidir parar, os valores armazenados no vetor devem ser impressos. Sempre que o vetor estiver com mais de 70% das posições preenchidas, ele deve dobrar de tamanho. Dica: Faça um menu para o usuário poder escolher se inseri um novo elemento ou finaliza a computação.

**Alocação
de Memória**

Felipe
Teixeira

Alocação
de Memória

Alocação
de
Memória:
malloc

Alocação
de
Memória:
calloc

Alocação
de
Memória:
realloc

Alocação
de
Memória:
free

Exercícios

Alocação de Memória

Estrutura de Dados

Prof. Msc. Felipe Leivas Teixeira

Versão 1.0