



Printf e Scanf - Programação II

Autor Prof.: Delcino Picinin Júnior

Descrição

A linguagem C define duas funções que podem ser usadas para escrever e ler valores.

Os valores são normalmente lidos do teclado e escritos no monitor.

Você precisa incluir a biblioteca «**stdio.h**» para poder usar o **printf** para escrita e **scanf** para leitura.

Função printf

O primeiro argumento de é um string entre aspas. Esta string, que sempre deve estar presente, pode especificar através de caracteres especiais (%) quantos outros argumentos estarão presentes nesta invocação da função. Estes outros argumentos serão variáveis cujos valores serão formatados e apresentados na tela.

Código 1: Exemplo Printf

```
1 #include <stdio.h>
2
3 void main(){
4     int v1;
5
6     v1=10;
7
8     printf ("\n O valor da variável é %d ",v1);
9
10 }
```

A tabela abaixo apresenta os tipos de argumentos.

Símbolo	Descrição
d,i	Inteiros
x,X	Hexadecimal
c	Caracter
f	Float ou Double
s	String (vetor de caracteres)

Toda a variável tem um conteúdo e um endereço, você pode escrever o conteúdo e o endereço de uma variável com o **printf**. Para escrever o endereço, use o símbolo **&** antes do nome da variável. Normalmente para escrever endereços, adota-se o formatador hexadecimal (X).

Código 2: Exemplo printf de endereço

```

1 #include <stdio.h>
2
3 void main(){
4     int v1;
5
6     v1=10;
7
8     printf ("\n O valor da variável é %d ",v1); //conteúdo
9     printf ("\n O endereço da variável é %X",&v1); //endereço
10
11 }
```

Uma variável pode ser um ponteiro, nesse caso seu conteúdo é o endereço de outra variável.

Na declaração de ponteiros, o asterisco (*) antecede o nome da variável. Vetores estáticos embora não possuam asterisco no nome, também são ponteiros.

Como o conteúdo de um ponteiro é um endereço, adota-se o formatador (X) para hexadecimal.

É possível usar o **printf** para escrever o conteúdo para onde o ponteiro aponta, neste caso use o símbolo asterisco (*) antecede o nome do ponteiro.

Código 3: Exemplo printf de endereço e valor de ponteiro

```
1 #include <stdio.h>
2
3 void main(){
4     int v1;
5     int *v2; //ponteiro para inteiro
6
7     v1=10;
8
9     v2=&v1; // v2 recebe endereço de v1
10
11     printf ("\n O valor da variável v1 é %d ",v1); //conteúdo de v1
12     printf ("\n O endereço da variável v1 é %X ",&v1); //endereço de v1
13
14     printf ("\n O valor da variável v2 é %X ",v2); //conteúdo de V2, que é o endereço de v1
15
16     printf ("\n O endereço da variável v2 é %X ",&v2); //endereço de v2
17
18     printf ("\n O valor para onde o endereço guardado em v2 aponta é %d ",*v2);
19     //valor para onde endereço em v2 aponta
20
21 }
```

Função scanf

O tipo de leitura é especificado através de caracteres especiais (%). Não pode ser colocado texto ou espaço em branco antes ou após o formatador (%), exceto separando vários formatadores. A inclusão de espaços ou texto não afetará a compilação, mas sim o funcionamento do programa. Os formatadores são os mesmos do **printf**, já apresentados.

Para o **scanf** realizar a leitura, ele altera o conteúdo da variável, então ele precisa saber o endereço dessa variável. Somente com o endereço é possível alterar o conteúdo. Esse é o motivo de passarmos o símbolo **&** antes do nome da variável.

Código 4: Exemplo de scanf

```
1 #include <stdio.h>
2
3 void main(){
4     int val1, val2;
5
6     printf ("\n Digite dois valores :");
7     scanf ("%d %d",&val1,&val2);
8
9     printf ("\n O valor da variável val1 é %d ",val1); //conteúdo de val1
10    printf ("\n O valor da variável v2 é %d ",val2); //conteúdo de val2
11 }
```

Os endereços são atribuídos para as variáveis no momento que ocorre a declaração. O **scanf** não lê endereço, ele recebe um endereço e irá ver um valor e atribuir na variável para onde esse endereço aponta.

Se você passar um ponteiro como argumento do **scanf**, não use o símbolo **&**, pois o **scanf** não irá atribuir um valor para esse ponteiro, mas sim para o local onde o conteúdo desse ponteiro (que é um endereço) aponta.

Código 5: Exemplo de scanf com ponteiro

```
1 #include <stdio.h>
2
3 void main(){
4     int val1, val2;
5     int *ponteiro1;
6
7     ponteiro1=&val1;
8
9     printf("\n Digite dois valores :");
10    scanf("%d %d",ponteiro1,&val2);
11
12    printf("\n O valor da variável val1 é %d ",val1); //conteúdo de val1
13
14    printf("\n O valor da variável v2 é %d ",val2); //conteúdo de val2
15
16    printf("\n O valor da variável val1 é %d ",*ponteiro1);
17    //conteúdo de val1
18 }
```

Figura 1: Ponteiro

