



Introdução à Programação

Eduardo Silva Lira

XLVIII Programa de Verão do IME-USP

São Paulo - SP, Jan 2019



INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
UNIVERSIDADE DE SÃO PAULO



Dica!

- Ponteiros podem ou devem ser inicializados?!
 - **NULL**

Alocação dinâmica de memória

- É possível alocar a quantidade desejada de memória, no momento desejado?
 - Não sei quantos alunos uma escola tem
 - O meu sistema roda em escolas diferentes

Alocação dinâmica de memória

- Utilizar ponteiro
 - O sistema nos disponibiliza um bloco
 - Apontamos para este bloco

Alocação dinâmica de memória

- Funções úteis
 - malloc
 - realloc
 - calloc
 - free

Alocação dinâmica de memória

- Funções úteis
 - malloc
 - realloc
 - calloc
 - free
- Onde estas funções estão?
`#include <stdlib.h>`

Alocação dinâmica de memória

- Funções úteis
 - **malloc**
 - Aloca **n** bytes
 - **realloc**
 - Altera quantidade de armazenamento
 - Realoca se necessário
 - **calloc**
 - Aloca **n** posições de tamanho **size**
 - Inicializa-os para **zero**
 - **free**
 - Devolte ao sistema um bloco

Alocação dinâmica de memória

- Funções úteis
 - **malloc**
 - Aloca **n** bytes
 - **realloc**
 - Altera quantidade de armazenamento
 - Realoca se necessário
 - **calloc**
 - Aloca **n** posições de tamanho **size**
 - Inicializa-os para **zero**
 - **free**
 - Devolte ao sistema um bloco



Importante!

Alocação dinâmica de memória

- Função **malloc**: Aloca **size** bytes
 - Sintaxe
 - Retorna um ponteiro
 - Retorna **NULL** se não alocou
 - Recebe **size** em bytes

```
void *malloc (size_t size);
```

Alocação dinâmica de memória

- Função **calloc**: Aloca **n** objetos de tamanho **size** bytes. Inicializa com zero.
 - Sintaxe
 - Retorna um ponteiro
 - Retorna **NULL** se não alocou
 - Recebe **size** em bytes
 - Recebe também **nobj**

```
void *calloc (size_t nobj, size_t size);
```

Alocação dinâmica de memória

- Função **realloc**: realocar o objeto para o qual **p** aponta.
 - Sintaxe
 - Retorna um ponteiro
 - Retorna **NULL** se não encontrou
 - Recebe **size** em bytes (novo tamanho)
 - Recebe o ponteiro **p**

```
void *realloc(void *p, size_t size);
```

Liberação de memória

- Espaços não mais ocupados são liberados automaticamente em C?

Liberação de memória

- Espaços não mais ocupados são liberados automaticamente em C?
 - Defina não mais ocupados!
 - Alocação **estática**
 - Sim
 - Alocação **dinâmica**
 - Não

Liberação de memória

- Espaços não mais ocupados são liberados automaticamente em C?
 - A solução:
 - Utilizar a função **free**

Alocação dinâmica - Exercícios

- Faça um programa para
 - Ler uma quantidade **n**
 - Alocar **n** espaços em memória para guardar números inteiros positivos
 - Ao digitar um valor inválido, exibir todos os valores, na ordem digitada

- Para saber mais sobre as opções do compilador gcc:
 - <https://gcc.gnu.org/onlinedocs/gcc/Option-Index.html>
- Mais sobre bibliotecas:
 - https://www.tutorialspoint.com/c_standard_library/index.htm
- Alguns fóruns ativos para tirar dúvidas
 - <http://clubedaprogramacao.com/forum/index.php>
 - <https://pt.stackoverflow.com/>

Dúvidas?