



Alocação Dinâmica - Programação II

Autor Prof.: Delcino Picinin Júnior

Descrição

As declarações abaixo alocam espaço na memória para algumas variáveis. A alocação é estática, ou seja, acontece antes que o programa comece a ser executado.

Código 1: Exemplo por alocação estática

```
1 char c;  
2 int i;  
3 int v[10];
```

Em muitas aplicações, a quantidade de memória a alocar só se torna conhecida durante a execução do programa. Para lidar com essa situação é preciso recorrer à alocação dinâmica de memória. A alocação dinâmica é administrada pelas funções **malloc**, **realloc** e **free**, que estão na biblioteca `stdlib`. Para usar essa biblioteca, inclua a correspondente interface no seu programa.

Função - Malloc

A função `malloc` (o nome é uma abreviatura de *memory allocation*) aloca espaço para um bloco de bytes consecutivos na memória RAM (*random access memory*) do computador e devolve o endereço desse bloco. O número de bytes é especificado no argumento da função.

O exemplo a baixo mostra um caso onde é lido o tamanho que o vetor deve ter, então alocado o vetor, e por fim lido valores para esse vetor.

Código 2: Exemplo de malloc

```
1 #include<stdio.h>
2 #include<stdlib.h>
3
4
5 int main(void){
6
7     int *v;
8     int n;
9     scanf ("%d", &n);
10    v = (int*) malloc (n * sizeof (int ));
11    for (int i = 0; i < n; i++){
12        scanf ("%d", &v[i]);
13    }
14    free (v);
15 }
```

Função - Realloc

Às vezes é necessário alterar, durante a execução do programa, o tamanho de um bloco de bytes que foi alocado por **malloc**. Isso acontece, por exemplo, durante a leitura de um arquivo que se revela maior que o esperado. Nesse caso, podemos recorrer à função **realloc** para redimensionar o bloco de bytes fora da função. Não é uma cópia, mas sim um ponteiro.

Código 3: Exemplo de realloc

```
1 #include<stdio.h>
2 #include<stdlib.h>
3
4
5 int main(void){
6
7     int *v;
8
9     v = (int*) malloc (1000 * sizeof (int ));
10    for (int i = 0; i < 1000; i++){
11        scanf ("%d", &v[i]);
12    }
13    v = (int*) realloc (v, 2000 * sizeof (int ));
14    for (int i = 1000; i < 2000; i++){
15        scanf ("%d", &v[i]);
16    }
17    free (v);
18 }
```
