ALGORITMOS E ESTRUTURAS DE DADOS I

Prof. Caio César de Freitas Dantas

Alocação Dinâmica de Memória

Alocação estática

- Os tipos de dados tem tamanho predefinido.
- O compilador vai alocar de forma automática o espaço de memória necessário.
- A alocação estática é feita em tempo de compilação.

- Este tipo de alocação tende a desperdiçar recursos, já que nem sempre é possível determinar previamente qual é o espaço necessário para armazenar as informações.

- Quando não se conhece o espaço total necessário, a tendência é o programador exagerar pois é melhor superdimensionar do que faltar espaço.

Alocação Dinâmica de Memória

Alocação Dinâmica

- Na alocação dinâmica podemos alocar espaços durante a execução de um programa, ou seja, a alocação dinâmica é feita em tempo de execução.
- Isto é bem interessante do ponto de vista do programador, pois permite que o espaço em memória seja alocado apenas quando necessário.
- Além disso, a alocação dinâmica permite aumentar ou até diminuir a quantidade de memória alocada.

Alocação Dinâmica de Memória

A linguagem C oferece meios de requisitarmos espaços de memória em tempo de execução.

Uso da memória

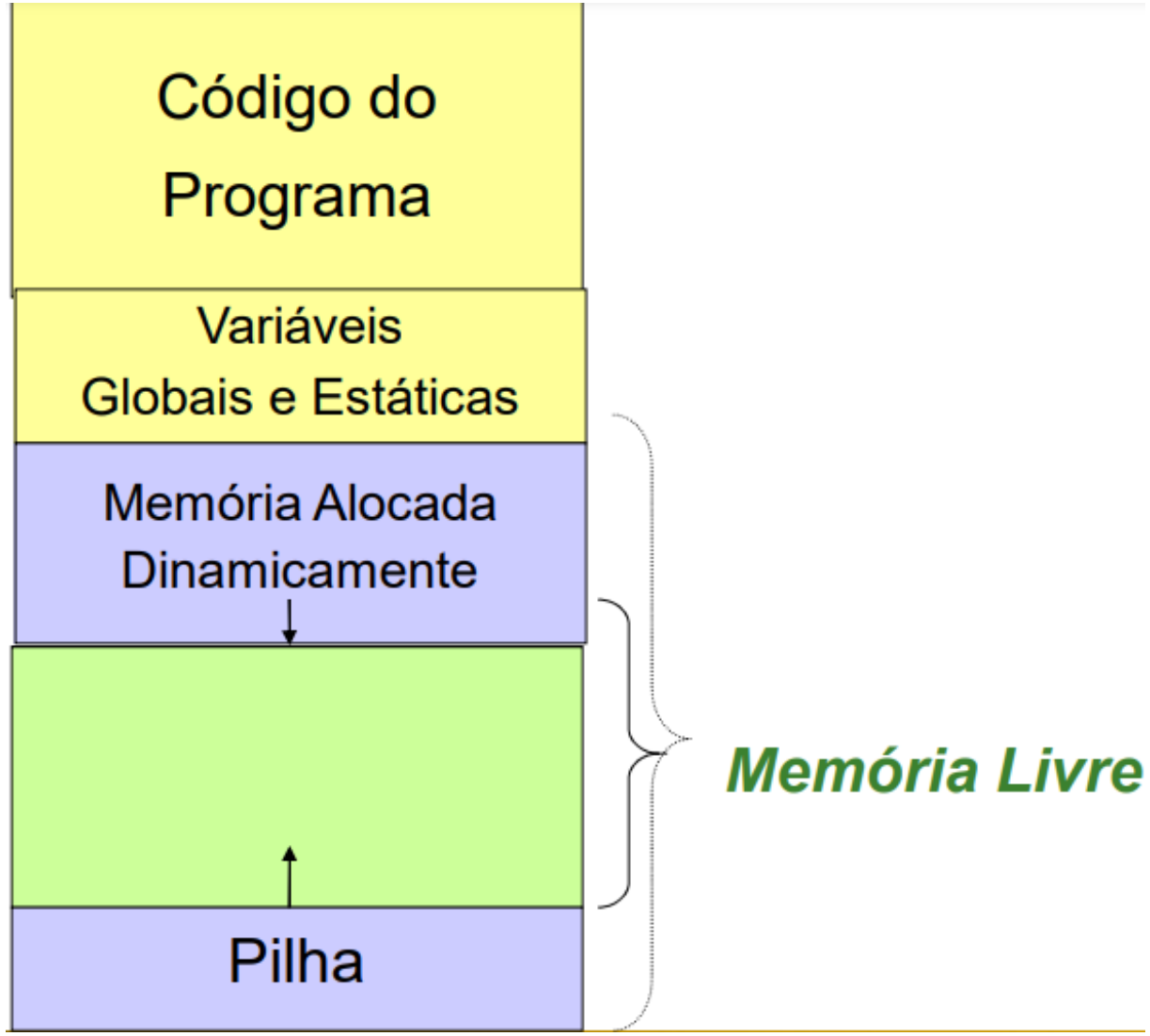
- Uso de variáveis globais (e estáticas). O espaço reservado para uma variável global existe enquanto o programa estiver sendo executado.
- Uso de variáveis locais. Neste caso, o espaço existe apenas enquanto a função que declarou a variável está sendo executada, sendo liberado para outros usos quando a execução da função termina.
- Reservar memória requisitando ao sistema, em tempo de execução, um espaço de um determinado tamanho.

Alocação Dinâmica de Memória

- O espaço alocado dinamicamente permanece reservado até que explicitamente seja liberado pelo programa. Por isso, podemos alocar dinamicamente um espaço de memória numa função e acessá-lo em outra.

- A partir do momento que liberarmos o espaço, ele estará disponibilizado para outros usos e não podemos mais acessá-lo. Se o programa não liberar um espaço alocado, este será automaticamente liberado quando a execução do programa terminar.

Alocação Dinâmica de Memória



Alocação Dinâmica de Memória

- Para executar um determinado programa, o S.O. carrega na memória o código do programa, em linguagem de máquina.

- Além disso, o S.O. reserva os espaços necessários para armazenar as variáveis globais (e estáticas) do programa

- O restante da memória livre é utilizado pelas variáveis locais e pelas variáveis alocadas dinamicamente.

Alocação Dinâmica de Memória

- Cada vez que uma função é chamada, o S.O. reserva o espaço necessário para as variáveis locais da função.
    - Este espaço pertence à pilha de execução e, quando a função termina, é liberado.
- A memória não ocupada pela pilha de execução pode ser requisitada dinamicamente.
    - Se a pilha tentar crescer mais do que o espaço disponível existente, dizemos que ela “estourou” e o programa é abortado com erro.

Alocação Dinâmica de Memória

Para permitir que o espaço para estruturas de dados de um programa possam ser alocados durante a execução do mesmo, a linguagem C define funções de biblioteca chamadas funções de alocação dinâmica de memória.

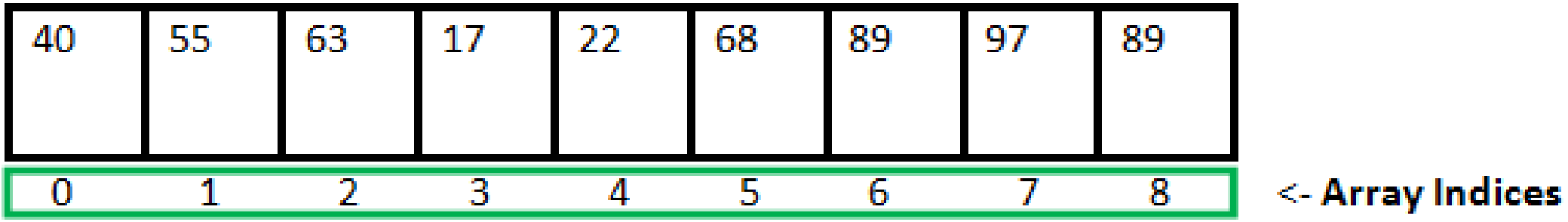
Estas funções são:

```
#include <stdlib.h>

void *malloc (int tamanho)
void *calloc (int numero_elementos, int tamanho_elemento)
void *realloc (void *area_alocada, int novo_tamanho)
void free (void *area_alocada)
```

Alocação Dinâmica de Memória

- Como C é uma linguagem estruturada, ela possui algumas regras fixas de programação.
- Um deles inclui alterar o tamanho de uma array.
- Uma array é uma coleção de itens armazenados em locais de memória contíguos.



Array Length = 9
First Index = 0
Last Index = 8

Alocação Dinâmica de Memória

Como pode ser visto que o comprimento (tamanho) do array acima feito é 9. Mas e se houver a necessidade de alterar este comprimento (tamanho). Por exemplo:

- Se houver uma situação em que apenas 5 elementos sejam necessários para serem inseridos nesta array. Neste caso, os 4 índices restantes estão apenas desperdiçando memória neste array. Portanto, é necessário diminuir o comprimento (tamanho) da array de 9 para 5.

- Há uma array de 9 elementos com todos os 9 índices preenchidos. Mas é necessário inserir mais 3 elementos nesta array. Neste caso, são necessários mais 3 índices. Portanto, o comprimento (tamanho) da array precisa ser alterado de 9 para 12.

Alocação Dinâmica de Memória

- Como eu peço para o usuário informar o tamanho do vetor?
  - A forma mais segura de fazer isso é com alocação dinâmica de memória.
- Para alocar dinamicamente o tamanho do vetor, utilizamos a função malloc.
- A função malloc (memory allocation) aloca espaço para um bloco de bytes consecutivos na memória RAM do computador e devolve o endereço desse bloco. O número de bytes é especificado no argumento da função.

Alocação Dinâmica de Memória

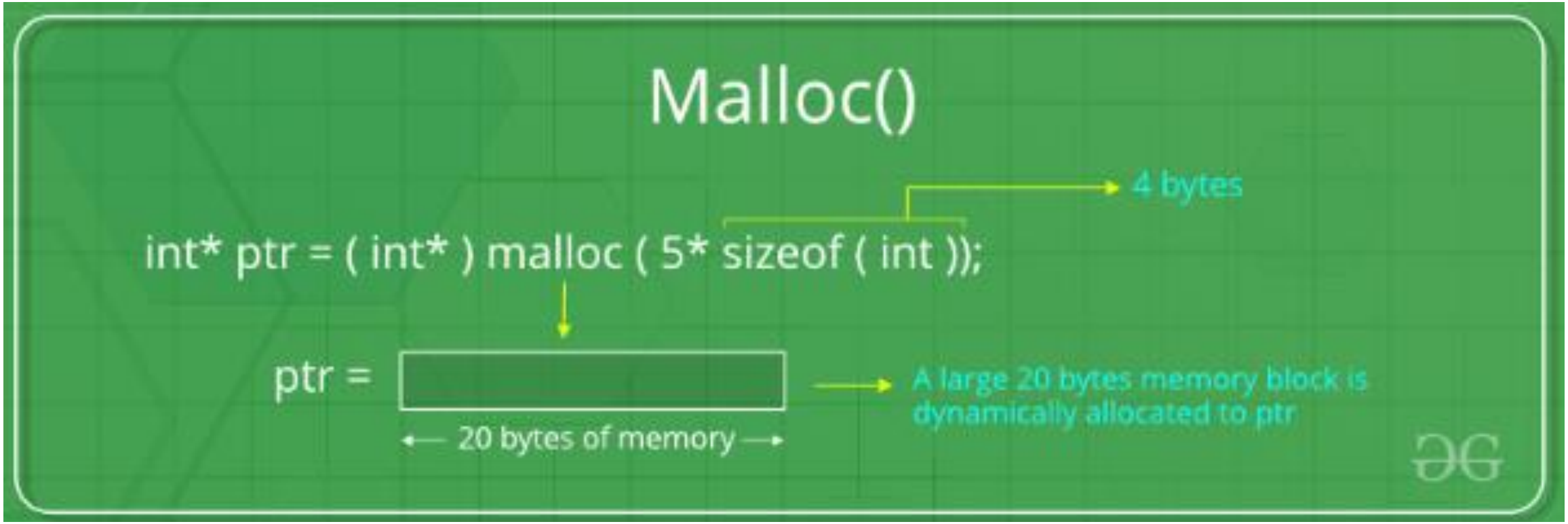
- A Função Malloc

```
ptr = (Tipo_dado*) malloc(byte-size)
Por Exemplo:
ptr = (int *) malloc (100 * sizeof (int));
```

Como o tamanho de int é de 4 bytes, essa instrução alocará 400 bytes de memória.
E, o ponteiro ptr contém o endereço do primeiro byte na memória alocada.

Alocação Dinâmica de Memória

- A Função Malloc



Se o espaço for insuficiente, a alocação falha e retorna um ponteiro NULL.

Alocação Dinâmica de Memória

```c
// Obtém o número de elementos para o array
printf("Digite a quantidade de elementos:");
scanf("%d",&n);

// Alocar memória dinamicamente usando malloc()
ptr = (int*)malloc(n * sizeof(int));

// Verifica se a memória foi bem sucedida
// alocado por malloc ou não
if (ptr == NULL) {
    printf("Memória não alocada.\n");
    exit(0);
}
else {
    // A memória foi alocada com sucesso
    printf("Memória alocada com sucesso usando malloc.\n");
```

Alocação Dinâmica de Memória

- A Função Malloc

```
Insira o número de elementos: 5
Memória alocada com sucesso usando malloc.
Os elementos da array são: 1, 2, 3, 4, 5,
```

Alocação Dinâmica de Memória

- A Função Calloc

- O método “calloc” ou “alocação contígua” em C é usado para alocar dinamicamente o número especificado de blocos de memória do tipo especificado. É muito semelhante a malloc(), mas tem dois pontos diferentes e são:

    - Ele inicializa cada bloco com um valor padrão '0’.
    - Ele tem dois parâmetros ou argumentos comparados a malloc().

Alocação Dinâmica de Memória

- A Função Calloc

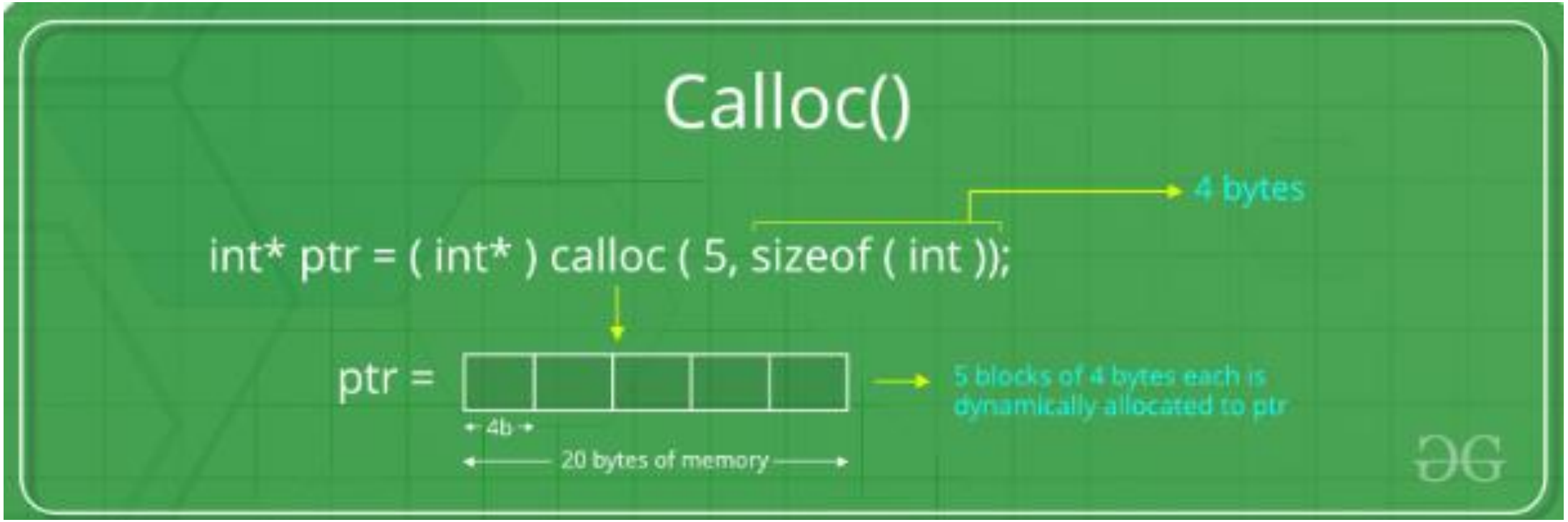
```
ptr = (Tipo_dado*)calloc(n, element-size);
n é o número de elementos e element-size é o tamanho de cada elemento.

Por exemplo:
ptr = (float *) calloc (25, sizeof (float));
```

Essa instrução aloca espaço contíguo na memória para 25 elementos, cada um com o tamanho do float.

Alocação Dinâmica de Memória

- A Função Calloc



Se o espaço for insuficiente, a alocação falha e retorna um ponteiro NULL.

Alocação Dinâmica de Memória

```
// Obtém o número de elementos para o array
printf("Digite a quantidade de elementos:");
scanf("%d",&n);
    // Alocar memória dinamicamente usando calloc()
ptr = (int*)calloc(n, sizeof(int));
 // Verifica se a memória foi bem sucedida
// alocado por calloc ou não
if (ptr == NULL) {
    printf("Memória não alocada.\n");
    exit(0);
}
else {
    // A memória foi alocada com sucesso
    printf("Memória alocada com sucesso usando calloc.\n");
```

Alocação Dinâmica de Memória

- A Função Calloc

```
Insira o número de elementos: 5
Memória alocada com sucesso usando calloc.
Os elementos da array são: 1, 2, 3, 4, 5,
```

Alocação Dinâmica de Memória

- A Função Free;
- O método “livre” em C é usado para desalocar dinamicamente a memória.
- A memória alocada usando as funções malloc() e calloc() não é desalocada por conta própria.
- Portanto, o método free() é usado, sempre que ocorre a alocação dinâmica de memória.
- Ajuda a reduzir o desperdício de memória ao liberá-la.

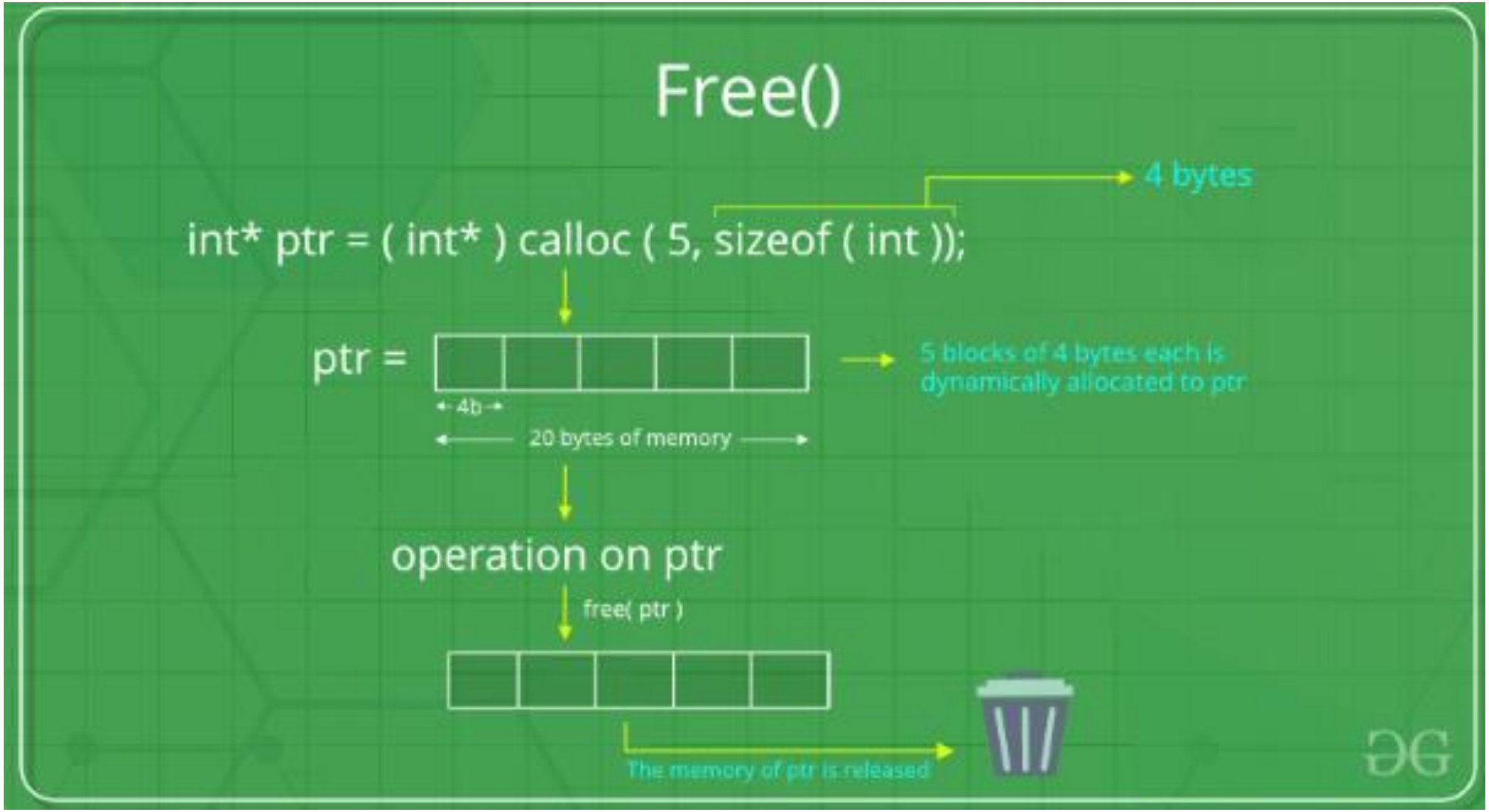
Alocação Dinâmica de Memória

- A Função Free;

- Sintaxe:

```
free(ptr);
```

Alocação Dinâmica de Memória



Alocação Dinâmica de Memória

```c
// Obtém o número de elementos para o array
printf("Digite a quantidade de elementos:");
scanf("%d",&n);
// Alocar memória dinamicamente usando malloc()
ptr = (int*)malloc(n * sizeof(int));
 // Alocar memória dinamicamente usando calloc()
ptr1 = (int*)calloc(n, sizeof(int));
 // Verifica se a memória foi alocada
if (ptr == NULL || ptr1 == NULL) {
    printf("Memória não alocada.\n");
    exit(0);
}
```

Alocação Dinâmica de Memória

```
else {
    // A memória foi alocada com sucesso
    printf("Memória alocada com sucesso usando malloc.\n");
    // Libera a memória
    free(ptr)
    printf("Memória Malloc liberada com sucesso.\n");
    // A memória foi alocada com sucesso
    printf("Memória alocada com sucesso usando calloc.\n");
    // Libera a memória
    free(ptr1)
    printf("Memória Calloc liberada com sucesso.\n");
```

Alocação Dinâmica de Memória

- A Função Free;

```
Insira o número de elementos: 5
Memória alocada com sucesso usando malloc.
Memória Malloc liberada com sucesso.

Memória alocada com sucesso usando calloc.
Memória Calloc liberada com sucesso.
```

Alocação Dinâmica de Memória

- A Função Realloc;
- O método de “realocação ” ou “realocação” em C é usado para alterar dinamicamente a alocação de memória de uma memória previamente alocada.
- Em outras palavras, se a memória alocada anteriormente com a ajuda de malloc ou calloc for insuficiente, realloc pode ser usado para realocar a memória dinamicamente.
- A realocação de memória mantém o valor já presente e novos blocos serão inicializados com o valor de lixo padrão.

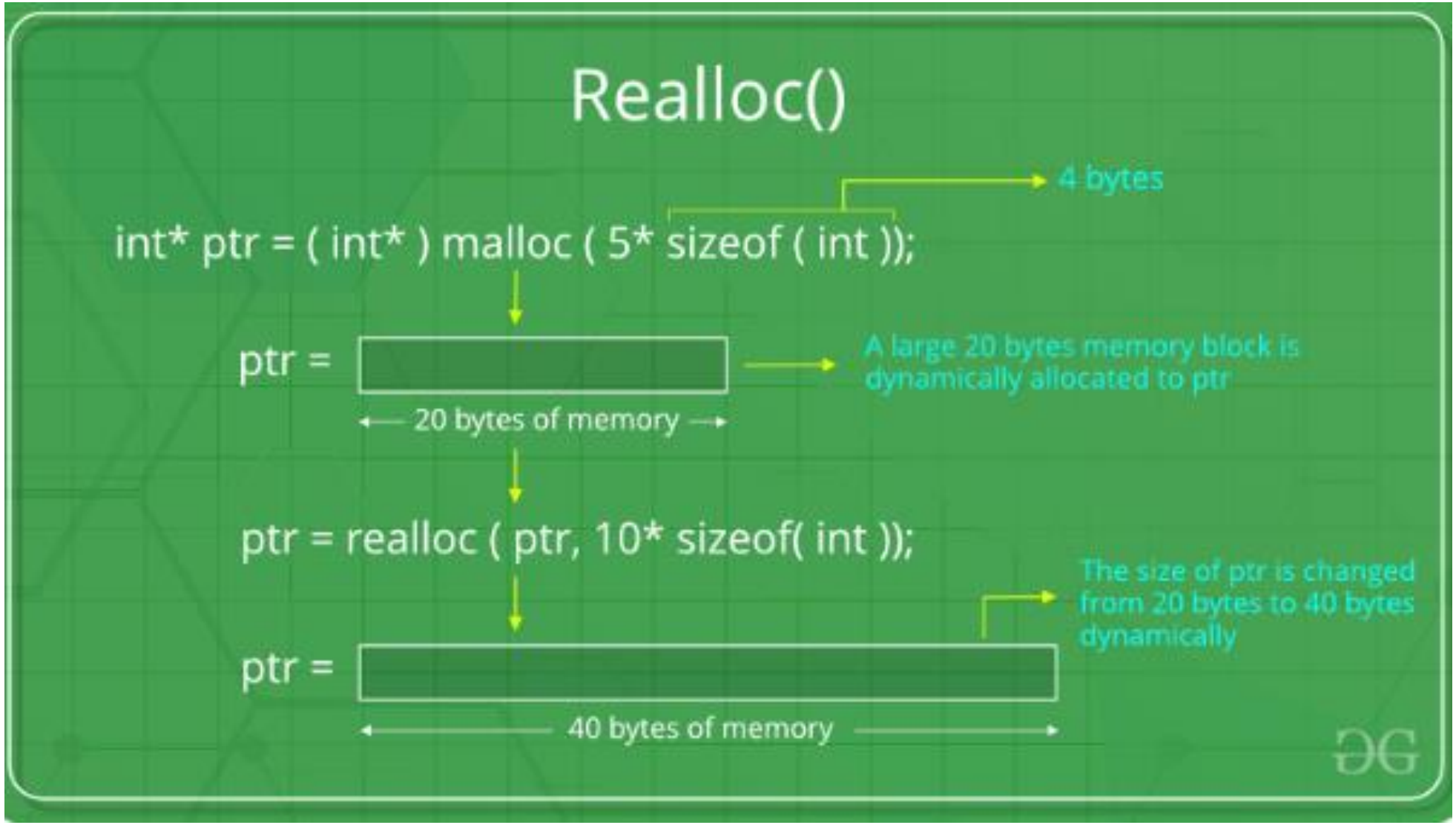
Alocação Dinâmica de Memória

- A Função Realloc;
- Sintaxe

```
ptr = realloc(ptr, newSize);
onde ptr é realocado com o novo tamanho 'newSize'.
```

Se o espaço for insuficiente, a alocação falha e retorna um ponteiro NULL.

Alocação Dinâmica de Memória



Alocação Dinâmica de Memória

```c
//Pega o número de elementos do array
n = 5;
printf("Digite o numero de elementos: %d\n", n);
 // Alocar memória dinamicamente usando calloc()
ptr = (int*)calloc(n, sizeof(int));
// A memória foi alocada com sucesso
printf("Memória alocada com sucesso usando calloc.\n");
// Obtém o novo tamanho do array
n = 10;
printf("\n\nDigite o novo tamanho do array:: %d\n", n);
 // Realoca memória dinamicamente usando realloc()
ptr = realloc(ptr, n * sizeof(int));
 // A memória foi alocada com sucesso
printf("Memória realocada com sucesso usando realloc.\n");
```

Alocação Dinâmica de Memória

- A Função Realloc;

```
Insira o número de elementos: 5
Memória alocada com sucesso usando calloc.
Os elementos da array são: 1, 2, 3, 4, 5,

Insira o novo tamanho da array: 10
Memória realocada com êxito usando realocar.
Os elementos da array são: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
```

Alocação Dinâmica de Memória

- A Função Realloc;

Função	Descrição
Malloc	Aloca o número especificado de bytes;
Realloc	Aumenta ou diminui o tamanho do bloco de memória especificado, movendo-o se necessário;
Calloc	Aloca o número especificado de bytes e os inicializa para zero;
Free	Libera o bloco especificado de memória de volta para o sistema;

FIM!