Programação de Computadores I

Prof. Rafael Nunes

A Linguagem C

Parte 5

Introdução às *Variáveis*

Como podemos nomear as nossas *variáveis* ?

Nomes de Variáveis

- As variáveis no C podem ter qualquer nome se ***duas condições forem satisfeitas***:
  - o nome deve ***começar*** com ***uma letra*** ou ***sublinhado*** (_)
  - os caracteres ***subseqüentes*** devem ser ***letras***, números ou ***sublinhado*** (_)

Nomes de Variáveis

- Há apenas mais duas restrições:
  - o nome de uma variável **não** pode ser igual a uma ***palavra reservada***
  - **nem** igual ao ***nome de uma função*** declarada pelo programador, ou pelas bibliotecas do C.

Nomes de Variáveis

- Variáveis de ***até 32 caracteres*** são aceitas

- É bom sempre lembrar que o C é "case sensitive" e portanto deve-se prestar atenção às ***maiúsculas e minúsculas***.

Os *Tipos* da Linguagem C

Quais são os *5 tipos básicos* da Linguagem C?

Os Tipos da Linguagem C

- O C tem 5 tipos básicos:
  - char
  - int
  - float
  - void
  - double

Destes todos, não vimos ainda o último

“Double”

O double é o *ponto flutuante duplo* e pode ser visto como um ponto flutuante com *muito mais precisão*.

Os Tipos da Linguagem C

- Para cada um dos tipos de variáveis existem os ***modificadores de tipo***.
- Os modificadores de tipo do C ***são quatro***:
  - signed,
  - unsigned,
  - long
  - short.

Os Tipos da Linguagem C

- Ao ***float*** **não** se pode aplicar nenhum
- Ao ***double*** pode-se aplicar ***apenas o long***
- Os ***quatro*** podem ser aplicados a **inteiros**
- A intenção é que ***short*** e ***long*** devam prover ***tamanhos diferentes de inteiros*** onde isto for prático.

Os Tipos da Linguagem C

- O int normalmente terá o tamanho natural para uma *determinada máquina*.
  - Ou seja, numa *máquina de 16 bits*, o int provavelmente terá 16 bits.
  - Numa *máquina de 32*, o int deverá ter 32 bits.

Os Tipos da Linguagem C

- Na verdade, cada compilador é livre para escolher tamanhos adequados para o seu próprio hardware com algumas restrições:
  - Os *shorts* e *ints* devem ocupar pelo menos 16 bits,
  - Os *longs* pelo menos 32 bits,
  - O *short* não pode ser maior que int,
  - O *int* não pode ser maior que long.

A seguir estão listados os tipos de dados permitidos e seu valores *máximos* e *mínimos* em um ***compilador*** típico para um hardware de 16 bits

Os Tipos da Linguagem C

Tipo	Num de bits	Intervalo	
		Inicio	Fim
char	8	-128	127
unsigned char	8	0	255
signed char	8	-128	127
int	16	-32.768	32.767
unsigned int	16	0	65.535
signed int	16	-32.768	32.767
short int	16	-32.768	32.767
unsigned short int	16	0	65.535
signed short int	16	-32.768	32.767
long int	32	-2.147.483.648	2.147.483.647
signed long int	32	-2.147.483.648	2.147.483.647
unsigned long int	32	0	4.294.967.295
float	32	3,4E-38	3.4E+38
double	64	1,7E-308	1,7E+308
long double	80	3,4E-4932	3,4E+4932

Os Tipos da Linguagem C

- O tipo ***long double*** é o tipo de ponto flutuante com maior precisão.
- É importante observar que os intervalos de ponto flutuante, na tabela anterior, estão indicados em ***faixa de expoente***...
- ... mas os números podem assumir valores tanto ***positivos*** quanto ***negativos***.

Declaração e Inicialização de *Variáveis*

Declaração e Inicialização de Variáveis

- As variáveis no C devem ser declaradas antes de serem usadas

A forma geral da declaração de variáveis é:

tipo_da_variável lista_de_variáveis;

Declaração e Inicialização de Variáveis

- As variáveis da lista de variáveis terão todas o mesmo tipo e deverão ser separadas por vírgula.
- Como o ***tipo default do C*** é o **int**, quando vamos declarar variáveis int com algum dos modificadores de tipo, basta colocar o nome do modificador de tipo.
  - Assim um ***long*** sozinho será equivalente a um ***long int***

- Alguns exemplos:

```
char ch, letra;
long count;   //long int count
float pi;
```

Declaração e Inicialização de Variáveis

- Há três lugares nos quais podemos declarar variáveis.
- O primeiro é **fora de todas as funções** do programa.
  - Estas variáveis são chamadas ***variáveis globais*** e podem ser usadas a partir de ***qualquer lugar no programa***.
  - Pode-se dizer que, como elas estão fora de todas as funções, ***todas as funções as vêem***.

Declaração e Inicialização de Variáveis

- O segundo lugar no qual se pode declarar variáveis é no **início de um bloco de código**.
- Estas variáveis são chamadas ***locais***
- Elas ***só têm validade dentro do bloco*** no qual são declaradas
- ... isto é, ***só a função à qual ela pertence*** sabe da existência desta variável, dentro do bloco no qual foram declaradas.

Declaração e Inicialização de Variáveis

- O terceiro lugar onde se pode declarar variáveis é na **lista de parâmetros de uma função**.

- Mais uma vez, apesar de estas variáveis receberem ***valores externos***, estas variáveis ***são conhecidas apenas pela função*** onde são declaradas.

Veja o *exemplo* a seguir...

```
1.  #include <stdio.h>
2.  int contador;
3.  int func1(int j) {
4.          ...
5.  }
6.  int main()
7.  {
8.          char condicao;
9.          int i;
10.         for (i=0; ...)
11.         {               /* Bloco do for */
12.             float f2;
13.                 ...
14.             func1(i);
15.         }
16.                 ...
17.         return(0);
18.}
```

Vamos entender...

Declaração e Inicialização de Variáveis

- A variável ***contador*** e uma variável global, e é ***acessível de qualquer parte*** do programa.
- As variáveis ***condição*** e ***i***, só existem dentro de main(), isto é ***são variáveis locais*** de main.
- A variável float ***f2*** é um exemplo de uma ***variável de bloco***, isto é, ela somente é conhecida dentro do bloco do for, pertencente à função main.
- A variável inteira ***j*** é um exemplo de ***declaração na lista de parâmetros*** de uma função (a função func1).

Declaração e Inicialização de Variáveis

- As regras que regem onde uma variável é válida chamam-se ***regras de escopo*** da variável.
- Há mais dois detalhes que devem ser ressaltados.
  - Duas ***variáveis globais*** **não** podem ***ter o mesmo nome***.
  - O mesmo vale para duas variáveis locais de uma mesma função.
  - Já duas ***variáveis locais***, de funções diferentes, **podem** ***ter o mesmo nome*** sem perigo algum de conflito.

Declaração e Inicialização de Variáveis

- Podemos ***inicializar variáveis no momento de sua declaração***.

- Para fazer isto podemos usar a forma geral

tipo_da_variável nome_da_variável = constante;

Declaração e Inicialização de Variáveis

- Isto é ***importante*** pois quando o C cria uma variável ***ele não a inicializa***.
- Isto significa que até que um primeiro valor seja atribuído à nova variável ela tem um ***valor indefinido*** e que **não pode** ***ser utilizado para nada***.
- Nunca presuma que uma variável declarada ***vale zero*** ou ***qualquer outro valor***.

Declaração e Inicialização de Variáveis

- Exemplos de inicialização:

```
char ch='D';
int count=0;
float pi=3.141;
```

Exercício

Exercício

- Escreva um programa que declare uma variável inteira global e atribua o valor 10 a ela.
- Declare outras 5 variáveis inteiras locais ao programa principal e atribua os valores 20, 30, ..., 60 a elas.
- Declare 6 variáveis caracteres e atribua a elas as letras c, o, e, l, h, o .
- Finalmente, o programa deverá imprimir:
  - As variáveis inteiras contem os números: 10,20,30,40,50,60
  - O animal contido nas variáveis caracteres é o: coelho

Até a próxima...