

Programação em C

Variáveis e escopo

Agostinho Brito

2021

O que são variáveis

- Usamos variáveis em notação matemática para associar um símbolo a uma quantidade que pode ser alterada durante um determinado estudo. Ex: $x = 10$.
- As variáveis são espécies de recipientes de memória onde podemos guardar informações para recuperarmos em algum tempo futuro.
- Em programação, uma variável é um local de memória utilizado para guardar um determinado valor numérico ou o resultado de uma expressão. Normalmente é representada por uma sequência de caracteres.
- C é uma linguagem fortemente tipada, de sorte que as variáveis possuem um tipo de dado associado, que normalmente é um valor inteiro ou real, ou alguma combinação destes.
- Linguagens fortemente tipadas permitem um tratamento mais rígido do que é guardado na memória e de como os valores podem ser manipulados.

O que são variáveis

- Usamos variáveis em notação matemática para associar um símbolo a uma quantidade que pode ser alterada durante um determinado estudo. Ex: $x = 10$.
- As variáveis são espécies de recipientes de memória onde podemos guardar informações para recuperarmos em algum tempo futuro.
- Em programação, uma variável é um local de memória utilizado para guardar um determinado valor numérico ou o resultado de uma expressão. Normalmente é representada por uma sequência de caracteres.
- C é uma linguagem fortemente tipada, de sorte que as variáveis possuem um tipo de dado associado, que normalmente é um valor inteiro ou real, ou alguma combinação destes.
- Linguagens fortemente tipadas permitem um tratamento mais rígido do que é guardado na memória e de como os valores podem ser manipulados.

O que são variáveis

- Usamos variáveis em notação matemática para associar um símbolo a uma quantidade que pode ser alteradas durante um determinado estudo. Ex: $x = 10$.
- As variáveis são espécies de recipientes de memória onde podemos guardar informações para recuperarmos em algum tempo futuro.
- Em programação, uma variável é um local de memória utilizado para guardar um determinado valor numérico ou o resultado de uma expressão. Normalmente é representada por uma sequência de caracteres.
- C é uma linguagem **fortemente tipada**, de sorte que as variáveis possuem um tipo de dado associado, que normalmente é um valor inteiro ou real, ou alguma combinação destes.
- Linguagens fortemente tipadas permitem um tratamento mais rígido do que é guardado na memória e de como os valores podem ser manipulados.

- C possui basicamente duas classes de dados primitivos: inteiros e reais, podendo o tipo da variável ser escolhido conforme a faixa de valores numéricos que poderá receber.
- Quanto mais extensa a faixa de dados, mais bytes a variável ocupa na memória.
- Por exemplo, para armazenar valores com idades de pessoas, o tipo de dado apropriado contemplará uma faixa menor que aquela usada para armazenar a quantidade de pessoas em um supermercado.
- Um dos tipos de dados mais comuns em C é `int`. Variáveis desse tipo ocupam **PELO MENOS** 16 bits na memória e são capazes de guardar valores INTEIROS na faixa $[-32.767, +32.767]$.

```
int main(void) {  
    int x;  
}
```

```
#include <stdio.h>
int main(void) {
    int x;
    scanf("%d", &x);
    printf("x = %d", x);
}
```

- Onde criar as variáveis? (Escopo)

Variável LOCAL São visíveis e podem ser alteradas apenas dentro da sua função.

Variável GLOBAL São visíveis e podem ser alteradas por todas as funções do programa.

- Variáveis locais devem ser preferidas em detrimento das globais, para melhor portabilidade de código.

```
#include <stdio.h>
int y; // variavel global
int main(void) {
    int x; // variavel local
    scanf("%d", &x);
    printf("x = %d", x);
}
```

Lendo/escrevendo um dado real

- O tipo de dado `float` é usado para armazenar valores fracionários (reais) e possui 32 bits reservada para sua representação.
- Pode assumir valores na faixa $\pm[1.175494e - 38, 3.402823e + 38]$

```
#include <stdio.h>
int main(void) {
    float x;
    scanf("%f", &x);
    printf("x = %f", x);
}
```


Outros tipos de dados

Tipo	Num. Bits	Formato	Intervalo	
(signed) char	8	%c	-128	127
unsigned char	8	%c	0	255
(signed) short int	16	%hi	-32.768	32.767
unsigned short int	16	%hu	0	65.535
(signed) int	32	%i ou %d	-2.147.483.648	2.147.483.647
unsigned int	32	%u	0	4.294.967.295
(signed) long int	64	%li	-9.223.372.036.854.775.807	9.223.372.036.854.775.807
unsigned long int	64	%lu	0	18.446.744.073.709.551.615
float	32	%f	1.1755e-38	3.4028e+38
double	64	%lf	2.2251e-308	1.7977e+308
long double	80	%Lf	3.3621E-4932	1.1897E+4932

Qual tipo usar?

Comece pelo menor!

```
#include <stdio.h>
int main(void) {
    int x;
    float y;
    x = 48;
    y = -3.2421212234e10;
    printf("x = %d\n", x);
    printf("y = %g\n", y);
}
```

Cuidado com a atribuição de valores fora da faixa

- Os tipos de dados são armazenados de formas bem características na memória.
- Nos tipos sinalizados (signed), por exemplo, os números negativos são armazenados numa forma chamada **complemento de 2**. O bit mais significativo do número é usado para tomar decisões no processo de codificação/decodificação.
- Nos tipos em ponto flutuante, a atribuição de valores fora dos limites de representação do tipo pode causar inconsistências, como a representação de infinitos e NaN (not a number).
- É importante que o programador fique muito atento às faixas e realize testes exaustivos para assegurar que o armazenamento não cause problema ao programa.

- Os tipos de dados são armazenados de formas bem características na memória.
- Nos tipos sinalizados (signed), por exemplo, os números negativos são armazenados numa forma chamada **complemento de 2**. O bit mais significativo do número é usado para tomar decisões no processo de codificação/decodificação.
- Nos tipos em ponto flutuante, a atribuição de valores fora dos limites de representação do tipo pode causar inconsistências, como a representação de infinitos e NaN (not a number).
- É importante que o programador fique muito atento às faixas e realize testes exaustivos para assegurar que o armazenamento não cause problema ao programa.



Praticando o uso de variáveis...

E se não for suficiente?

- A linguagem C foi concebida para contemplar um amplo espectro de aplicações...
- Entretanto, os tipos previstos podem não ser suficientes para coisas mais específicas. Ex: calcular uma grande quantidade de dígitos de π .
- Nesses casos, é preciso usar combinação de recursos de programação para criar tipos de dados compostos que consigam atender aos algoritmos que farão o cálculo desejado.



Obrigado