

Programas e Operadores

Prof. Angelo Loula
UEFS

1

Programando em C

- Estrutura básica de um programa:

```
int main() {  
    instruções  
}
```

- Exemplo:

```
#include <stdio.h>  
int main() {  
    printf("Este é um programa em C\n");  
}
```

2

Programando em C

- Observações:
 - Palavras reservadas: palavras que tem significado especial previamente definido, não podendo ser usadas como identificador de variáveis. Ex: scanf, printf, main, etc;
 - Comentários: texto explicativo entre /* e */, para auxiliar no entendimento dos diversos blocos do programa;
 - Recuo: artifício de alinhamento das instruções para auxiliar no entendimento da lógica do programa;
 - Toda instrução deve terminar com ; (ponto e vírgula).
 - O C diferencia maiúscula de minúsculas
 - Evite escrever só com maiúsculas.

3

Programando em C

- Regras práticas para programas legíveis:
 - Usar comentários e documentar código;
 - Dar nomes significativos às variáveis;
 - Colocar parênteses nas expressões para que a ordem das operações fique clara;
 - Usar recuo para separar visualmente os blocos do programa;
 - Alinhar informações relacionadas;
 - Evitar longas linhas de programação, quebrando-as quando necessário;

4

Programando em C

```
#include <stdio.h>

int main(){
    int num1,num2,soma;

    num1 = 1;
    num2 = 2;

    /* Realiza a soma */
    soma = num1 + num2;

    /* Exibe resultado */
    printf("A soma é
%d",soma);
}
```

OU

```
#include <stdio.h>
int main(){
int A,B,C;
A=1;B=2;C=A+B;
printf("%d",C);
}
```

5

Elaboração de Algoritmos

```
media=((Prova1*2+Prova2*3+Prova3*5)/10+(Trab1+Trab2+Trab3
+Trab4)/10)/2;
```

OU

```
media = (
    ( (Prova1 * 2) + (Prova2 * 3) + (Prova3 * 5) ) /10
+ ( Trab1 + Trab2 + Trab3 + Trab4 ) / 10
) /2;
```

6

Conectivos Lógicos

- Símbolos da lógica proposicional
- Dois valores: verdadeiro ou falso
- Operadores Lógicos: **E** , **OU**, **NÃO**

E	OU	NÃO
f E f = f	f OU f = f	NÃO f = v
v E f = f	v OU f = v	NÃO v = f
f E v = f	f OU v = v	
v E v = v	v OU v = v	

7

Operadores

- Em C:
 - 0 (falso) vs. 1(verdadeiro)
 - Operadores Lógicos
 - ! (NÃO) , || (OU) , && (E)
 - Operadores relacionais
 - == , != , > , < , >= , <=
 - resultado booleano
 - (10 > 5) , (5 == 5) , (10!=5) → verdade
 - (10 <= 5) , (5 != 5) , (10 == 5) → falso

8

Operadores

- Em C:
 - Operadores Aritméticos
 - Unários: - (valor negativo)
 - Binários: +, -, *, /, %
 - 10 + 3 = 13
 - 10 - 3 = 7
 - 10 * 3 = 30
 - 10 / 3 = 3,3333...
 - 10 % 3 = 1 (resto da divisão inteira)

9

Operadores

- Em C, precedência:

primeiro	!	(negação),
	*	/ % (multiplicativos)
	+	- (aditivos)
	<	> <= >= (inequalidade)
	==	!= (igualdade)
	&&	(E lógico)
último		(OU lógico)

10

Primeiros Programas

```
#include <stdio.h>
int main(){
    printf("Oi, mundo!\n");
}
```

```
#include <stdio.h>
int main(){
    printf("%d",21);
}
```

11

Primeiros Programas

#include <nome biblioteca>

- Avisa ao compilador que serão usados procedimentos, funções, macros, variáveis ou constantes declarados no arquivo especificado (no exemplo anterior, stdio.h)
- Permite a reutilização de código
- Outras bibliotecas de uso comum em C
 - #include <math.h> /* Funções matemáticas */
 - #include <stdlib.h> /* Funções de gerência de memória */
 - #include <string.h> /* Funções de manipulação de strings */

12

Primeiros Programas

`printf(formato,...)`

- Função de saída formatada em tela da biblioteca `stdio.h`
- É exibido o conjunto de caracteres definido em `formato` com exceção dos especificadores de formato. No lugar dos especificadores de formatação, são usados os argumentos posteriores.

`%d` decimal
`%e` notação científica
`%f` ponto flutuante
`%u` decimal sem sinal
`%c` caractere simples
`%s` cadeia de caracteres
`%o` octal
`%x` hexadecimal

13

Primeiros Programas

`printf`

Caracteres de Escape

`\a` Caracter de alerta
`\b` Retrocesso
`\f` Alimentação de formulário
`\n` Nova linha
`\r` Retorno de carro
`\t` Tabulação horizontal
`\v` Tabulação vertical
`\\` Contra-barras
`\?` Ponto de interrogação
`\'` Aspas simples
`\"` Aspas duplas
`\ooo` Número octal
`\xhh` Número hexadecimal

14

Primeiros Programas

```
printf("%d",26);

printf("%f",123.4);

printf("%e",123.4); /*1.234e+2*/

printf("26 em hexa: %x",26); /*1A*/

printf("A primeira letra é %c",'A');

printf("Disciplina: %s","EXA841");
```

15

Primeiros Programas

```
#include <stdio.h>
int main(){
    printf("%d", 7 + 2 );
    printf("%d", 7 - 2 );
    printf("%d", 7 * 2 );
    printf("%d", 7 % 2 );
}

#include <stdio.h>
int main(){
    printf("%d", 1 && 0 );
    printf("%d", 1 || 0 );
    printf("%d", ! 1 );
}
```

16

Primeiros Programas

```
#include <stdio.h>
int main(){
    printf("%d", (5 + 7) / (9 - 3) + 1
    );
}
#include <stdio.h>
int main(){
    printf("%d", (1 && 0) or 1 );
}
```

17

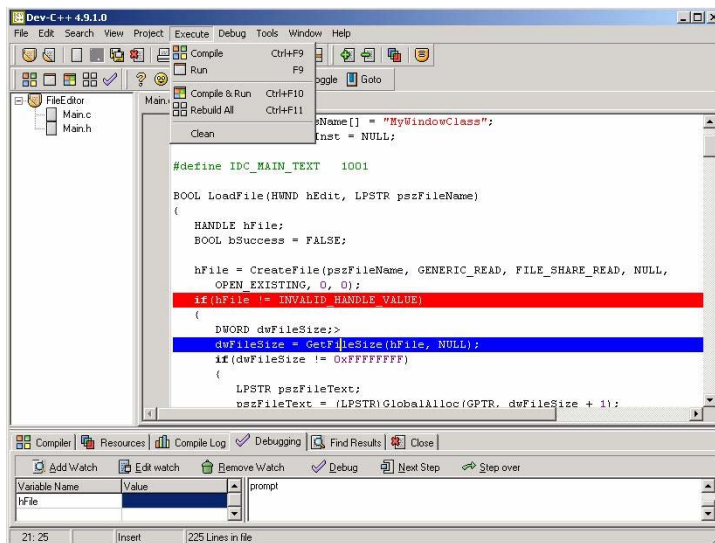
Primeiros Programas

```
#include <stdio.h>
int main(){
    printf("7 + 2");
    printf("%d", 7 + 2 );
}
```

18

DevC++

<http://www.bloodshed.net/devcpp.html>



19