

MCTA028-15: Programação Estruturada



Aula 1: Introdução a Linguagem C, Code::Blocks e Correção Automática no Moodle

Wagner Tanaka Botelho

wagner.tanaka@ufabc.edu.br / wagtanaka@gmail.com

Universidade Federal do ABC (UFABC)

Centro de Matemática, Computação e Cognição (CMCC)

Linguagens de Programação

Linguagens de Programação

- Atualmente existem **várias linguagens**:
 - Python, C, C++, Java, Haskell, Ruby, entre outras.
- Qual é a **melhor** linguagem?
 - Essa é uma pergunta muito **discutida** entre os **programadores** e cada um tem uma **opinião** diferente;
 - **C** é **melhor** do que **JAVA**?
 - **JAVA** é **pior** do que **Python**?

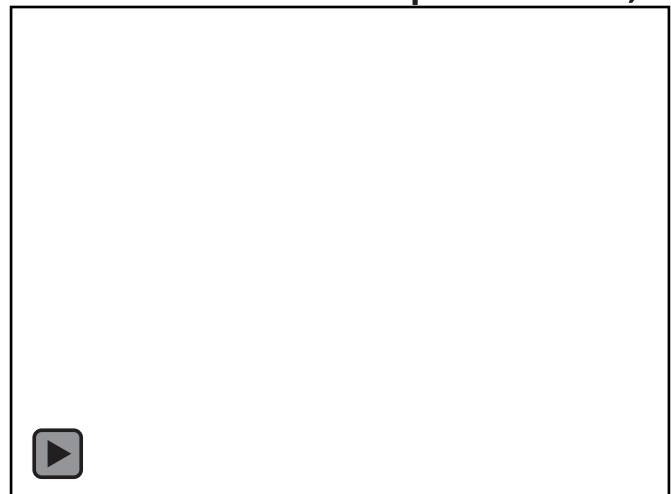
Linguagens de Programação

- Qual é a **minha** opinião?
 - Não gosto de **rotular** que uma é **melhor** do que a outra;
 - Acredito que tudo depende da **aplicação**.
- Muitos falam que JAVA, por ser uma **linguagem “mais lenta”**, não serve para controlar um robô. Será que isso é verdade?
 - Mas é claro que em **algumas aplicações**, **C** é mais **recomendado** do que **JAVA**;
 - **Python** está sendo bastante utilizada na **Robótica**.



2005-2008

AMD Athlon XP-M 1400 1.2GHz
256Mb RAM
HD: 40Gb



Linguagem C

Um Pouco de História

Linguagem C

- A **linguagem C** foi desenvolvida a partir das linguagens *Basic Combined Programming Language* (BCPL) e B:
- O **BCPL** foi criado, em 1967, por **Martin Richards** (Cambridge University);
- **Ken Thompson** modelou recursos da **linguagem B**, tendo como base o **BCPL**, para criar as primeiras versões do **UNIX**.

Código em BCPL: Imprimir "Olá Mundo!"

```
GET "LIBHDR"

LET START () BE
$(
  WRITES ("Olá Mundo!*N")
$)
```

<https://pt.wikipedia.org/wiki/BCPL>

Código em B: Imprimir "Hello, world!"

```
main() {
  putchar('Hello, world!*n');
}
```

<http://encurtador.com.br/fjtY5>

Martin Richards



<https://www.cl.cam.ac.uk/~mr10/>

Ken Thompson

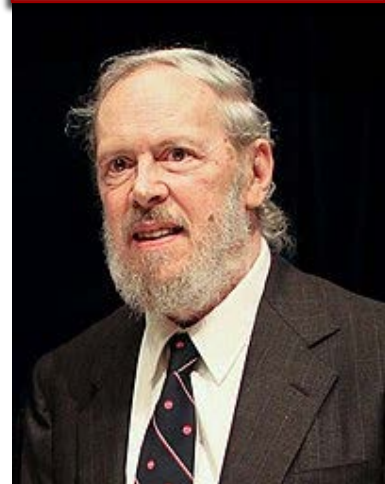


<http://encurtador.com.br/auxY4>

Linguagem C

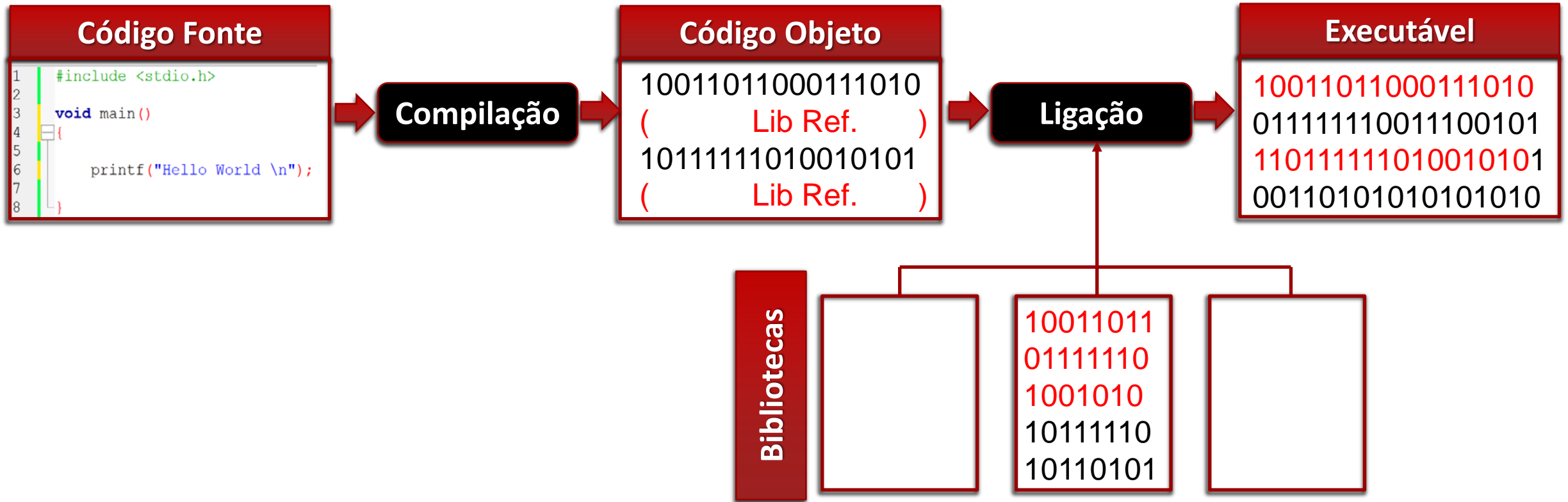
- A **linguagem C** foi desenvolvida, a partir da **B**, por **Dennis Ritchie**, do Bell *Laboratories*, e **implementada** em um computador DEC PDP-11, em **1972**:
 - Na realidade, a linguagem usa muitos dos **importantes conceitos** do **BCPL** e **B**.
- Inicialmente, o C ficou **conhecido** como a linguagem de desenvolvimento do **UNIX**:
 - Entretanto, hoje em dia, existem vários **Sistemas Operacionais** (SO) implementados em C:
 - É claro que a linguagem C não é somente utilizada para desenvolver SO.

Dennis Ritchie



https://pt.wikipedia.org/wiki/Dennis_Ritchie

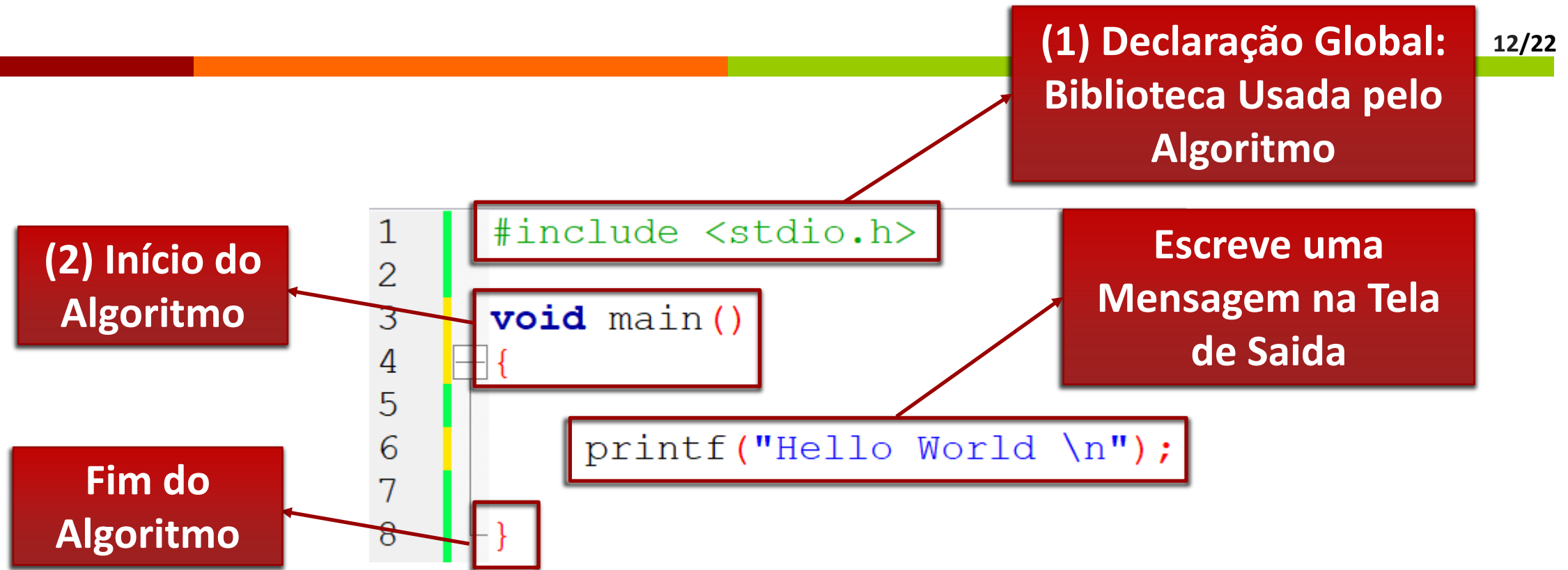
Compilação



+ **Código Objeto**: é a versão em **Linguagem de Máquina** do **Código Fonte**;

+ **Ligação**: os algoritmos em C possuem referências para **funções** definidas em **outro lugar**, como `#include<stdio.h>`. Por isso, deve-se ligar o **Código Objeto** com o **código das funções** que estão faltando para produzir o **Executável**

Esqueleto de um Algoritmo em C



(1)

`#include <nome_da_biblioteca>`: **biblioteca** é um **conjunto de funções** (pedaços de códigos) já implementados e que podem ser utilizados pelo programador;

`<stdio.h>`: função de **leitura do teclado** e **escrita na tela**.

Indentação do Algoritmo

Indentação do Algoritmo

➤ Consegue entender o seguinte código?

```
1  #include <stdio.h>
2  void main() {printf("Hello World \n");}
```

➤ O algoritmo indentado fica mais fácil, certo?

```
1  #include <stdio.h>
2
3  void main()
4  {
5
6      printf("Hello World \n");
7
8  }
```

A Função Printf

```
1  #include <stdio.h>
2
3  void main()
4  {
5
6      printf("Hello World \n");
7
8  }
```

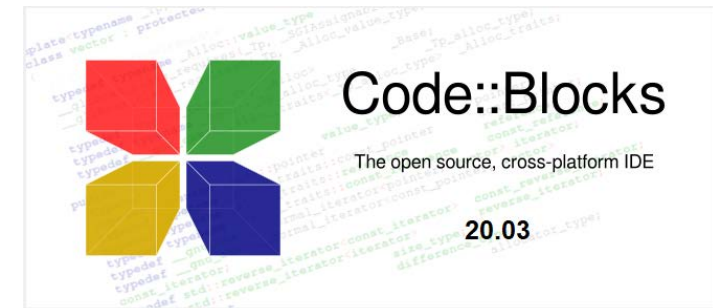
➔ É utilizada para **imprimir**, por exemplo, a frase *Hello World* na tela.

Sequência de Escape	Descrição
\n	Nova linha. Posiciona o cursor da dela no início da próxima linha.
\t	Tabulação horizontal. Move o cursor da tela para a próxima posição de tabulação.
\a	Alerta. Faz soar o alarme do sistema.
\\	Barra invertida. Insere um caractere de barra invertida em uma string.
\"	Aspas. Insere um caractere de aspas em uma string.

IDE – Code::Blocks

IDE

- Os **programas** em geral são desenvolvidos com a ajuda dos chamados **Ambientes Integrados de Desenvolvimento**, ou simplesmente *Integrated Development Environment* (**IDE**);
- Fornecem **recursos importantes** para **auxiliar** e **facilitar** o trabalho do programador:
 - Identificação de **palavras reservadas**;
 - Identificação de **erros de sintaxe**, **depuração** e **compilação**.
- Exemplos:
 - Eclipse (<http://www.eclipse.org/>);
 - Dev-C++ (<https://sourceforge.net/projects/orwellddevcpp/>);
 - **Code::Blocks** (<http://www.codeblocks.org/>);
 - Etc.



<http://www.codeblocks.org/>

- Features
- Screenshots
- Downloads
 - Binaries
 - Source
 - SVN
- Plugins
- User manual
- Licensing
- Donations

Quick links

- FAQ
- Wiki
- Forums
- Forums (mobile)
- Nightlies
- Ticket System
- Browse SVN
- Browse SVN log



• Mac OS X

NOTE: For older OS'es use older releases. There are releases for many OS version and platforms on the **Sourceforge.net** page.

NOTE: There are also more recent *nightly builds* available in the **forums** or (for Ubuntu users) in the **Ubuntu PPA repository**. Please note that we consider nightly builds to be *stable*, usually.

NOTE: We have a **Changelog for 20.03**, that gives you an overview over the enhancements and fixes we have put in the new release.

NOTE: The default builds are 64 bit (starting with release 20.03). We also provide 32bit builds for convenience. The 32 bit packages have either the same 64 bit compiler as the 64 bit version included or (which is recommended) a 32 bit version of the compiler.



Windows XP / Vista / 7 / 8.x / 10:

Inclui o Code::Blocks, compilador GCC e o debugger GDB da MinGW.

File	Date	Download from
codeblocks-20.03-setup.exe	29 Mar 2020	FossHUB or Sourceforge.net
codeblocks-20.03-setup-nonadmin.exe	29 Mar 2020	FossHUB or Sourceforge.net
codeblocks-20.03-nosetup.zip	29 Mar 2020	FossHUB or Sourceforge.net
codeblocks-20.03mingw-setup.exe	29 Mar 2020	FossHUB or Sourceforge.net
codeblocks-20.03mingw-nosetup.zip	29 Mar 2020	FossHUB or Sourceforge.net
codeblocks-20.03-32bit-setup.exe	02 Apr 2020	FossHUB or Sourceforge.net
codeblocks-20.03-32bit-setup-nonadmin.exe	02 Apr 2020	FossHUB or Sourceforge.net
codeblocks-20.03-32bit-nosetup.zip	02 Apr 2020	FossHUB or Sourceforge.net
codeblocks-20.03-32bit-mingw-32bit-setup.exe	29 Dec 2020	FossHUB or Sourceforge.net
codeblocks-20.03-32bit-mingw-32bit-nosetup.zip	29 Dec 2020	FossHUB or Sourceforge.net
codeblocks-20.03-32bit-mingw-64bit-setup.exe	02 Apr 2020	FossHUB or Sourceforge.net
codeblocks-20.03-32bit-mingw-64bit-nosetup.zip	02 Apr 2020	FossHUB or Sourceforge.net

Correção Automática no Moodle

Correção Automática no Moodle

- Os **exercícios** serão disponibilizados no Moodle:
 - Até sexta-feira às 8h;
 - Prazo de entrega é de **uma semana**.
- Você deve **assistir o vídeo**, disponível na **Semana 01 – Prática**, que explica o funcionamento do sistema.

Referências

- Slides do Prof. Luiz Rozante;
- SALES, André Barros de; AMVAME-NZE, Georges. Linguagem C: roteiro de experimentos para aulas práticas. 2016;
- BACKES, André. Linguagem C Completa e Descomplicada. Editora Campus. 2013;
- SCHILDT, Herbert. C Completo e Total. Makron Books. 1996;
- DAMAS, Luís. Linguagem C. LTC Editora. 1999;
- DEITEL, Paul e DEITEL, Harvey. C Como Programar. Pearson. 2011.