

Tipos de Dados

Módulo 2 Aula 5

Linguagem C, o Curso Definitivo WR Kits

Autor: Dr. Eng. Wagner Rambo

Tipos de dados básicos

- ▶ Sem considerar ponteiros, que estudaremos em módulos mais adiante, a linguagem C tem 4 tipos de dados básicos (além do void que se aplica às funções, assunto para os módulos futuros).
- ▶ char: armazena valores inteiros e caracteres. Ocupa 1 byte de memória;
- ▶ int: armazena valores inteiros. Ocupa 2 ou 4 bytes de memória, dependendo da máquina usada;
- ▶ float: armazena números reais. Ocupa 4 bytes da memória;
- ▶ double: armazena números reais com maior precisão. Ocupa 8 bytes de memória.

Inteiros int

- ▶ int de 2 bytes pode guardar valores de -32768 a 32767;
- ▶ int de 4 bytes pode guardar valores de -2147483648 a 2147483647.

Reais float

- ▶ Armazenam valores numéricos com parte fracionária (ponto flutuante);
- ▶ Valores com 6 dígitos de precisão.

Reais double

- ▶ Armazenam valores numéricos com parte fracionária (ponto flutuante);
- ▶ Valores com 10 a 12 dígitos de precisão.

Caracteres char

- ▶ Utilizado para armazenar caracteres (conforme tabela ASCII) e também pode guardar valores inteiros;
- ▶ char tem 1 byte, portanto armazena valores de -128 a 127.

Modificadores de tipo

- ▶ Existem 4 modificadores:
- ▶ unsigned: determina que o tipo não terá sinal (int ou char);
- ▶ signed: determina que o tipo terá sinal (normalmente seu uso é redundante, pois ao declararmos uma variável, já está implícito que a mesma tem sinal, caso não utilizemos unsigned). Pode ser usado em algumas implementações onde char por padrão não tem sinal;
- ▶ short: usado principalmente para estabelecer o limite de 2 bytes para o tipo int;
- ▶ long: usado para estabelecer o limite de 4 bytes para o tipo int e também para aumentar a precisão do tipo double;

Vantagem do unsigned

- ▶ Aumentar o espaço positivo em variáveis do tipo int e char, quando trabalhamos com este tipo de valores.
- ▶ unsigned int de 2 bytes vai de 0 a 65535;
- ▶ unsigned int de 4 bytes vai de 0 a 4294967296;
- ▶ unsigned char vai de 0 a 255;
- ▶ OBS.: para garantir um int de 2 bytes utilize unsigned short int; e para garantir um int de 4 bytes utilize unsigned long int;

int implícito

- ▶ Se você declarar variáveis assim:

```
unsigned guarda_valor;  
short guarda_data;  
long salva_numero;  
unsigned short val1;  
unsigned long val2;
```

- ▶ Serão todas do tipo inteiro, pois o int estará implícito no contexto.

Formatos de leitura e escrita

- ▶ As letras após o % (como já vimos na função printf), têm o objetivo de determinar o formato do dado que será lido ou escrito.
- ▶ %c char para um único caractere
- ▶ %d int para um inteiro em base decimal (%i também é usado)
- ▶ %o int para um inteiro em base octal
- ▶ %x int para um inteiro em base hexadecimal (letras minúsculas)
- ▶ %X int para um inteiro em base hexadecimal (letras maiúsculas)
- ▶ %f float para números reais (o mesmo para o tipo double)
- ▶ %e float para notação científica com 'e' minúsculo (double também)
- ▶ %E float para notação científica com 'E' maiúsculo (double também)

Formatos de leitura e escrita nos modificadores

- | | | |
|-------|----------------|-------------------------------|
| ▶ %hd | short int | short inteiro em base decimal |
| ▶ %ld | long int | long inteiro em base decimal |
| ▶ %hu | unsigned short | short inteiro positivo |
| ▶ %u | unsigned int | inteiro positivo |
| ▶ %lu | unsigned long | long inteiro positivo |

Bibliografia

- ▶ DAMAS, Luís; Linguagem C, décima edição.
- ▶ Disponível em: <https://amzn.to/3nGdlbN>