

## Localização das variáveis de um programa na memória

### Pedro M. Botelho

Como sabemos, o espaço de um programa é separado em segmentos, e, dependendo de como uma variável é inicializada, ou se ela é estática ou não, ela pode ser mandada para um local diferente. O segmento de dados, o **.data**, pode ser dividido em diversas seções, para diferentes usos. Vale lembrar que variáveis **estáticas** são espaços de memória definidos, ou seja, alocados, "permanentemente" na memória, pelo menos enquanto o programa estiver executando. Exemplos de variáveis não-estáticas são as salvas na **pilha**, já que são temporárias, sendo seus espaços de memória apagados e sobrescritos constantemente.

- Variáveis estáticas inicializadas globalmente, ou seja, **variáveis globais**, inicializadas no escopo geral do código, bem como **variáveis locais estáticas inicializadas**, são salvas na seção de dados, o **.data**. Ex: `static int contadorGeral = 0;`
- **Variáveis locais**, ou seja, variáveis declaradas ao longo do programa, são salvas no **segmento de pilha**, a não ser que você especifique, ou o próprio compilador, que ela deva ficar em um registrador ou que ela deva ser estática. Bom exemplo são as variáveis que você vai declarando e inicializando. **Variáveis automáticas** também são guardadas na pilha, como referências a outras variáveis na memória, e a objetos, no heap. Ex: `int i = 1;` ou `SomeClass ref = new SomeClass();` ref sendo uma referência, para o objeto no heap, é salvo na pilha. Os **parâmetros** também são passados por meio da pilha, seja por passagem de valor ou passagem por referência, assim como o endereço de retorno da função e sua moldura.
- Variáveis **estáticas**, e **constantes**, não inicializadas são colocadas no **.bss**. Caso a variável estática global, ou local, seja inicializada com zero o compilador pode colocá-la no **.bss**. Caso a variável não seja estática, mesmo não inicializada, ela será colocada na pilha. Ex: `static const int MAXIMO;`
- Variáveis alocadas de maneira dinâmica são guardadas no **heap**. O heap é muito maior e mais flexível que a pilha, então é usado quando se precisa guardar grandes quantidades. Variáveis alocadas **dinamicamente**, bem como **objetos**, são guardados no heap, e sua referência, variável que irá apontar para os espaços do heap, fica guardada na **pilha**. Ex: `SomeClass ref = new SomeClass();` ref, na pilha, aponta para um objeto da classe `SomeClass`, que fica no heap. `int* array = malloc(1000*sizeof(int));` array é uma referência para um vetor de 1000 posições de `int` guardado no heap.
- Constantes estáticas, sendo **inicializadas**, são colocadas em um local especial, na seção de **apenas-leitura** do segmento de dados, o **.rodata**. Ex: `static const int ZERO = 0;` Constantes não estáticas, sejam elas inicializadas ou não, podem ser colocadas no **.rodata** (se forem globais) ou na pilha (se forem locais).

A escolha de em qual segmento colocar uma variável depende de como aquela variável está sendo usada, de como ela foi inicializada e de quais modificadores foram usados. Tudo isso, e outros quesitos, são levados em conta pelo **compilador**, na hora de alocar as variáveis.