

Técnicas de Programação em Engenharia Elétrica I

- Engenharia de Sistemas Embarcados -

Prof. Alessandro Zimmer

Introdução à
Linguagem C

Operadores aritméticos

- Operadores Unários (um só operando)

– Incremento :

Pré Incremento

$A = 5 ;$

$B = ++ A ;$

$B = 6 , A = 6 !!$

Pós Incremento

$A = 5 ;$

$B = A ++;$

$B = 5 , A = 6 !!$

Operadores aritméticos

- Operadores Unários (um só operando)

– Decremento :

Pré Decremento

$$A = 7 ;$$

$$B = -- A ;$$

$$B = 6 , A = 6 !!$$

Pós Decremento

$$A = 7 ;$$

$$B = A --;$$

$$B = 7, A = 6 !!$$

Operadores aritméticos

- Operadores Binários (2 operandos)
 - adição : + (ex.: $x = x + 1$; $y = b + 2$;)
 - subtração : - (ex.: $y = y - 1$; $y = 3 - 5$;))
 - multiplicação : * (ex.: $a = j * k$; $a = 3 * 2$;)
 - divisão : / (ex.: $x = 5 / y$; $a = 4.35 / 3.93$;))
 - módulo : % (ex.: $x = 5 \% 2$; $a = b \% c$;)

Operadores aritméticos

- Operadores Binários – Formato Reduzido

$$x += 5; \quad \rightarrow x = x + 5;$$

$$x -= 5; \quad \rightarrow x = x - 5;$$

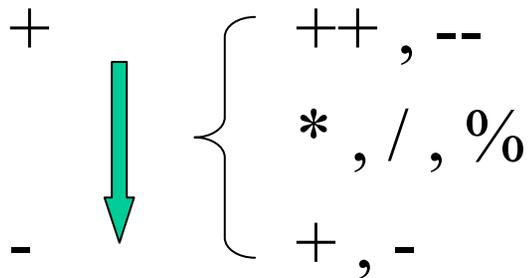
$$x *= 5; \quad \rightarrow x = x * 5;$$

$$x /= 5; \quad \rightarrow x = x / 5;$$

$$x \% = 5; \quad \rightarrow x = x \% 5;$$

Precedência de Operadores Aritméticos

- Ordem de precedência :



- Se dois operandos têm a mesma precedência, serão avaliados da esquerda para a direita.
- Utiliza-se parênteses para alterar a precedência das operações.

Precedência de Operadores Aritméticos

• Ex:

$$A = 3 ;$$

$$C = 5 ;$$

$$\text{media} = A + C / 2 ;$$

$$\text{media} = (A + C) / 2 ;$$

Atenção !! → Erro não detectado pelo compilador!!

Conversão de Tipos

- Se dois tipos diferentes são usados em uma mesma expressão, o tipo de menor representação é convertido implicitamente para o de maior representação (promoção de tipo), segundo:
char → short int → long int → float → double
- De preferência não misturar tipos nas expressões (ou convertê-los via operador *cast*).

Conversão de Tipos

- Ex:

char ch;

int i;

float f;

double d;

$$RES = (ch / i) + (f * d) - (f + i);$$

RES será de que tipo?

Operador Cast

- Força a conversão de tipos.
- Uso : (tipo) expressão;

Ex.:

float x;

x = (float) 3 / 2;

X = 1.5

x = 3 / 2;

x = 1.0 !!!

Operadores relacionais

- Usados na avaliação de expressões. Resultam sempre em falso (0) ou verdadeiro (-1, ou $\neq 0$)
- Operadores :
 - Igual : $==$
 - Maior que : $>$
 - Menor que : $<$
 - Maior ou igual a : $>=$
 - Menor ou igual a : $<=$
 - Diferente de : $!=$

Operadores lógicos

- Avaliam duas ou mais expressões relacionais.
- Operadores :
 - E (and) : `&&`
 - OU / (or) : `||`
 - NÃO / (not) : `!`
- Ex.:
 $(5 > 2) || (8 < 4)$ $(5 > 2) \&\& (8 < 4)$

Operadores Bit a Bit

- Manipulam diretamente os bits de valores inteiros (ou variáveis).
 - Deslocamento (shift) : \ll (esq) , \gg (dir)
 - Complemento : \sim
 - E (and) : $\&$
 - OU (or) : $|$
 - OU EXCLUSIVO (xor) : \wedge

Controle de Fluxo

Comando de Seleção/Alternativa

Instrução IF/ELSE

- Decide pela execução de um bloco de programa ou outro, de acordo com o resultado de uma expressão.
- Sintaxe

```
if (expressão)
{
    // bloco de programa se expressão
    verdadeira
}
else
{
    // bloco se expressão falsa
}
```

Controle de Fluxo

Exemplo 1:

```
#include <stdio.h>
void main(void)
{
    int a=5, b=7;

    if (a > b)
    {
        printf("A é maior que B\n");
    }
    else
    {
        printf("A não é maior que B\n");
    }
}
```

Controle de Fluxo

Exemplo 2:

```
#include <stdio.h>
void main(void)
{
    int a=5, b=7;
    if ((a > b) && (a < 0))
    {
        printf("A é maior que B\n");
        printf("A e B são negativos!\n");
    }
    else
        if (a > 0)
            printf("A é positivo!\n");
        else
            printf("A é negativo ou nulo!\n");
}
```

Controle de Fluxo

Exemplo 3:

```
#include <stdio.h>
void main(void)
{
    int a=5, b=7;
    if (a > b)
        if ( a == 0)
            printf("B é negativo !\n");
        else
            if ( b == 0)
                printf("A é positivo!\n");
            else
                printf("A é menor ou igual a
B\n");
}
```

Controle de Fluxo

Comando de Seleção/Alternativa

Operador Ternário ?:

- Sintaxe

<expressão 1> ? <expressão 2> : <expressão 3>

- Se <expressão 1> for verdadeira, retorna o resultado da <expressão 2>, caso contrário retorna a <expressão 3>.

Controle de Fluxo

Exemplo 1:

```
#include <stdio.h>
#include <math.h>
void main(void)
{
    float x,res;
    printf("Digite um número:\n");
    scanf("%f", &x);
    res = (x >= 0) ? sqrt(x) : -1;
    res <0 ? printf("Numero negativo!\n") :
    printf("Raiz %f", res);
}
```

Controle de Fluxo

Exemplo 2:

```
#include <stdio.h>
void main(void)
{
    int mes,dias;
    mes = dias = 0;
    printf("Entre com o mês:\n");
    scanf("%d", &mes);
    dias = (mes == 1 || mes == 3 || mes == 5 || mes
    == 7 || mes == 8 || mes == 10 || mes == 12) ? 31
    : ((mes == 2) ? 28 : 30);
    printf("O mês: %d tem %d dias\n", mes, dias);
}
```