# Introdução à linguagem C

Luís Charneca

luis.charneca@gmail.com

# Introdução ao C

- O C nasceu na década de 70. O seu inventor, Dennis Ritchie, implementou-o pela primeira vez usando um DEC PDP-11 correndo o sistema operacional UNIX. O C é derivado de uma outra linguagem: o B, criado por Ken Thompson. O B, por sua vez, veio da linguagem BCPL, inventada por Martin Richards.
- O C é uma linguagem de programação genérica que é utilizada para a criação de programas diversos como processadores de texto, sistemas operativos, programas de comunicação, programas para a automação industrial, gestores de base de dados, programas de projecto assistido por computador, programas para a solução de problemas da Engenharia, Física, Química e outras Ciências, etc ...

# Principais características

- **Alto grau de portabilidade**, que permite que os seus programas fonte sejam transportados entre máquinas sem grandes problemas
- **É de uso geral**, sendo eficiente tanto para programação de utilitários, como para sistemas operativos, processadores de texto, bases de dados e sistemas aplicativos em geral.
- **Gera código executável** (objecto) compacto e rápido em relação à maioria das outras linguagens compiladas, tornando a sua aplicação eficiente para micro-computadores de memória reduzida.
- Permite **total interacção** com o sistema operativo e inserção de código «assembly» no programa fonte, sendo assim ilimitada por software.
- Possui uma sintaxe de **poucos comandos** e um grande número de operadores aritméticos e lógicos.
- É uma **linguagem estruturada e modular**, o que permite a simplificação do uso das mais modernas técnicas de programação.
- Permite **estruturas de dados compostos** na forma de registos e campos.

# O C é "CASE SENSITIVE"

- O C é "Case Sensitive", isto é, *maiúsculas e minúsculas fazem diferença*. Se se declarar uma variável com o nome soma ela será diferente de **Soma**, **SOMA**, **SoMa** ou **sOmA**. Da mesma maneira, os comandos do C **if** e **for**, por exemplo, só podem ser escritos em minúscula, caso contrário o compilador não irá interpretá-los como comandos, mas sim como variáveis.

# PRIMEIROS PROGRAMAS

**Veja-se um primeiro programa em C:**

```
#include <stdio.h>
void main () /* Um Primeiro Programa */
{
printf ("Olá! Eu estou vivo!\n");
}
```

- Compilando e executando este programa dará como resultado a mensagem **Olá! Eu estou vivo!** no ecrã.

# Análise do programa por partes (1):

- A linha **#include <stdio.h>** diz ao compilador que ele deve incluir o arquivo-cabeçalho (biblioteca) stdio.h.
- Neste arquivo existem declarações de funções úteis para entrada e saída de dados (**std** = standard, padrão em inglês;

  **io** = Input/Output, entrada e saída

  **stdio** = Entrada e saída padronizadas).

O C possui diversos arquivos-cabeçalhos (bibliotecas).

# Análise do programa por partes (2):

- Quando se faz um programa, uma boa ideia é usar comentários que ajudem a elucidar o funcionamento do mesmo. No caso acima tem-se um comentário: **/* Um Primeiro Programa */**.

O compilador C ignora qualquer coisa que esteja entre **/*** e ***/**.
Um comentário pode, inclusive, ter mais de uma linha.
Outra forma aceite pelo C para utilizar comentários é utilizar os caraceteres **//**.
O C neste caso não considera a linha que tem como início os caracteres **//**.
Este tipo de comentário só pode ter uma linha.

# Análise do programa por partes (3):

- A linha **void main()** define uma função de nome main. Todos os programas em C têm que ter uma função main, pois é esta função que será chamada quando o programa for executado.
- O conteúdo da função é delimitado por chavetas **{ }**.
- O código que estiver dentro das chavetas será executado sequencialmente quando a função for chamada. A palavra **void** indica que esta função não retorna nada, isto é seu retorno é vazio.
- Posteriormente, ver-se-á que as funções em C podem retornar valores.

# Análise do programa por partes (4):

- A única coisa que o programa realmente faz é chamar a função **printf()**, passando a string (uma string é uma sequência de caracteres) **"Olá! Eu estou vivo!\n"** como argumento.
- É por causa do uso da função **printf()** pelo programa que se deve incluir o arquivo_cabeçalho **stdio.h** . A função **printf()** neste caso irá apenas colocar a string no ecrã do computador.

# Análise do programa por partes (5):

- O **\n** é uma constante chamada de **constante barra invertida**. No caso, o **\n** é a **constante barra invertida de "nova linha"** e ela é interpretada como um comando de mudança de linha, isto é, após imprimir **Olá! Eu estou vivo!** o cursor passará para a próxima linha.
- É importante observar também que os comandos do C terminam com **;**

# Exemplo de um programa um pouco mais complicado

```
#include <stdio.h>
void main ()
{
// Declaração de Variáveis
int Dias;
float Anos;
printf ("Entre com o número de dias: "); /* Entrada de
                                            Dados */
scanf ("%d",&Dias);
Anos=Dias/365.25;        /* Conversão Dias -> Anos */
printf ("\n\n%d dias equivalem a %f anos.\n",Dias,Anos);
}
```

# O Caracter Especial \

| | |
|---|---|
| \7 | Bell (sinal sonoro do computador) |
| \a | Bell (sinal sonoro do computador) |
| \b | Backspace (Retrocesso) |
| \n | New Line (Mudança de linha) |
| \r | Carriage Return |
| \t | Tabulação Horizontal |
| \v | Tabulação Vertical |
| \\ | Carácter \ |
| \’ | Carácter ‘ |
| \” | Carácter “ |
| \? | Carácter ? |

# Em Resumo…

- Em C um programa começa com a função **main**
- O código a executar é colocado entre **{ }**
- Um **Bloco** é formado por qualquer conjunto de instruções entre **{ }**
- Cada instrução deve ser terminar em **;**
- A disposição do código é arbitrária e depende das preferências de cada programador
- O C faz distinção entre maiúsculas e minúsculas – diz-se que é **Case Sensitive**
- As strings em C são delimitadas pelo carácter aspas **“ ”**
- Para escrever informação no ecrã usa-se a função **printf()**

# Em Resumo… (continuação)

- A função **printf()** não faz parte da linguagem C. Pertence à sua extensa biblioteca de funções
- Para acesso a esta e outras funções de *input/output* devemos incluir nos nossos programas o ficheiro **stdio.h** através da directiva ao compilador **#include <stdio.h>**
- As linhas começadas por # não são C, mas antes directivas ao compilador, por isso não devem terminar em **;**
- A representação de caracteres especiais faz-se através de um conjunto de dois ou mais caracteres, sendo o primeiro a barra invertida ****
- Os comentários são escritos entre ***/*** e ******* e são simplesmente ignorados pelo compilador