



# Programação (CK0226 – 2017.2)

Universidade Federal do Ceará  
Departamento de Computação  
Prof. Lincoln Souza Rocha  
([lincoln@dc.ufc.br](mailto:lincoln@dc.ufc.br))

# INTRODUÇÃO À PROGRAMAÇÃO NA LINGUAGEM C



# Alocação Dinâmica de Memória

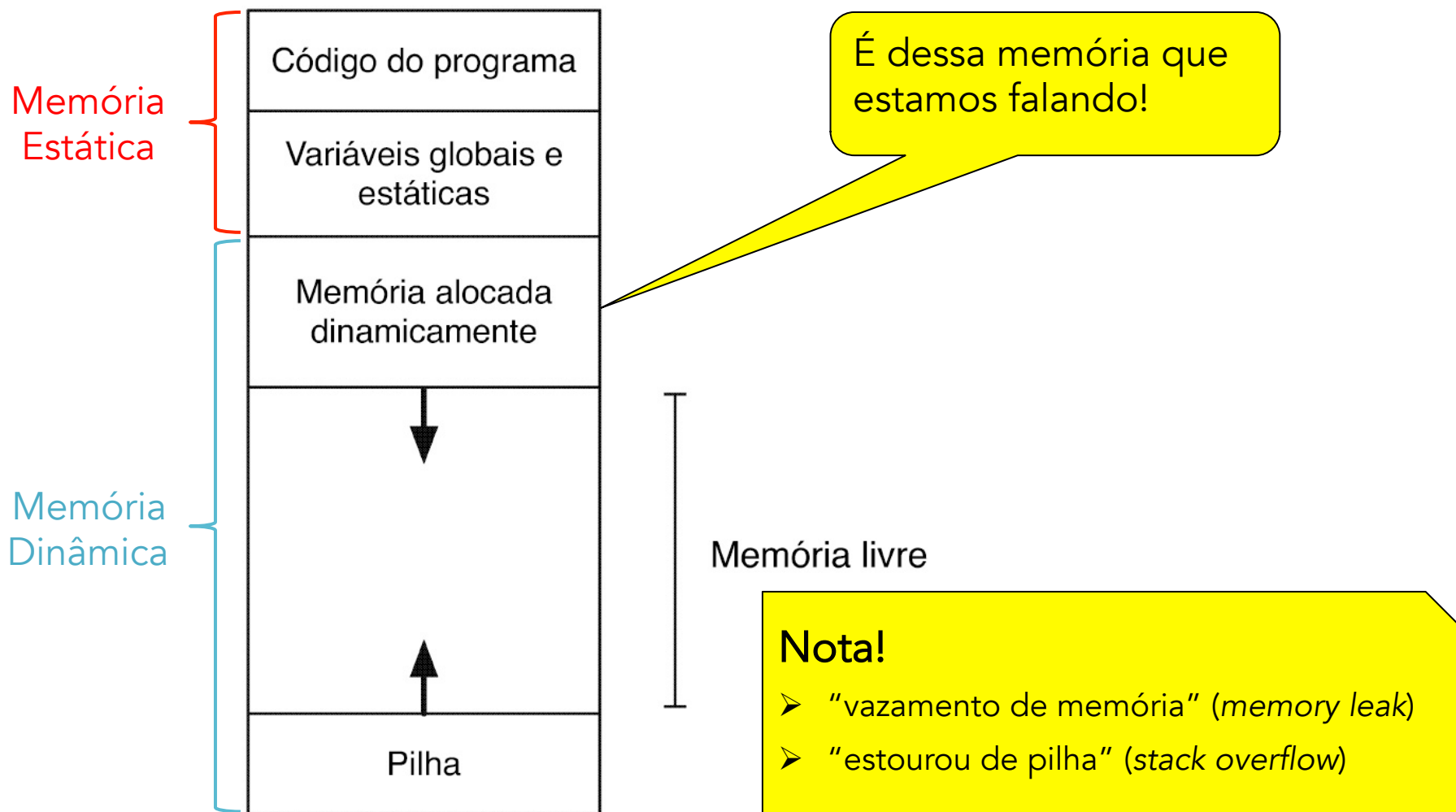


# Sumário

- Representação da Memória
- Alocação/Liberação de Memória



# Representação da Memória



# Alocação Dinâmica

Em C, a manipulação dinâmica de memória é feita por meio de funções da biblioteca padrão "stdlib.h".

int sizeof(<tipo>) - retorna o número de bytes ocupado por um dado tipo informado via argumento da função.

void\* malloc(<#bytes>) - recebe como parâmetro o número de bytes que se deseja alocar e retorna um ponteiro genérico para o endereço inicial da área de memória alocada, se houver espaço livre. Em caso contrário, retorna NULL.

void\* realloc(void\*, <#bytes>); - permite realocar um vetor preservando o conteúdo dos elementos que permanecem válidos após a realocação.

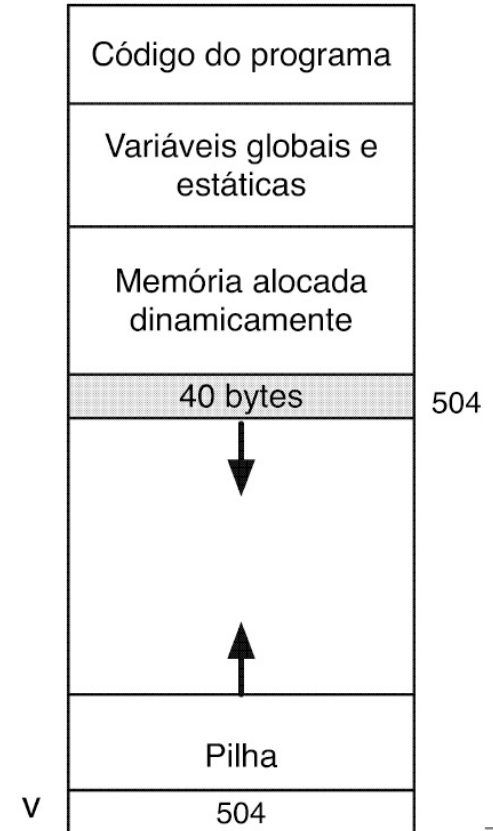
void free(\*void) - recebe como parâmetro um ponteiro para a área de memória que se quer liberar.



# Alocação Dinâmica

Como alocar dinamicamente um vetor de inteiros com 10 posições de memória?

```
int *v = (int *) malloc(10*sizeof (int));
```



# Alocação Dinâmica

Tratando casos em que não existe memória suficiente!

```
(...)  
int *v = (int*) malloc(10*sizeof(int));  
(...)  
if (v==NULL) {  
    printf("Memoria insuficiente.\n");  
    exit(1);  
}  
(...)
```



# Alocação Dinâmica

Liberando a memória no final!

```
(...)  
int *v = (int*) malloc(10*sizeof(int));  
(...)  
if (v==NULL) {  
    printf("Memoria insuficiente.\n");  
    exit(1);  
}  
(...)  
free(v);  
(...)
```

# Vetores: Média e Variância

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

float* captura (int n, float* x);
float media (int n, float* x);
float variancia (int n, float* x, float m);

int main (void ) {
    int n; /* número de valores */
    float *x; /* vetor dos valores */

    printf("Entre com o numero de valores: ");
    scanf("%d", &n);
    x = (float *) malloc(n*sizeof(float));

    if (x == NULL) {
        printf("Memoria insuficiente.\n");
        exit(1);
    }
    captura(n, x);
    float m = media(n, x);
    float v = variancia(n, x, m);
    printf("Media: % f\n Variancia: % f\n", m, v);
    free(x);
    return 0;
}
```

alocação de vetor com n posições

liberação da memória



# Vetores: Média e Variância

```
void captura (int n, float* x) {  
    printf("Entre com os valores:\n");  
    for (int i=0; i<n; ++i)  
        scanf("%f", & x[i]);  
}
```

```
float media (int n, float* x) {  
    float m = 0.0 f;  
    for (int i=0; i<n; ++i)  
        m += x[i];  
    return m / n;  
}
```

```
float variancia (int n, float* x, float m) {  
    float v = 0.0 f;  
    for (int i=0; i<n; ++i)  
        v += (x[i]-m) * (x[i]-m);  
    return v / n;  
}
```





# Programação (CK0226 – 2017.2)

Universidade Federal do Ceará  
Departamento de Computação  
Prof. Lincoln Souza Rocha  
([lincoln@dc.ufc.br](mailto:lincoln@dc.ufc.br))