UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

Fundamentos de Programação

CP41F

Operadores aritméticos. Operadores lógicos. Operadores binários.

Aula 7

Prof. Daniel Cavalcanti Jeronymo

Universidade Tecnológica Federal do Paraná (UTFPR)
Engenharia de Computação – 1º Período

Plano de Aula

- Operadores aritméticos
  - soma, subtração, multiplicação, divisão, módulo, incremento e decremento
- Operadores lógicos
  - not, and, or
- Operadores binários
  - Not, and, or, xor, shift left, shift right

Operadores aritméticos

- Operadores aritméticos
  - Considere A = 10 e B = 20

Operador	Descrição	Exemplo
+	Adiciona dois operandos	A + B = 30
-	Subtrai o Segundo operando do primeiro	A - B = -10
*	Multiplica dois operandos	A * B = 200
/	Divide o primeiro operando pelo Segundo	B / A = 2
%	Resto de uma divisão de inteiros	B % A = 0
++	Incremento por um	A++ = 11
--	Decremento por um	A-- = 9

Operadores aritméticos

- Operadores aritméticos
  - Unários – utilizam apenas um operando
    - +a – promoção de inteiro
    - -a – aditiva inversa
    - ++a e a++ são diferentes
    - --a e a-- também

Operadores aritméticos

- Operadores aritméticos
  - Promoção de inteiro
  - Se um int pode representar todos os valores de um tipo, o valor é convertido para int; caso contrário, é convertido para unsigned int.
  - signed char: -127 a 127
  - unsigned char: 0 a 255
  - signed short: -32767 a 32767
  - unsigned short: 0 a 65535
  - signed int: -2147483647 a 2147483647
  - unsigned int: 0 a 4294967295

Operadores aritméticos

- Operadores aritméticos
  - Promoção de inteiro em +a

```c
#include <stdio.h>

int main(void)
{
   char ch;
   short sh;
   int i;

   /* saida: 1 2 4 */
   printf("%d %d %d\n", sizeof(ch), sizeof(sh), sizeof(i));
   /* saida: 4 4 4 */
   printf("%d %d %d\n", sizeof(+ch), sizeof(+sh), sizeof(i));
}
```

Operadores aritméticos

- Operadores aritméticos
  - Promoção de inteiro – abaixo, a*b = 1200 que é maior que um inteiro, o código funciona?

```c
#include <stdio.h>

int main()
{
   char a = 30, b = 40, c = 10;
   char d = (a * b) / c;
   printf ("%d ", d);
   return 0;
}
```

Operadores aritméticos

- Operadores aritméticos
  - Conversão de lvalue para rvalue – o código abaixo funciona?

```
#include <stdio.h>

int main(void)
{
   char a = 10;
   char b = (+a)++;
}
```

Operadores aritméticos

- Operadores aritméticos
  - Ordem de incremento ou decremento

```c
#include <stdio.h>

int main(void)
{
   char a = 10;
   char b = 20;

   printf("a: %d b: %d\n", ++a, b++);
   printf("a: %d b: %d\n", a, b);
}
```

Operadores lógicos

- Operadores lógicos
  - A = 1 e B = 0

Operador	Descrição	Exemple
&&	AND lógico. Se ambos os operandos forem não-zero, então a condição é verdadeira.	(A && B) é falso.
||	OR lógico. Se qualquer um dos operandos for não-zero, então a condição é verdadeira.	(A || B) é verdade.
!	NOT lógico. Reverte o estado lógico do operando.Called Logical NOT Operator. Use to reverses the logical state of its operand. If a condition is true then Logical NOT operator will make false.	!(A && B) é verdade.

Operadores lógicos

- Operadores lógicos
  - Tabela verdade

P	Q	p && q
T	T	T
T	F	F
F	T	F
F	F	F

P	Q	p || q
T	T	T
T	F	T
F	T	T
F	F	T

P	!P
T	F
F	T

Operadores lógicos

- Operadores lógicos
  - Avalie as seguintes condições
    - 1) (true && true) || false
    - 2) (false && true) || true
    - 3) (false && true) || false || true

Operadores lógicos

- Operadores lógicos
  - Avalie as seguintes condições

- 1) (true && true) || false = true
- 2) (false && true) || true = true
- 3) (false && true) || false || true = true

Operadores binários

- Operadores binários
  - A = 0xF0 e B = 0x0F

Operador	Descrição	Exemplo
&	AND binário. Copia um bit para o resultado caso exista em ambos operandos.	(A & B) = 0x00
|	OR binário. Copia um bit para o resultado caso exista em qualquer um dos operandos.	(A | B) = 0xFF
^	XOR binário. Copia um bit para o resultado caso seja diferente entre os operandos.	(A ^ B) = 0xFF
~	Negação ou complement. Operador unário para inversão de bits.	(~A) = 0x0F
<<	Operador de deslocamento à esquerda. O valor do operando a esquerda é movido pela quantidade de bits especificada pelo operando a direita.	A << 1 = 0x1E0
>>	Operador de deslocamento à direita. O valor do operando a esquerda é movido pela quantidade de bits especificada pelo operando a direita.	A >> 1 = 0x78

Operadores binários

- Operadores binários
  - Utilidades:

    Bit Twiddling Hacks

    http://graphics.stanford.edu/~seander/bithacks.html

  - Exemplo: XOR swap