

# Introdução à Programação

*Aula 02*

**Marcos Silvano Almeida**

*[marcoossilvano@professores.utfpr.edu.br](mailto:marcoossilvano@professores.utfpr.edu.br)*

Departamento de Computação

UTFPR Campo Mourão

# Roteiro

- Linguagem C
- Algoritmo
- Primeiro programa
- Imprimindo textos

# Linguagem C

- Criada em 1972/1973
  - Dennis Ritchie (1941-2011)
  - Bell Labs
- Projetada para implementar utilidades para o **Unix**
  - Kernel implementado em C
- Unix dá origem a uma grande família de Sistemas Operacionais
- Características
  - Flexibilidade, produtividade e performance
  - Extremamente **popular**: muito fácil encontrar materiais e ferramentas
- Várias outras linguagens foram influenciadas por C
  - Java, C#, PHP, JavaScript, Lua



Dennis Ritchie

1969

1971 to 1973

1974 to 1975

1978

1979

1980

1981

1982

1983

1984

1985

1986

1987

1988

1989

1990

1991

1992

1993

1994

1995

1996

1997

1998

1999

2000

2001 to 2004

2005

2006 to 2007

2008

2009

2010

2011

2012 to 2015

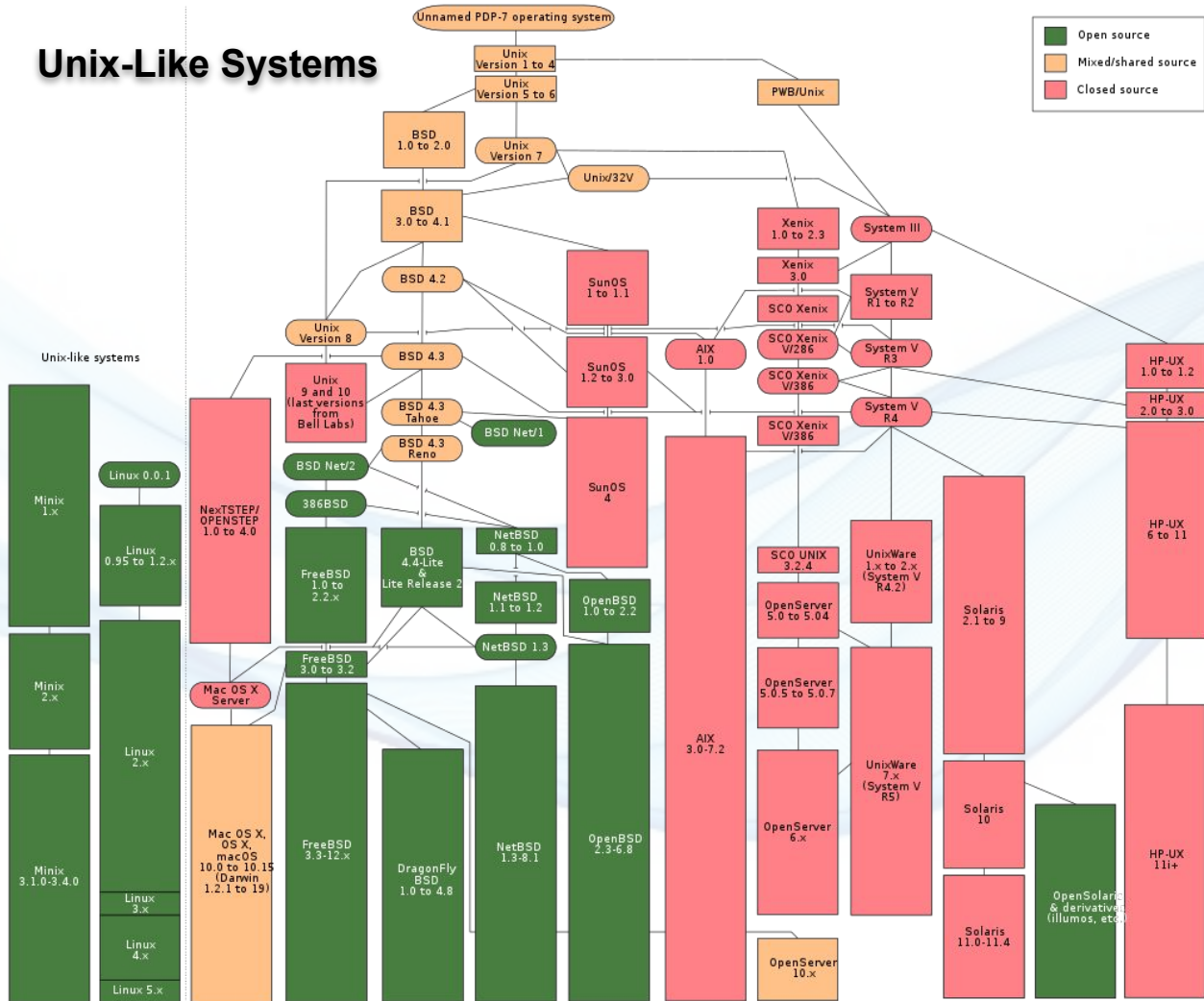
2016

2017

2018

2019

# Unix-Like Systems



# Open Source Software

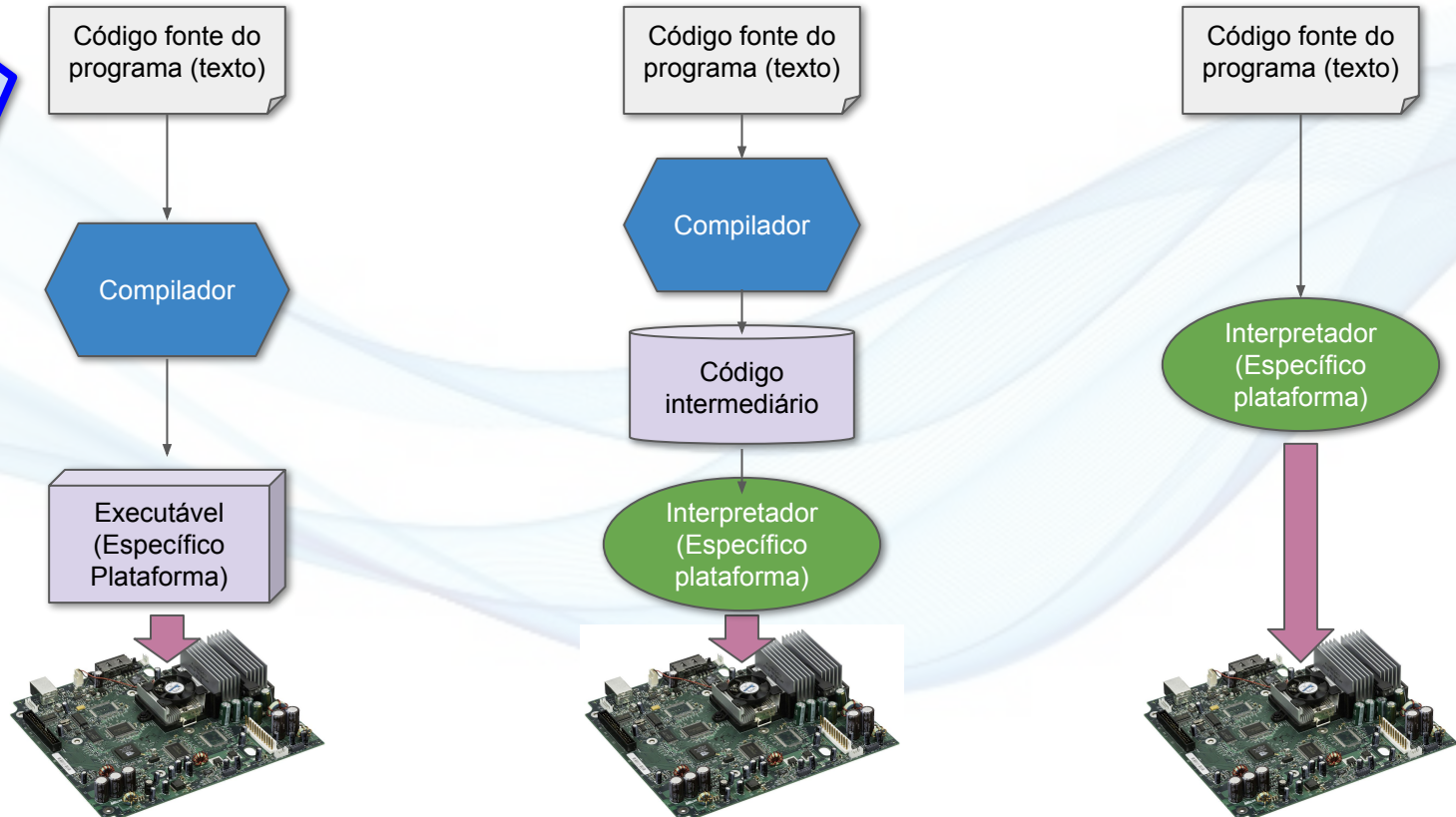


- Código aberto, pode ser modificado
  - Grande maioria é gratuito
  - Sistemas Operacionais open source são os mais utilizados no mundo
  - Ferramentas utilizadas em muitos produtos
- Linux, o mais comum
  - Android é Linux modificado
  - Smart TVs
    - Samsung: Tizen
    - LG: webOS
    - Panasonic: Firefox OS
  - Roteadores de Internet
- FreeBSD
  - Playstation 3, 4 e 5
  - MacOS, iOS



# Geração e execução do programa

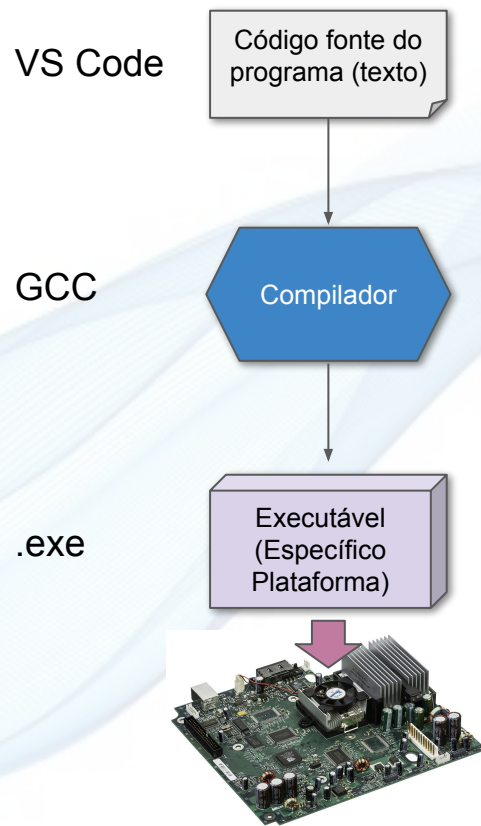
Processo que utilizaremos





# Preparando o ambiente

- Editor de texto/código
  - Visual Studio Code (VSCode)
- Compilador GCC
  - Geralmente contido no Linux
  - Necessário instalar no Windows
    - Pacote MinGW ou MinGW-W64
- Alternativa
  - <https://repl.it/languages/c>



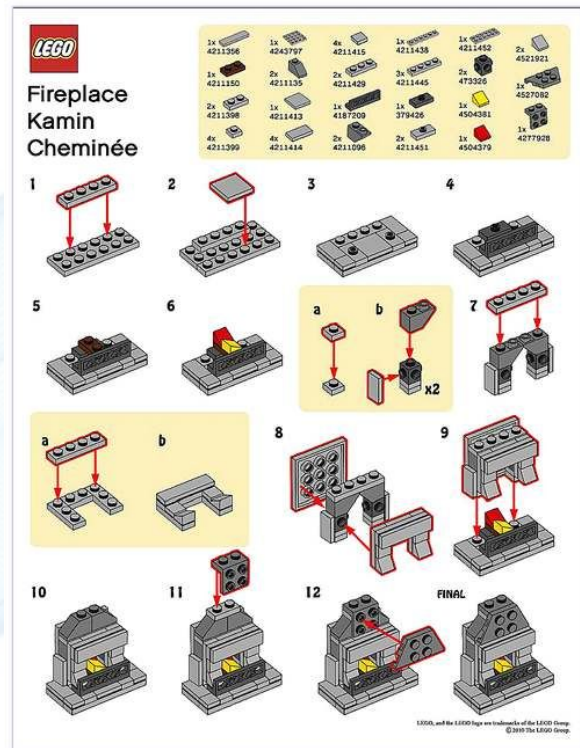


Algoritmo



# Algoritmo

- Algoritmo significa a execução de passos para se realizar uma tarefa
  - O termo é muito antigo → Babilônios 2.500 AC
- Exemplos
  - Algoritmos aritméticos: adição, subtração, divisão, multiplicação, decomposição em fatores primos, ...
  - Chegar a um local por indicação
  - Receita culinária
  - Manual de instruções de lego
  - Atividades de um funcionário de fábrica
- Em linguagem de programação, escrevemos o código **linha a linha**



# Linguagem C

- Primeiro Programa

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

Indica que estamos incluindo os recursos da biblioteca “**Standard Input/Output**”, pois queremos realizar saída (imprimir dados na tela)

```
int main() {
```

```
    printf("Primeiro programa em C\n");
```

printf( ) permite imprimir texto na saída padrão (terminal). Texto deve estar em “aspas”.  
OBS: / e “ possuem significado especial.

```
    return 0;
```

```
}
```

“main( )” deve retornar ao sistema um número inteiro (**int**). Portanto, ao final do programa, retornamos 0 (indica término com sucesso).

Bloco principal “main” do programa. Blocos iniciam em { e termina em }. O código deve ser escrito dentro do bloco.

# Linguagem C

- No VS Code, crie um arquivo chamado **prog01.c** e digite:

```
#include <stdio.h>

int main() {
    printf("Primeiro programa em C\n");
    return 0;
}
```

1. Visualizar terminal no VS Code: View ⇒ Terminal

2. Compilar o programa pelo terminal:

**\$ gcc prog01.c -o prog01.exe** << nome do fonte e executável

3. Executar programa:

**\$ prog01.exe** << chama executável gerado

**\$ ./prog01** << Linux (comum não ter extensão)

# Modificações do primeiro programa

- Imprimir múltiplas linhas

*PRIMEIRO PROGRAMA C:*

*Printf permite escrever texto na saída padrão.*

*Printf é chamado de função em linguagem de programação.*

- Imprimir uma caixa

```
+-----+  
|       |  
|  The  |  
|  Box  |  
|       |  
+-----+
```

# Linguagem C

- Escrevendo múltiplas linhas:

```
printf("C provides constructs that map machine instructions\n");  
printf("It was used in apps previously coded in assembly language.\n");
```

👉 forma mais comum

```
printf("C provides constructs that map machine instructions\n"  
      "It was used in apps previously coded in assembly language.\n");
```

```
printf("C provides constructs that map machine instructions\n \  
      It was used in apps previously coded in assembly language.\n");
```

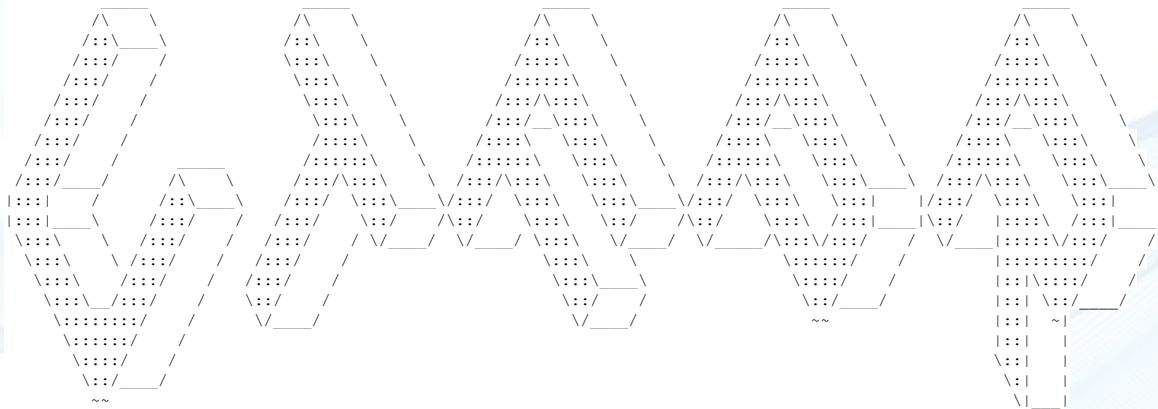
# Sequências de Controle para printf()

Sequência	Descrição	Comportamento
\n	Quebra linha	Posiciona o cursor no início da próxima linha.
\f	Quebra linha continuada	Posiciona o cursor na mesma coluna da próxima linha.
\r	Retorno do cursor	Posiciona o curso no início da linha atual.
\t	Tab horizontal	Move o cursor até o próximo “passo” de tabulação.
\a	Alerta	Produce um alerta sonoro ou visual.
\b	Backspace	Apaga caractere anterior, voltando o cursor em uma posição.
\\	Contra barra	Permite escrever o caractere \. (reservado na linhagem)
\'	Aspas simples	Permite escrever o caractere '. (reservado na linhagem)
\"	Aspas duplas	Permite escrever o caractere ". (reservado na linhagem)



# Modificações do primeiro programa

- Imprimir TEXT ART



>> <https://patorik.com/software/taag/#p=display&f=Alpha&t=UTFPR>