Linguagem de Programação I

Aula 2 – Introdução à Linguagem de Programação C

Prof Ana Carolina Sokolonski

Bacharelado em Sistemas de Informação
Instituto Federal da Bahia – Campus Feira de Santana

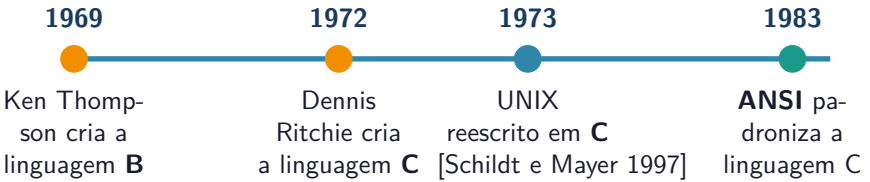
2026

Agenda

Breve Histórico

da Linguagem C

Linha do Tempo da Linguagem C



Implementações incompatíveis entre 1973 e 1983 motivaram a padronização da ANSI.

Aplicações Desenvolvidas em C

Sistemas Operacionais

UNIX e todas as suas variações (Linux, macOS, BSD...)

Planilhas

Lotus 1-2-3 e Microsoft Excel

Banco de Dados

dBase III, IV, e Microsoft Access (SGBDs)

Aplicações Gráficas

Efeitos especiais de filmes como:

Star Trek, Star Wars
Final Fantasy

Por que C?

Velocidade próxima ao Assembly, com portabilidade e legibilidade muito superiores.

Vantagens

da Linguagem de Programação C

Por que aprender C?

▪ Desempenho

Programas organizados e concisos com **alta velocidade de execução** e baixo consumo de memória — próximo ao Assembly.

▪ Portabilidade

Código portável entre diferentes **máquinas e sistemas operacionais**. A mesma fonte roda em Linux, Windows e macOS.

▪ Interação com o SO

Total acesso e controle sobre o **Sistema Operacional**.

▪ Alto e Baixo Nível

Alia características de linguagens de **alto e baixo nível**, dando liberdade ao programador para atuar próximo ao hardware quando necessário.

▪ Código Compacto

Apenas **42 comandos ANSI** — código rápido e compacto comparado a outras linguagens de complexidade semelhante.

▪ Dados Estruturados

Suporte a dados compostos em formato estruturado.

Identidade da Linguagem C

A linguagem C é compilada, fortemente tipada, de baixo nível e imperativa

Palavras Reservadas

da Linguagem C

Nomes Reservados (42 comandos)

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

Geração de Código

em C

Compilada vs. Interpretada

▪ Linguagem Compilada (C)

Lê **todo o código-fonte** de uma vez e gera o código objeto (linguagem de máquina). A compilação ocorre apenas uma vez; o executável roda diretamente.

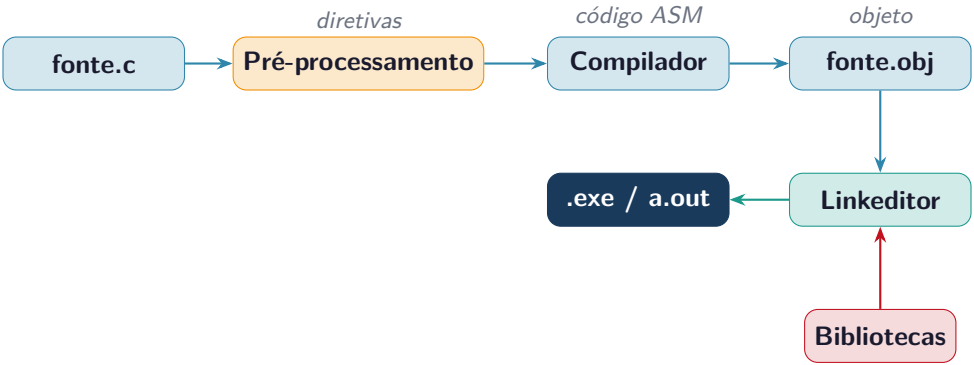
▪ Linguagem Interpretada

Lê, traduz e executa o código-fonte **a cada execução**, instrução por instrução. Mais lenta, porém mais fácil de depurar.

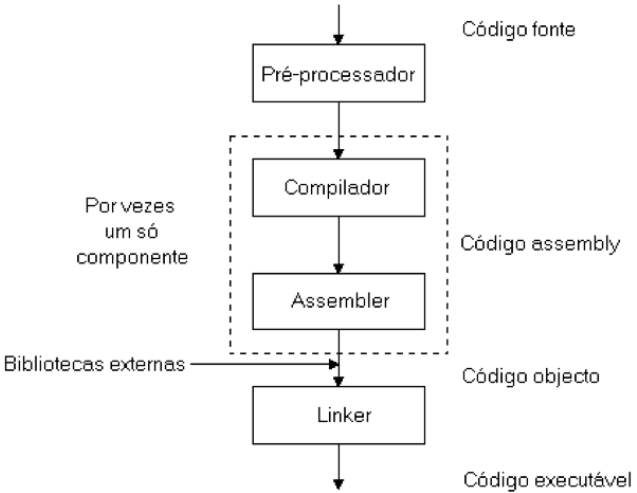
▪ Processo de compilação em C

O arquivo **.c** (source code) é compilado; se não houver erros, gera-se um **.obj**. Em seguida, a **linkedição** une o objeto às bibliotecas, produzindo o executável (**.exe** no Windows, sem extensão no Linux/macOS).

Fluxo de Compilação em C



Fluxo de Compilação em C



Bibliotecas

da Linguagem C

Bibliotecas da Linguagem C

O que são?

Conjuntos de funções prontas para realizar tarefas específicas. A **biblioteca padrão ANSI** fornece as funções básicas da linguagem.

Inclusão de biblioteca

Para sistemas (biblioteca padrão):

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

```
#include <stdlib.h>
```

Biblioteca própria

Para arquivos locais do projeto:

```
#include "minha_lib.h"
```

stdio.h
entrada/saída

stdlib.h
utilitários gerais

math.h
funções matemáticas

string.h
manipulação de strings

Primeiro Programa

na Linguagem C

Hello, World!

```c
/* Primeiro Programa em C */
#include <stdio.h>

void main(){
printf("Meu primeiro programa em C\n");
}
```

Estrutura básica

- `#include` — inclui biblioteca
- `int main()` — função principal
- `printf()` — imprime na tela
- `return 0;` — encerra o programa

Hello, World! Comentado

```
/* Primeiro Programa em C */ //comentários
#include <stdio.h> /*biblioteca de E/S */

void main() /*função principal – início do programa*/
{   /*marca início da função*/
    printf("Meu primeiro programa em C\n");
  /*função para escrever na tela*/
} /*marca fim da função*/
```

Observações Importantes

Regras obrigatórias

- Todo comando termina com `;`
- Variáveis devem ser **declaradas antes** de serem usadas
- Blocos usam `{}` — exceto comando único em seleção/repetição
- C é ***case sensitive***: `printf` ≠ `Printf`

Ciclo de desenvolvimento

1. Editar o arquivo `.c`
2. Compilar
3. Corrigir erros (se houver)
4. Executar
5. **Sempre recompilar** após qualquer alteração

Ambientes de Desenvolvimento

Linux

Qualquer editor de texto + GCC:

`vi`, `nano`, `gedit`

Windows

IDE recomendada:

Dev-C ou **Dev-C++**

(inclui compilador MinGW)

macOS

BBEdit ou

Xcode + clang

Sem instalação? Use compiladores online

OnlineGDB: `www.onlinegdb.com/online_c_compiler` **Programiz**: `www.programiz.com/c-programming/online-compiler`

Compilando no Linux

com GCC

▪ Compilar e executar

Compilar (gera `a.out` por padrão):

```
cc fonte.c
```

Compilar com nome específico:

```
cc fonte.c -o nomeexec
```

Executar:

```
./nomeexec
```

Para compilar usando o compilador GCC na linha de comando do LINUX: (verifica a sintaxe)

>> cc *fonte.c* <*ENTER*>

Para compilar e gerar executável:

>> cc *fonte.c* –o *nomeexec* <*ENTER*>

Para executar o arquivo gerado:

>> *./nomeexec* <*ENTER*>

OBS:

Se o nome do executável não for informado o *default* será *a.out*

GCC — Flags de Compilação

Flags principais

- `-g` monta e linkedita código-objeto
- `-o` compila e linkedita um fonte
- `-c` compila sem linkeditar (gera `.obj`)
- `-E` apenas pré-processa o fonte
- `-S` compila sem linkeditar (gera Assembly)

Flags de otimização

- `-O0` sem otimização
- `-O1` otimização leve
- `-O2` otimização moderada
- `-O3` otimização máxima
- `-Os` otimiza para tamanho (semelhante a -O2, reduz tamanho do executável)

Usando um

Compilador Online

GDB Online


onlinegdb.com
Bookmarks
Gmail
Maps
Google
Scratch - Educators
overleaf
Online C Compiler
All Bookmarks
OnlineGDB beta
online compiler and debugger for c/c++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login
Learn Python with KodeKloud
Run
Debug
Stop
Share
Save
{ } Beautify
Language
-- select --
source code
/***
Welcome to GDB Online.
GDB online is an online compiler and debugger tool for C, C++, Python, PHP, Ruby,
C#, OCaml, VB, Perl, Swift, Prolog, Javascript, Pascal, COBOL, HTML, CSS, JS
Code, Compile, Run and Debug online from anywhere in world.
***/
#include <stdio.h>
int main()
{
printf("Hello World");
return 0;
}

Referências I

SCHILDT, H.; MAYER, R. *C completo e total*. Pearson Universidades, 1997. ISBN 9788534605953. Disponível em: <https://books.google.com.br/books?id=PbI0AAAACAAJ>.