

Alocação Dinâmica de Memória

- realloc

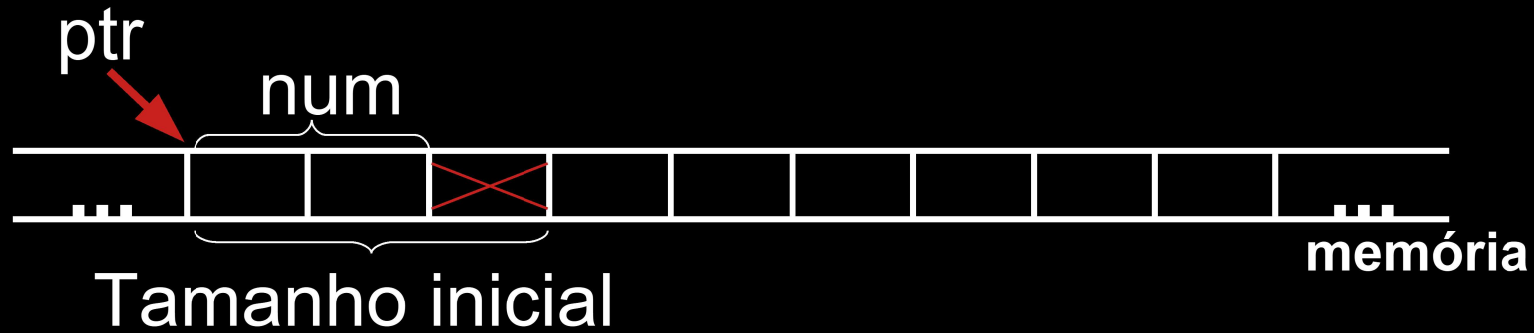
A função **realloc()** serve para realocar memória e tem a seguinte forma:

```
void *realloc (void *ptr, unsigned int num);
```

A função modifica o tamanho da memória previamente alocada apontada por ***ptr** para aquele especificado por **num**.

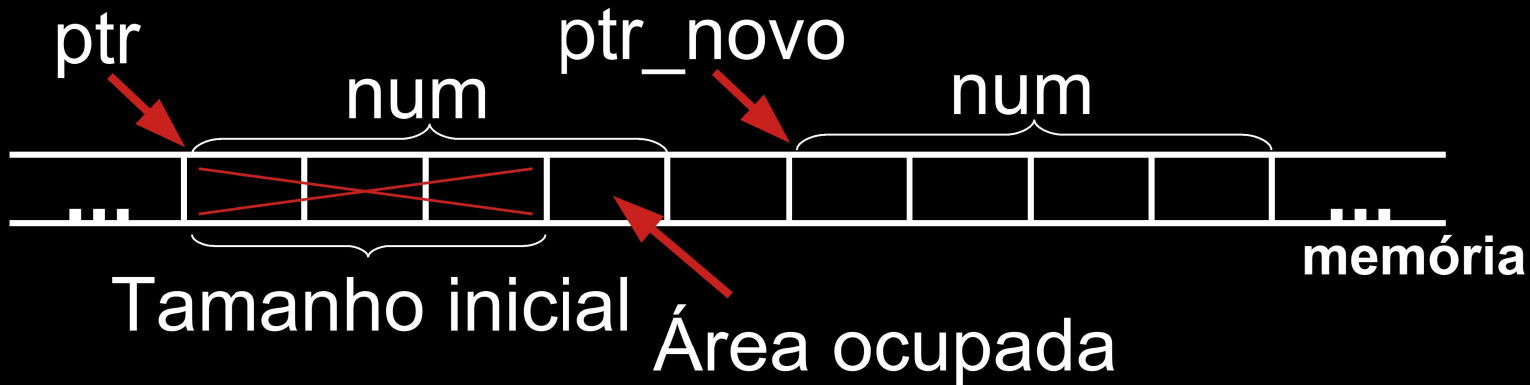
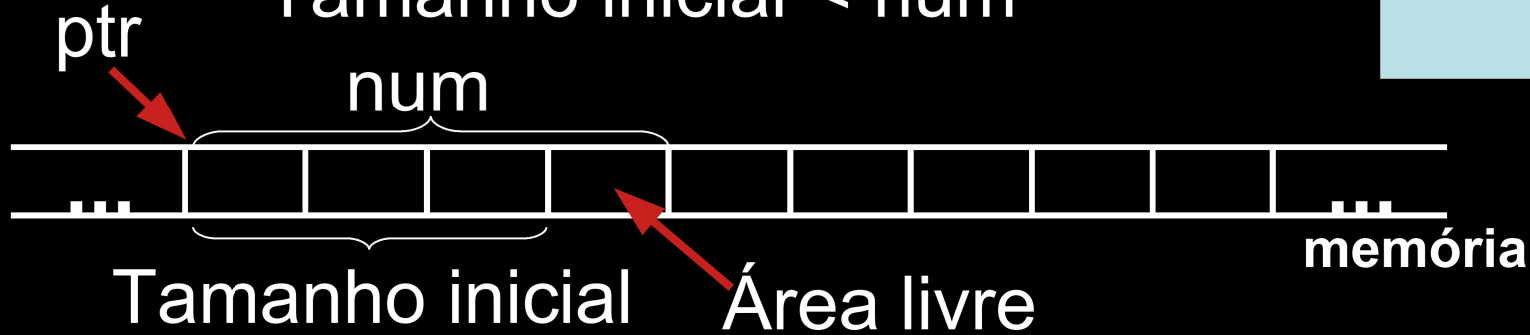
Situações possíveis:

Tamanho inicial > num



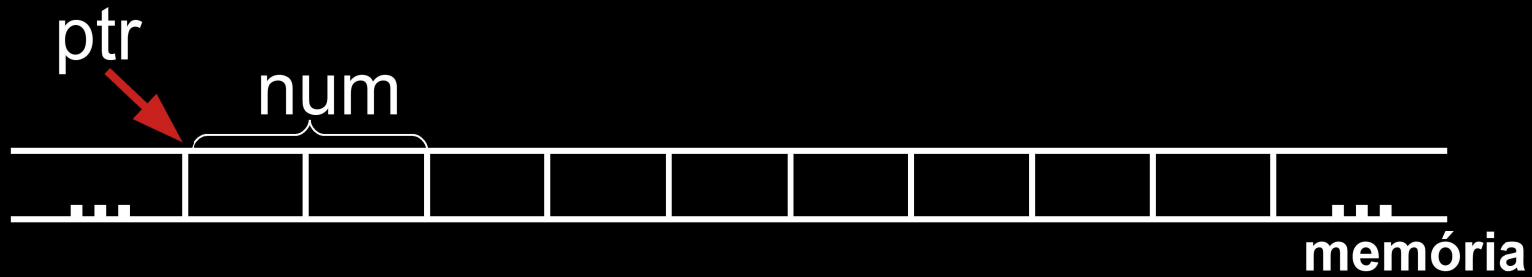
Situações possíveis:

Tamanho inicial < num



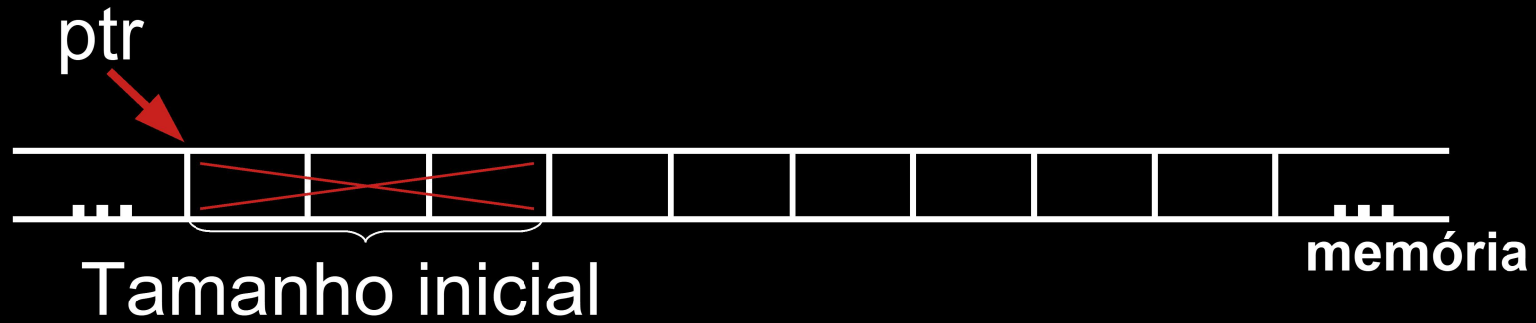
Situações possíveis:

ptr == NULL



Situações possíveis:

num == 0 (zero)



```
#include <stdio.h>
#include <stdlib.h>
main (void) {
    int *p, a, i;
    /* ... */
    a = 30;
    p = (int *) malloc (a*sizeof(int));
    if (!p) {
        printf ("** Erro: Memoria Insuficiente **");
        exit (1);
    }
    for (i=0; i<a ; i++)
        p[i] = i*i;
    a = 100;
    p = (int *)realloc (p, a*sizeof(int));
    if (!p) { printf ("\nERR0!\n"); exit (1); }
    for (i=30; i<a ; i++)
        p[i] = a*i*(i-6);
    ...
}
```

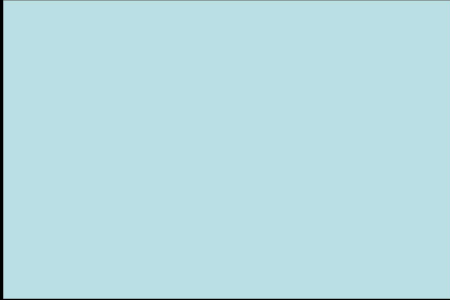
Alocação Dinâmica de Memória

- free

Quando alocamos memória dinamicamente é necessário que a mesma seja liberada quando esta não for mais necessária.

Para isto existe a função **free()** cuja forma é:

```
void free (void *p);
```



```
#include <stdio.h>
#include <stdlib.h>
int main (void)
{
    int *p, a;
    /* ... */
    p=(int *)malloc(a*sizeof(int));
    if (!p) {
        printf ("** Erro: Memoria Insuficiente **");
        exit (1);
    }
    /* ... */
    free(p);
    /* ... */
}
```

Alocação Dinâmica de Memória

Exercício:

Escreva um programa em C que manipule um vetor de inteiros não nulos alocado dinamicamente. O programa recebe inteiros, através da entrada padrão, e os insere no vetor. A cada inteiro que é inserido a área de memória necessária para armazenar um inteiro é incrementada ao número de bytes necessários para armazenar o vetor. O vetor não ocupa memória inicialmente. Quando o usuário entrar com o inteiro 0 (zero), o programa será finalizado e o mesmo não pertencerá ao vetor. Após o processo de inserção o vetor deve ser impresso na saída padrão. Libere a memória utilizada antes do final do processamento.

```
#include <stdio.h>
#include <stdlib.h>
int main () {
    int *vetor=NULL, num_elementos=0, num;
    do {
        printf ("\nEntre com o número inteiro que deseja inserir%s",
            " no vetor (entre com zero para finalizar o programa): ");
        scanf ("%d",&num);
        if (num) {
            num_elementos++;
            vetor =(int*)realloc(vetor,
                num_elementos*sizeof(int));
            if (!vetor) { printf ("\nERRO!\n"); exit (1); }
            vetor[num_elementos-1]=num;
        }
    }while(num);
    printf("\nOs elementos do vetor são:");
    for (;num<num_elementos;num++)
        printf("\n0 %d° elemento do vetor eh %d", num+1, *(vetor+num));
    free(vetor);
}
```

Alocação Dinâmica de Memória

Exercício:

Escreva um programa em C que manipule vetores de inteiros não nulos alocados dinamicamente. O programa recebe inteiros não nulos, através da entrada padrão, e os insere no vetor. A cada inteiro que é inserido a área de memória necessária para armazenar um inteiro é incrementada ao número de bytes necessários para armazenar o vetor. Um vetor não ocupa memória inicialmente. Quando ocorre o fornecimento do inteiro 0 (zero), o programa percebe que o mesmo não pertence ao vetor e que o vetor já teve todos os valores de seus elementos fornecidos.

Alocação Dinâmica de Memória

O programa então apresenta o vetor na saída padrão, com layout adequado, e continua a leitura de valores para outro vetor, procedendo da mesma forma. O fornecimento de um vetor sem nenhum elemento indica que o programa deve finalizar seu processamento. Após a apresentação de cada vetor na saída padrão a memória alocada dinamicamente para armazená-lo deve ser liberada. Obs.: As entradas não devem ser solicitadas ao usuário.

Alocação Dinâmica de Memória

Exemplo de entrada:

8

15

2

7

34

0

3

9

0

0

Exemplo de Saída:

Vetor = { 8, 15, 2, 7, 34 }

Vetor = { 3, 9 }

```
#include <stdio.h>
#include <stdlib.h>
int main () {
    int *vetor, num_elementos, num;
    do {
        vetor=NULL;
        num_elementos=0;
        do {
            scanf ("%d",&num);
            if (num) {
                num_elementos++;
                vetor =(int*)realloc(vetor, num_elementos*sizeof(int));
                if (!vetor) { printf ("\nERRO!\n"); exit (1); }
                vetor[num_elementos-1]=num;
            }
        }while(num);
        if (num_elementos) {
            printf("Vetor = { ");
            for (;num<num_elementos;num++)
                printf(" %d, ", *(vetor+num));
            printf("\b\b }\n");
            free(vetor);
        }
    }while(num_elementos);
    return 0;
}
```