



SCC0221 – Introdução à Ciência de Computação I

Prof.: Dr. Rudinei Goularte

(rudinei@icmc.usp.br)

Operadores e condicionais em C

Instituto de Ciências Matemáticas e de Computação - ICMC
Sala 4-229



Operadores

Unários:

Operador	Significado	Exemplo (variável x)
+	mais unário ou positivo	+x;
-	menos unário ou negativo	- x;
!	NOT ou negação lógica	!x;
&	endereço	&x;
*	conteúdo (ponteiros)	(*x);
++	pré ou pós incremento	++x; ou x++;
--	pré ou pós decremento	--x; ou x--;



Operadores

Binários:

<u>Operador</u>	<u>Significado</u>	<u>Exemplo (variáveis x e y)</u>
+	adição de dois números	$x + y;$
-	subtração de dois números	$x - y$
*	multiplicação de dois números	$x * y;$
/	quociente de dois números	$x / y;$
%	resto da divisão inteira	$x \% y;$



Operadores bit a bit

Operações bit-a-bit (vetores)

<<	desloca à esquerda	<code>/* x << 2 */</code>
>>	desloca à direita	<code>/* x >> 2 */</code>
^	ou exclusivo	<code>/* x ^ 0xF0 */</code>
&	E bit-a-bit	<code>/* x & 0x07 */</code>
	OU bit-a-bit	<code>/* x 0x80 */</code>
~	Complemento bit-a-bit	<code>/* ~ x */</code>



Atribuição

= atribui

+= soma e atribui

-= subtrai e atribui

*= multiplica e atribui

/= divide e atribui quociente

%= divide e atribui resto

&= E bit-a-bit e atribui

|= OU bit-a-bit e atribui

<<= shift left e atribui

$x = y;$

$x += y; \Leftrightarrow x = x + y;$

$x -= y; \Leftrightarrow x = x - y;$

$x *= y; \Leftrightarrow x = x * y;$

$x /= y; \Leftrightarrow x = x / y;$

$x \% = y; \Leftrightarrow x = x \% y;$

$x \& = y; \Leftrightarrow x = x \& y;$

$x |= y; \Leftrightarrow x = x | y;$

$x \<\< = y; \Leftrightarrow x = x \<\< y;$

...



Operadores Relacionais

- Aplicados a variáveis que obedecem a uma relação de ordem, retornam 1 (true) ou 0 (false)

Operador

>

>=

<

<=

==

!=

Relação

Maior do que

Maior ou igual a

Menor do que

Menor ou igual a

Igual a

Diferente de



Operadores Lógicos

- Operam com valores lógicos e retornam um valor lógico verdadeiro (1) ou falso (0)

Operador	Função	Exemplo
&&	AND (E)	<code>c >= '0' && c <= '9'</code>
	OR (OU)	<code>(a == 'F' b != 32)</code>
!	NOT (NÃO)	<code>(!var)</code>



Tabela Verdade

a	b	!a	!b	a && b	a b
0	0	1	1	0	0
0	1	1	0	0	1
1	0	0	1	0	1
1	1	0	0	1	1

Precedência

Maior precedência

() [] ->

! ~ ++ -- . -(unário) (cast) *(unário) &(unário) sizeof

* / %

+ -

<< >>

<<= >>=

== !=

&

^

|

&&

||

?

= += -= *= /=

Menor precedência



Expressões

- expressões são compostas por:
 - operandos: a , b , x , Meu_dado , 2 , ...
 - operadores: $+$, $-$, $\%$, ...
 - pontuação: $()$, $\{ \}$, $"$, $'$

- Ex:

x

14

$x + y$

$(x + y) * z + w - v$



Expressões

- expressões retornam um valor:

`x = 5 + 4 /* retorna 9 */`

- esta expressão retorna 9 como resultado da expressão e atribui 9 a x

`((x = 5 + 4) == 9) /* retorna true */`

- na expressão acima, além de atribuir 9 a x, o valor retornado é utilizado em uma comparação



Expressões

- expressões podem aparecer em diversos pontos de um programa
 - comandos `/* x = y; */`
 - parâmetros de funções `/* sqrt(x + y); */`
 - condições de teste `/* if (x == y) */`
- a ordem em que uma expressão é avaliada depende da prioridade dos operadores e da pontuação.



Comando de Seleção if

- Forma geral:

if (*expressão*) *sentença*;
else *sentença*;

- *sentença* pode ser uma única sentença, um bloco de sentenças, ou nada.
- O **else** é opcional.



Comando de Seleção if

if (*expressão*) *sentença*;
else *sentença*;

- Se *expressão* é verdadeira ($\neq 0$), a *sentença* seguinte é executada. Caso contrário, a *sentença* do **else** é executada.
- O uso de if-else garante que apenas uma das *sentenças* será executada.



Comando de Seleção if

- O comando if pode ser aninhado.
 - Um comando if aninhado é um if que é sentença de outro comando if ou else.
 - ANSI C especifica mínimo de 15 níveis.
- Cuidado: um else se refere, sempre, ao if mais próximo, que está dentro do mesmo bloco do else e não está associado a outro if.



Exercícios

- O que será impresso? Explique.

```
#include <stdio.h>
int main (void){
    int a = 1, b = 2, c = 3;

    if (a == 2)
        if(b == 3)
            printf("%d", a);
    else
        printf("%d", b);
    printf("%d", c);
    return(0);
}
```




O Comando switch

- `switch (expressão){`
 - `case constante1: sequência1; break;`
 - `case constante2: sequência2; break;`
 - `...`
 - `default: sequência_n;``}`



O Comando switch

- Padrão ANSI especifica que switch pode ter, pelo menos, 257 comandos case.
- switch só pode testar igualdade.
- Duas constantes case no mesmo switch não podem ter valores idênticos.
- Se constantes caracter são usadas em um switch, elas são automaticamente convertidas em seus valores inteiros.
- break é opcional.
- default é opcional.
- switch pode ser aninhado.



O Comando switch

```
#include <stdio.h>
int main (void){
```

```
    char ch;
    scanf("%c", &ch);
```

```
    switch (ch){
        case '1': printf("1"); break;
        case '2': printf("2"); break;
        case 'a': printf("a"); break;
        case '8': printf("8"); break;
        case '3': { printf("3");
                    printf("\n três");
                    break;
                }
```

```
        default: printf("Nada!");
    return(0);
```

```
}
```



Para Saber Mais...

- Kelley, A.; Pohl, I. "A Book on C".
Capítulos 2, 3 e 4.
- SCHILDT, H. "C - Completo e Total",
MakronBooks, 1997.