



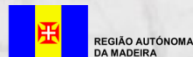
ESCOLA PROFISSIONAL CRISTÓVÃO COLOMBO

www.epcc.pt

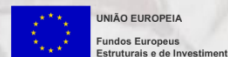


Sistemas Operativos – Processamento Computacional

Operadores e Expressões



REGIÃO AUTÓNOMA
DA MADEIRA



Autor da apresentação

09/09/2014

Expressões

Uma expressão é uma sequência de operadores e operandos, possivelmente contendo também parêntesis, que denotam um valor. Os operandos podem ser constantes, variáveis (incluindo *array's*) e chamadas de funções.

Operadores aritméticos

Operador	significado
+	Adição
-	Subtração
*	Multiplicação
/	Divisão
%	Resto da divisão inteira

O operador `/` aplicado a inteiros ou caracteres o resultado será a divisão inteira, truncando qualquer parte fraccionária.

Operadores aritméticos - Exemplo das divisões

```
int x,y = 0;
```

```
x = 12;
```

```
y = 3;
```

```
printf("%d \n", x/y); /*4*/
```

```
printf("%d \n", x%y); //1
```

```
x = 1;
```

```
y = 3;
```

```
printf("(x/y)=%d , (x%%y)=%d", x/y, x%y); // 0 1
```

```
float f = 1.0e-5;
```

```
printf("float->%f",f);
```

Operadores aritméticos - Incremento

Maior prioridade

Menor prioridade

Operador	
++	--
*	/ %
+	-

```
int soma,x,j;  
soma =x = j = 0;
```

```
soma = ++x - j; /* soma = (x + 1) - 0 */  
printf(" %d \n",soma);  
printf(" %d \n",x);
```

1
1

Operadores relacionais (Return 0-False;1-True)

Operador	significado
>	maior
<	menor
>=	maior ou igual
<=	menor ou igual
==	Igual
!=	diferente

Os operadores relacionais, tal como o nome indica, têm como objectivo obter a relação que os operandos podem ter entre si.

Operadores relacionais -Exemplo

```
int i,j;
printf("Introduza dois número\n");
scanf("%d %d",&i,&j);
system("pause");
printf(" %d == %d é %d\n",i,j,i==j);
printf(" %d != %d é %d\n",i,j,i!=j);
printf(" %d <= %d é %d\n",i,j,i<=j);
printf(" %d >= %d é %d\n",i,j,i>=j);
printf(" %d > %d é %d\n",i,j,i>j);
printf(" %d < %d é %d\n",i,j,i<j);
```


Precedência entre Operadores relacionais e Aritméticos

Os operadores relacionais têm prioridade inferior aos aritméticos.

Por exemplo:

Operação : $i < x - 1$

Será processado como $i < (x - 1)$

Precedência entre Operadores relacionais e Aritméticos -Exemplo

```
#include <stdio.h>
#include <stdlib.h>
#include <locale.h> /* biblioteca de idiomas*/

int main(){
    //configurar idioma para pt-pt
    setlocale(LC_ALL, "Portuguese");

    int i,x;
    printf(" %d < %d - 1 é %d\n",i,x,i < x - 1);
    printf(" %d < (%d - 1) é %d\n",i,x,i < (x - 1));
    printf(" (%d < %d) - 1 é %d\n",i,x,(i < x) - 1);
}
```



Desafio – Codifique em linguagem c um programa para a saída seguinte.

```
Introduza dois números
```

```
10 20
```

```
Números digitados: 10 20
```

```
Press any key to continue . . .
```

```
10 + 20 é 30
```

```
10 - 20 é -10
```

```
10 * 20 é 200
```

```
10 / 20 é 0,50
```

```
-----  
Process exited after 8.76 seconds with return value 0
```

```
Press any key to continue . . .
```

Desafio - Solução

```
int i,j;  
printf("Introduza dois números\n");  
scanf("%d %d",&i,&j);  
printf("Números digitados: %d %d\n",i,j);  
system("pause");  
printf(" %d + %d é %d\n",i,j,i+j);  
printf(" %d - %d é %d\n",i,j,i-j);  
printf(" %d * %d é %d\n",i,j,i*j);  
printf(" %d / %d é %0.2f\n",i,j,(float)i/j);
```

Operadores Lógicos

Operador	significado
&&	E lógico
	OU lógico
!	Negação

Os operadores lógicos são usados para efectuar as operações lógicas elementares: E lógico, OU lógico e Negação.

Operadores Lógicos -Prioridade

	Operador
↑	!
	> >= < <=
	== !=
	&&

Diagram illustrating the priority of logical operators in C. An upward arrow on the left indicates increasing priority, from 'Menor prioridade' (Lower priority) at the bottom to 'Maior prioridade' (Higher priority) at the top.

Os operadores lógicos têm uma prioridade inferior relativamente aos operadores aritméticos e relacionais.

Operadores Lógicos – Exemplos Prioridade

1 && !0 || 1

!0 = 1

1 && 1 = 1

1 || 1 = 1

1 && !(0 || 1)

(0 || 1) = 1

!1 = 0

1 && 0 = 0

Operadores lógicos -Exemplo

```
int i,j;  
printf(" Introduza dois valores. \n");  
scanf("%d %d",&i,&j);  
  
printf(" %d E %d é %d\n",i,j,(i > 20 ) && (j > 10));  
printf(" %d E %d é %d\n",i,j,i > 20 && j > 10);  
printf(" %d OU %d é %d\n",i,j,i || j);  
printf(" NÃO %d é %d\n",j,!j);
```