

**MAC 2166 – Introdução à Computação**

POLI - PRIMEIRO SEMESTRE DE 2007

Material Didático

Prof. Ronaldo Fumio Hashimoto

**ESQUELETO DE UM PROGRAMA EM C**

Todo programa em C a ser desenvolvido nesse curso, deve possuir o seguinte esqueleto:

gcc	Dev-C++
<pre> 1 # include &lt;stdio.h&gt; 2 3 int main() { 4  /* declaração de variáveis */ 5 6  /* lista de comandos */ 7 8 9  return 0; 10 }</pre>	<pre> 1 # include &lt;stdio.h&gt; 2 # include &lt;stdlib.h&gt; 3 int main() { 4  /* declaração de variáveis */ 5 6  /* lista de comandos */ 7 8  system ("pause"); 9  return 0; 10 }</pre>

**Descrição do Esqueleto**

Vamos por enquanto considerar estes esqueletos como uma “receita de bolo”, ou seja, todo programa em C deve conter os comandos das linhas 1, 2, 3, 9 and 10. Note as pequenas diferenças entre os programas para os compiladores gcc e Dev-C++ nas linhas 2 e 8.

Para facilitar, vamos considerar neste curso o esqueleto para o compilador gcc. Assim, se você usar o compilador Dev-C++, você deve acrescentar os comandos das linhas 2 e 8. Com exceção destas duas linhas, os programas em C para os compiladores gcc e Dev-C++ são idênticos.

**Exemplo de um Programa**

Para entendermos melhor, considere o programa em C apresentado na Fig. 1 (para o compilador gcc).

**Comentários**

Primeiramente, os textos entre os símbolos */\** e *\*/* (linhas 5, 7, 11 e 18) são *comentários*. Comentários não interferem no programa, mas auxiliam os programadores a entender e documentar o código.

---

```

1 # include <stdio.h>
2
3 int main() {
4
5     /* Primeiro programa em C */
6
7     /* declarações: todas as variáveis utilizadas precisam ser declaradas */
8
9     int idade;
10
11    /* início do programa */
12
13    printf ("Quantos anos voce tem?: ");
14    scanf ("%d", &idade);
15
16    printf ("%d? Puxa, voce parece que tem so %d anos!\n", idade, idade * 2);
17
18    /* fim do programa */
19
20    return 0;
21 }

```

---

Figura 1: Primeiro Programa

## Declaração de Variáveis

Na linha 9, temos uma declaração de uma variável de nome `idade`. Esta variável guarda números de tipo `int` (inteiro). Todos os programas que iremos desenvolver precisam de variáveis, ou seja, instâncias de armazenamento de dados que nos permitem resolver o problema. Todas as variáveis utilizadas precisam ser declaradas no início do programa. A forma de declaração de variáveis é:

```
int <nome_da_variavel>;
```

## Funções de Leitura e Impressão

Todos os programas devem comunicar-se com o usuário através de funções de impressão (na tela) e de leitura (pelo teclado). Basicamente, nos nossos programas, o usuário fornece números inteiros para o programa através da leitura pelo teclado (função `scanf`); enquanto que o programa fornece ao usuário os resultados via impressão de mensagens na tela (função `printf`). No nosso exemplo, a função de impressão na tela está sendo utilizada nas linhas 13 e 16; enquanto que a função de leitura pelo teclado está sendo utilizada na linha 14.

## Função de Impressão na Tela

Basicamente, a função `printf` imprime todos os caracteres que estão entre aspas. Assim, o `printf` da linha 13 imprime a mensagem (sem aspas) "Quantos anos voce tem?: ". Note o espaço em branco no final da mensagem que também é impresso!

Agora observe o `printf` da linha 16. Este `printf` tem duas diferenças com relação ao `printf` da linha 13. A primeira diferença é que dentro da mensagem do `printf` da linha 16 (caracteres que estão entre aspas) podemos encontrar duas seqüências de caracteres: "%d" e "\n". Além disso, depois da mensagem, temos duas expressões aritméticas envolvendo a variável `idade` separadas por vírgulas: (a) "`idade`" (seria como a expressão aritmética "`idade * 1`"); and (b) a expressão aritmética "`idade * 2`".

O `printf` da linha 16 imprime na tela todos os caracteres que estão entre aspas, com exceção da seqüência de caracteres "%d" e "\n".

Para cada seqüência de caracteres "%d", a função `printf` imprime na tela um número inteiro que é resultado das expressões aritméticas contidas no `printf` separadas por vírgula. Assim, o primeiro "%d" imprime na tela o conteúdo da variável "idade" e segundo "%d" imprime na tela o resultado da expressão "idade \* 2" (uma vez que a expressão "idade" vem antes da expressão "idade \* 2" no `printf` da linha 16).

A seqüência de caracteres "\n", indica à função `printf` para "pular de linha", isto é, faz com que o cursor da tela vá para a próxima linha. No `printf` da linha 16, como a seqüência está no final da mensagem, isto significa que depois de imprimir a mesma na tela, o cursor irá para a próxima linha.

### Função de Leitura pelo Teclado

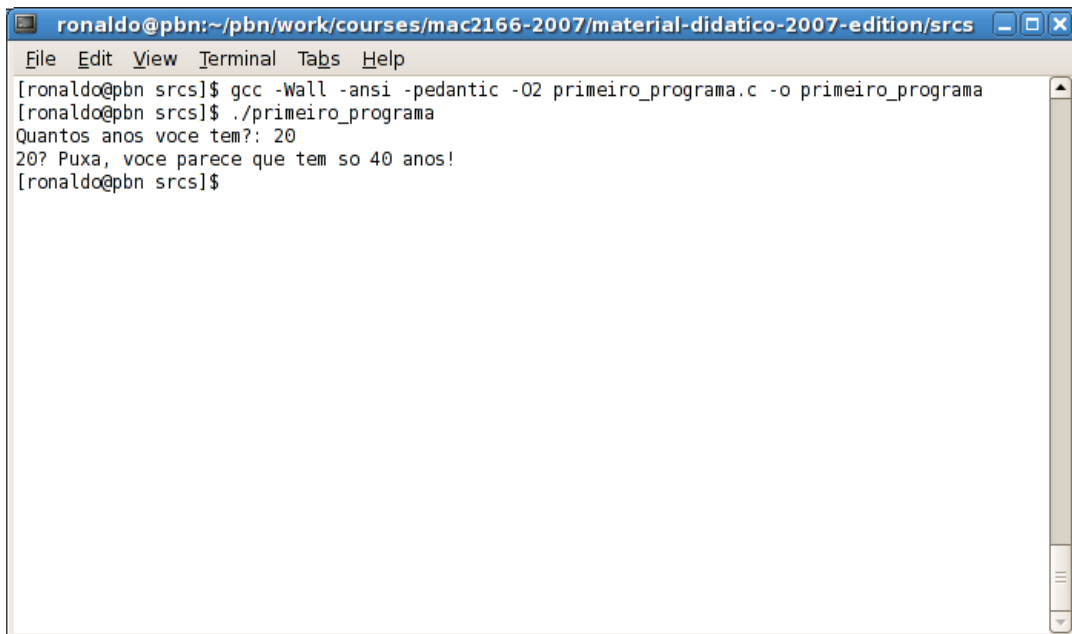
Para ler um número inteiro pelo teclado, você deve usar a função `scanf` da seguinte forma:

```
scanf ("%d", &<nome_da_variavel>);
```

O `scanf` irá esperar o usuário digitar um número inteiro pelo teclado e, após o usuário digitar a tecla <ENTER>, armazenará o número digitado na variável <nome\_da\_variavel>. Um exemplo está na linha 14 do primeiro programa: o número digitado irá ser armazenado na variável `idade`. Observe que no `scanf` deve-se colocar o caractere "&" antes do nome da variável.

### Retornando ao nosso Exemplo

Executando o programa, temos:



Observe que:

1. O número "20" que aparece depois da mensagem

```
"Quantos anos voce tem?: "
```

foi digitado pelo usuário e lido pela função `scanf`.

2. Este número "20" aparece ao lado da mensagem, pois o `printf` que a imprime na tela não tem a seqüência de caracteres `\n` no final; caso contrário, o número "20" seria digitado na próxima linha.
3. Uma vez que o usuário, depois de digitar o número 20, deve dar um <ENTER>, o cursor automaticamente irá para a próxima linha; observe que a mensagem

```
"20? Puxa voce parece que tem so 40 anos!"
```

aparece na próxima linha.

4. Os números "20" e "40" (resultados das expressões aritméticas "idade" e "idade \* 2") são colocados no lugar do `%d` do segundo `printf` do programa.

### **Impressão de `%d` e `\n`**

Para imprimir na tela a seqüência de caracteres "`%d`", você deve usar

```
printf ("%d");
```

e para imprimir "`\n`", você de usar

```
printf ("\n");
```

### **Dúvidas**

Dúvidas deste material pode ser enviadas para o “Fórum para assuntos específicos da turma WEB”.