

Linguagem de Programação II

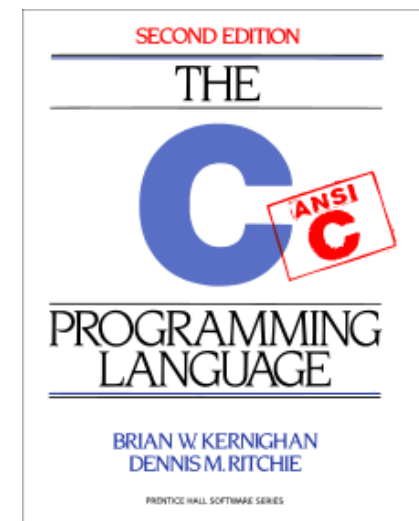
Prof. Mario Bessa

Aula 1

<http://mariobessa.info>

Bibliografia

- Deitel, Paul; Deitel Harvey. Linguagem C, 6ª Edição. Pearson Editora, 2011.
- Kernighan, Brian W.; Ritchie, Dennis M. The C Programming Language, 2nd Edition. Prentice Hall, 1988.



IDE

(Integrated Development Environment)

- Code::Blocks (Windows)

<http://www.codeblocks.org>

<http://www.youtube.com/watch?v=LfIS1YQcZT>



Code::Blocks

- Xcode (Mac OS)

App store

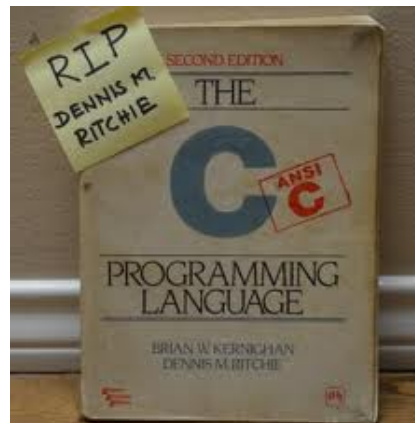


Por que aprender C?

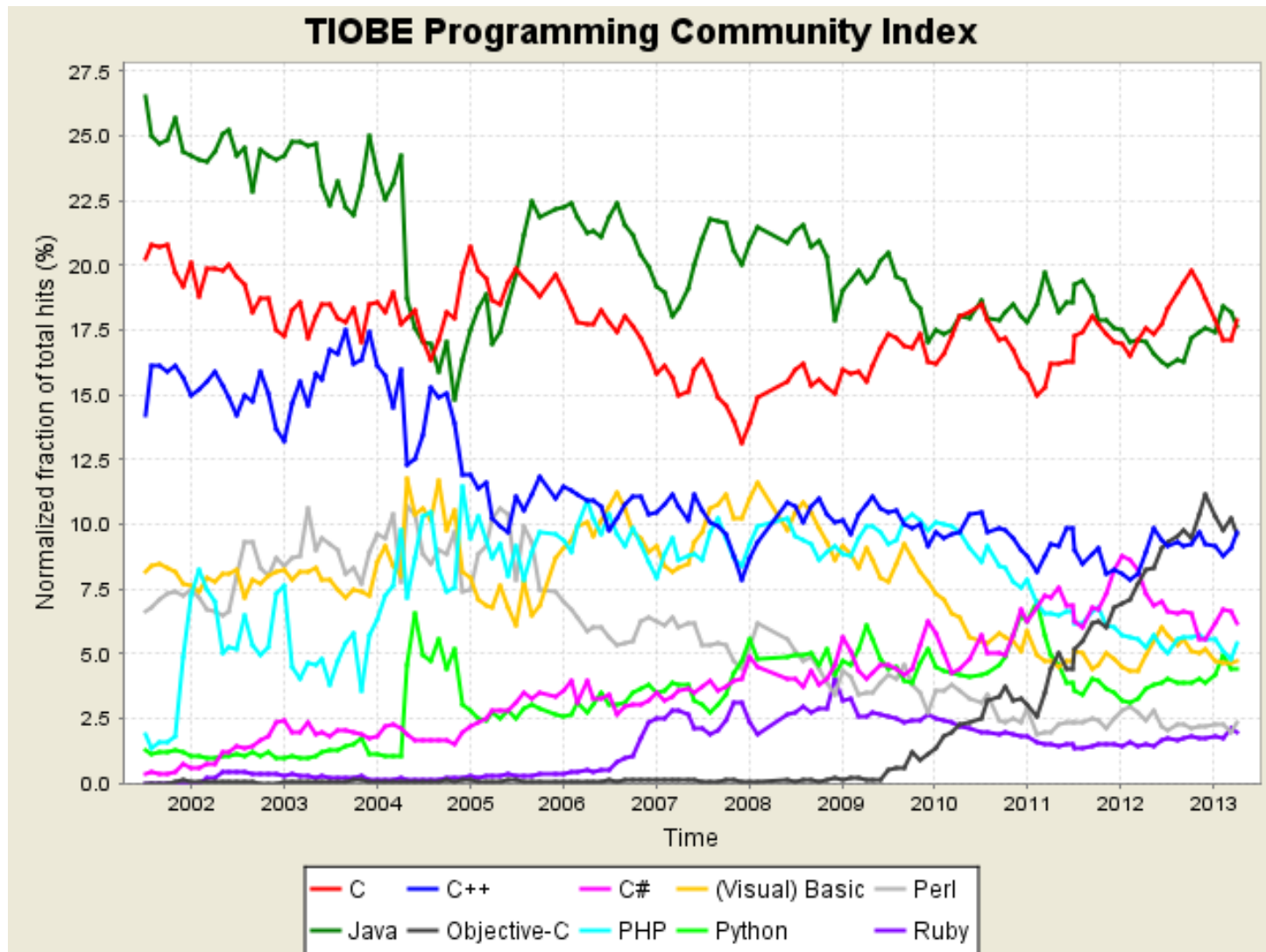
- Linguagem de programação genérica que permite criar diversos tipos de aplicações (portable).
- Linguagem que pode ser usada em programação de baixo e alto nível.
- Compacta, rápida e poderosa.
- Muitas linguagens tem C como pedigree: java, php, objective-C etc.

A origem de C

- Desenvolvida em 1972 por Dennis Ritchie no Bell Labs e tinha como objetivo desenvolver a estrutura do sistema Unix.
- Surgem diversas implementações de C.
- ANSI C (American National Standard Institute), em 1983 padronizou a linguagem C.



Utilização de C

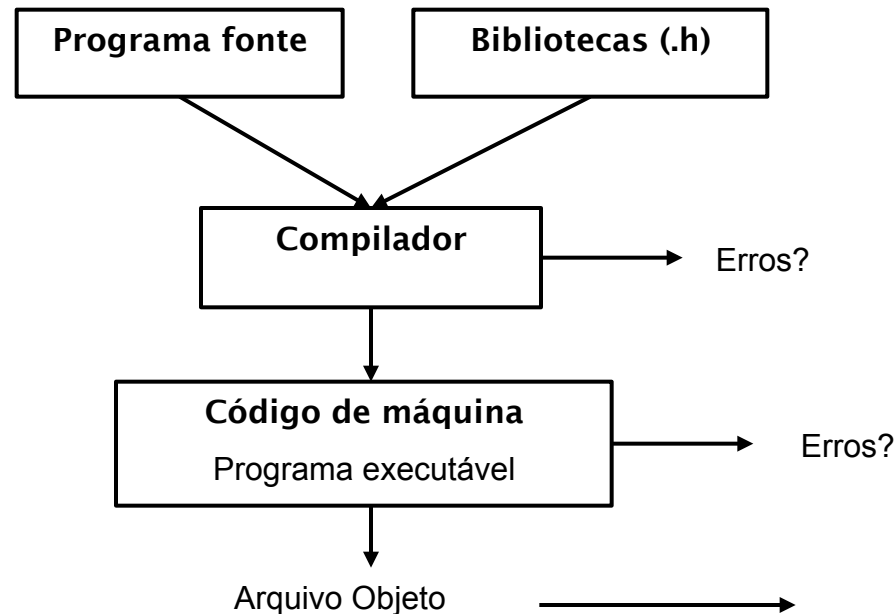


Característica da linguagem C

- Rapidez
- Simples
- Portátil
- Popular
- Alto nível / Baixo nível

Característica da linguagem C

- Compilada
 - indica que o código é convertido para um executável binário



01000111 01100101 01110010 01110110 11100001 01110011 0



Característica da linguagem C

- Modular

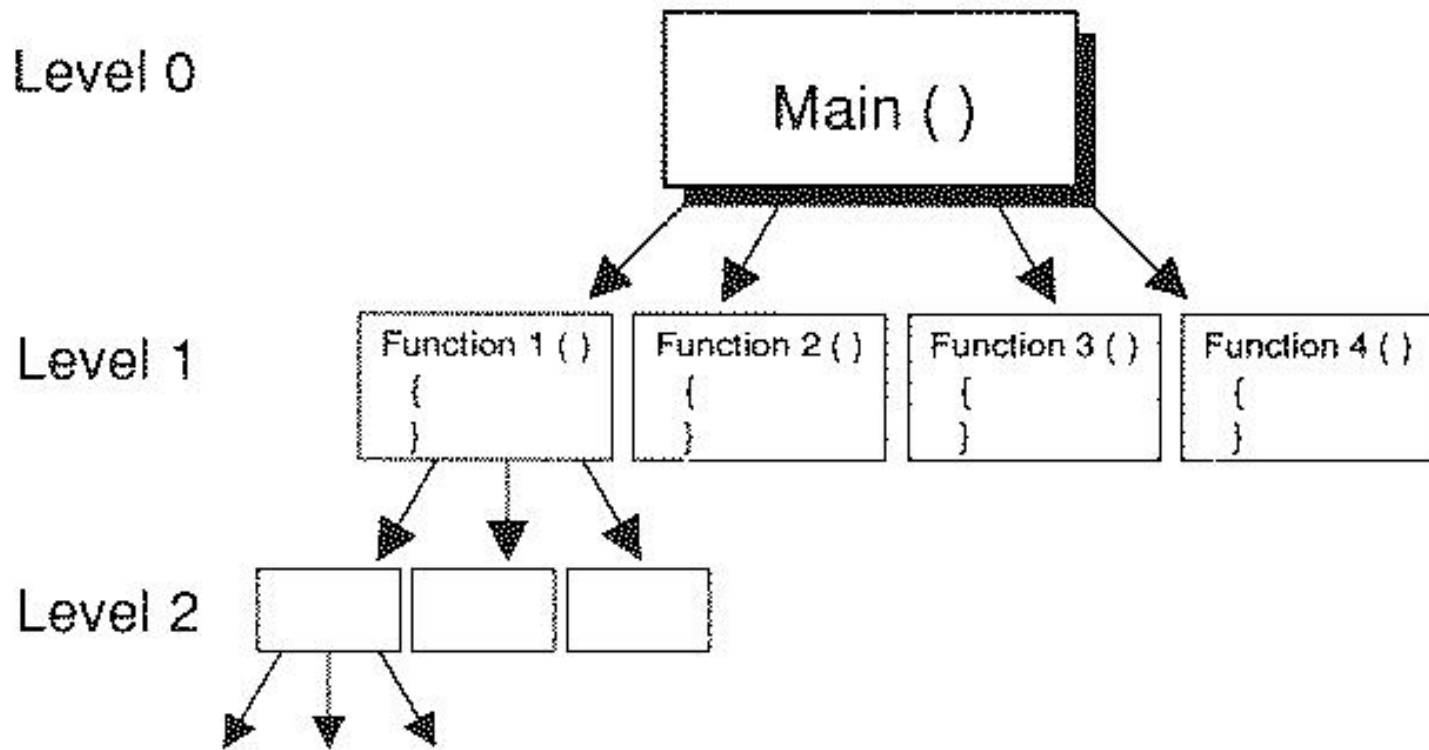


Figure 7.1. C Programs are built from functions.

Característica da linguagem C

- A linguagem C é case sensitive.
- A linguagem C possui um conjunto reduzido de palavras-chave: 32.

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

Característica da linguagem C

- Bibliotecas muito poderosas.
 - O núcleo da linguagem C é pequeno e simples.
 - A funcionalidade da linguagem é provida através do uso de bibliotecas com funções já prontas.
 - Não precisamos reinventar a roda.
 - Isso é o que faz C tão portátil.
 - Você também pode fazer seu conjunto de funções e disponibilizar através de bibliotecas.
 - O cabeçalho carrega códigos necessários para a execução de determinadas tarefas que serão descritas no corpo do programa

Uso de bibliotecas

Cabeçalho	Explicação
<assert.h>	Contém macros e informações que acrescentam diagnósticos que auxiliam a depuração do programa.
<ctype.h>	Contém protótipos de função para funções que testam certas propriedades dos caracteres, e protótipos de função para funções que podem ser usadas para converter letras minúsculas em maiúsculas, e vice-versa.
<errno.h>	Define macros que são úteis na comunicação de condições de erro.
<float.h>	Contém os limites de tamanho de ponto flutuante do sistema.
<limits.h>	Contém os limites de tamanho de inteiros do sistema.
<locale.h>	Contém protótipos de função e outras informações que permitem que um programa seja modificado para o local em que estiver sendo executado. A noção de local permite que o sistema de computação trate de diferentes convenções que expressam dados como datas, horas, valores monetários e números grandes em qualquer lugar do mundo.
<math.h>	Contém protótipos de função para funções da biblioteca matemática.
<setjmp.h>	Contém protótipos de função para funções que permitem evitar a sequência normal de chamada e de retorno de função.
<signal.h>	Contém protótipos de função e macros que lidam com diversas condições que podem surgir durante a execução do programa.
<stdarg.h>	Define macros que lidam com uma lista de argumentos para uma função cujo número e cujo tipo são desconhecidos.
<stddef.h>	Contém as definições comuns de tipo usadas pela C para realizar cálculos.
<stdio.h>	Contém protótipos de função para as funções da biblioteca-padrão de entrada/saída, e informações usadas por eles.
<stdlib.h>	Contém protótipos de função para conversões de números em texto e de texto em números, alocação de memória, números aleatórios e outras funções utilitárias.
<string.h>	Contém protótipos de função para funções de processamento de strings.
<time.h>	Contém protótipos de função e tipos para manipulação de hora e data.

Figura 5.6 ■ Alguns dos cabeçalhos da biblioteca-padrão.

Uso de bibliotecas

- Exemplo
 - Em um programa para fazer cálculos de seno, não há a necessidade de se criar funções com as rotinas de cálculo de seno
 - Tudo isso já está pronto em um arquivo de sistema (biblioteca math.h)

```
#include <stdio.h>
```

```
#include <math.h>
```

Forma de um programa em C

```
#include <stdio.h>
```

```
int main() {  
    printf("Meu primeiro programa.\n");  
    printf("Deu tudo certo.");  
    return 0;  
}
```

Forma de um programa em C

- Um programa em C é composto dos seguintes elementos:
 - Diretivas de compilação
 - Definições de tipos
 - Protótipos de funções
 - Funções
 - Comentários

Diretivas de compilação

- São os comandos que instruem o compilador realizar determinadas tarefas antes de iniciar a compilação.
- Também são chamadas de diretivas de pré-processamento.
- Comandos do pré-processador não necessariamente pertencem à linguagem C, mas são executados antes do programa ser compilado (pré-processamento).
- Eles começam sempre com # e são comumente usados para incluir o conteúdo de outro arquivo no programa (`#include`) e definir símbolos (`#define`), por exemplo.
- Não terminam com ponto e vírgula e tradicionalmente estão no início do arquivo fonte.

Definição de tipos

- Fortemente tipada: característica marcante onde toda variável possui um tipo explícito.
- A linguagem C possui 5 tipos básico de dados:
 - char
 - int
 - float
 - double
 - void

Funções em um programa em C

- A unidade básica de um programa em C é a **função**.
- Todo programa em C é uma coleção de uma ou mais funções, escrito em alguma ordem arbitrária.

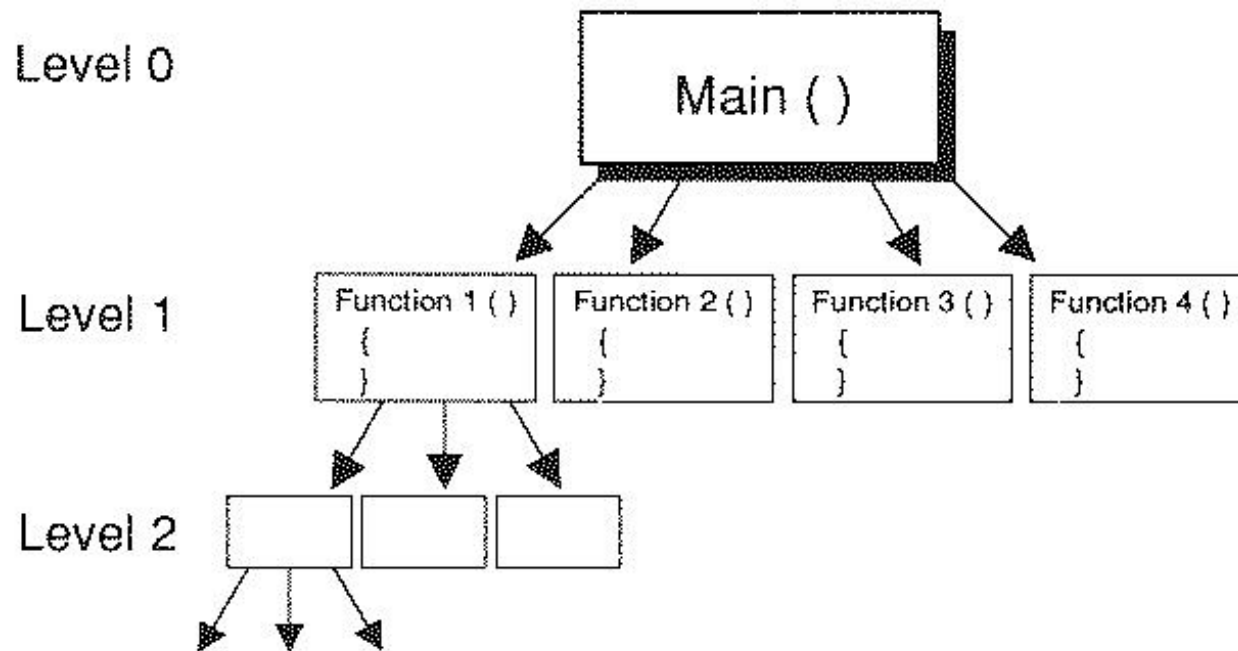


Figure 7.1. C Programs are built from functions.

Funções em um programa em C

- A função `main()` é sempre o ponto de partida de um programa em C.
- Somente uma função `main()` pode existir no programa.
- As chaves indicam o início e o fim de um bloco (nesse caso o programa).
- Em um programa escrito em C, o ponto e vírgula é o indicador de fim de uma declaração.

Protótipo de funções

- A declaração sem definição da função é conhecida como protótipo da função.
- O protótipo informa somente o tipo, o nome e, opcionalmente, os parâmetros formais da função:

[tipo] nome_funcao ([param_formais]);

Comentários em um programa em C

- Comentários são maneiras de inserir observações e lembretes em um programa sem afetar seu conteúdo.
- Comentários não têm um lugar fixo em um programa.
- Programas podem conter qualquer número de comentários sem perder velocidade.
- Isto é porque os comentários são removidos do código pelo compilador quando converte o programa fonte em código de máquina.
- Os melhores programadores comentam seus programas enquanto os escrevem, não depois.

Comentários em um programa em C

- Comentários são delimitados pelos seguintes caracteres: `/* ... */` ou `//`
- Exemplo

```
/* .....um.....  
.....conjunto de.....  
.....linhas.....*/
```

```
// apenas uma linha
```