

Linguagem C

(conversão de tipos)

André Tavares da Silva

andre.silva@udesc.br

Conversão de tipos

- Quando são realizadas operações com variáveis de tipos diferentes, é necessário converter as variáveis para um mesmo tipo para realizar adequadamente uma operação.
- Estas conversões podem ser explícitas ou implícitas. Explícita é quando o programador indica qual a conversão deve ser feita e implícita é feita pelo compilador sem a intervenção do programador.

Conversão de tipos

- No código abaixo, ao escrever `i = a`, essa não é uma atribuição comum, pois o compilador precisa gerar código extra para converter o *char* para *int*. Esta conversão é feita de maneira implícita e silenciosa pelo compilador porque qualquer valor de *char* pode ser armazenado em um *int*.

```
int i;  
char a;  
a = 64;  
i = a;
```

Conversão de tipos

- Ao se escrever $a = i$, alguns compiladores gerarão aviso indicando a perda de informação. Isto porque normalmente valores de *int* (4 bytes) não podem ser armazenados em um *char* de apenas 1 byte.

```
int i;  
char a;  
i = 64321;  
a = i;
```

Conversão explícita: *cast*

- O operador *cast* tem a função de forçar a conversão de valores para um tipo diferente.
- A sintaxe usada para realizar um *cast* é colocar o nome do tipo entre parênteses antes do valor a ser convertido.

```
int i;  
char a;  
i = 64321;  
a = (char) i;
```

Operação com inteiro

- Ao se fazer operações com números inteiros (principalmente divisão), é necessário tomar alguns cuidados. Isso porque ao se utilizar variáveis e números inteiros o compilador fará a operação utilizando valores inteiros. Exemplos:

```
int a,b;  
float x;  
a = 7;  
b = 4;  
x = a/b; // x = 1 (7/4 = 1)
```

```
int a,b;  
float media;  
a = 7;  
b = 4;  
media = (a+b)/2; // x = 5
```

Operação com inteiro

- Para solucionar o problema, podemos utilizar *cast* ou forçar a conversão implícita para valor real. Ao colocar um valor real na operação, o compilador fará a operação utilizando valores reais.

```
int a,b;  
float x;  
a = 7;  
b = 4;  
x = (float)a/b; // x=7.0/4
```

```
int a,b;  
float media;  
a = 7;  
b = 4;  
media = (a+b)/2.0; // x=5.5
```

Operação com valores reais

- Devido ao problema em relação à precisão dos tipos usados pelo computador, também é preciso tomar cuidado com valores reais, principalmente em relação à comparação (condição de *if*'s, *while*'s, *for*,...).

```
if (1.8 == (1.8/3.0)*3.0)
    printf("Iguais");
else
    printf("Diferentes");
```


Operação com valores reais

- Para resolver esse problema, em vez de fazer a comparação com igualdade, verifica-se se a diferença entre ambos os valores é mínima. Esta diferença dependerá da precisão desejada na aplicação.

```
if (fabs(1.8 - (1.8/3.0)*3.0) < 0.00001)
    printf("Iguais");
else
    printf("Diferentes");
```