

# Conceitos de Computação

Disciplina – Curso



Prof. Daniel Saad Nogueira  
Nunes

Instituto Federal de Brasília, Câmpus  
Taguatinga



# Sumário

---

- 1 Introdução
- 2 Organização
- 3 Computação Pessoal
- 4 Internet
- 5 Linguagens
- 6 Ambiente



# Sumário

---

## 1 Introdução



# Introdução

---

- Qual a noção moderna de um computador?



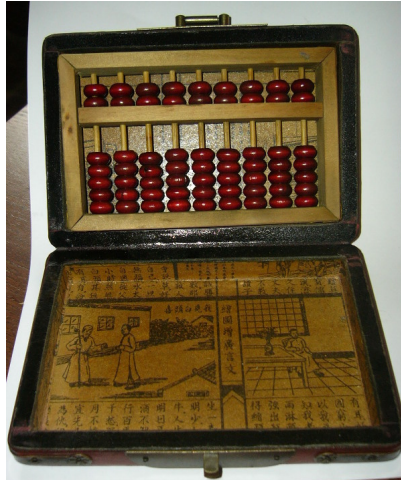
# Introdução

---

- Um computador é um dispositivo capaz de realizar cálculos e tomar decisões lógicas com uma velocidade milhões ou mesmo bilhões de vezes mais rápida do que os seres humanos.
- Uma pessoa fazendo os mesmos cálculos demoraria vários anos para chegar no mesmo resultado.
- Qual é o papel dos computadores na história?



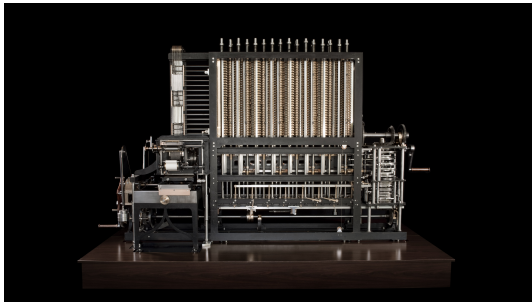
# Computadores





# Computadores

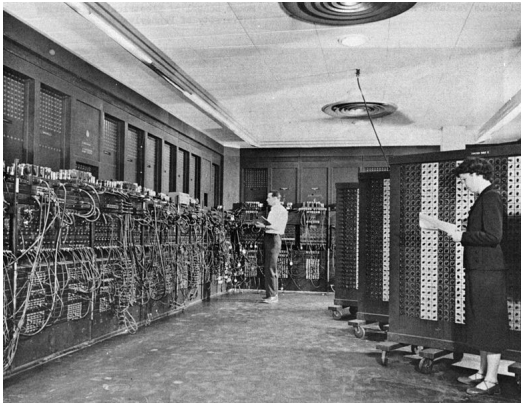
---





# Computadores

---







# Computadores

---





# Computadores





# Computadores

---





# Computadores

---

- Os computadores processam dados através da execução de programas.
- Os programas por sua vez consistem de várias instruções.
- Programadores especificam quais as instruções de um programa a serem seguidos por um computador.



# Computadores

---

- Hardware: vários dispositivos físicos que constituem um sistema computacional.
  - ▶ Teclado.
  - ▶ Telas.
  - ▶ Mouse.
  - ▶ CPU.
  - ▶ Memória.
  - ▶ Discos.
  - ▶ ...



# Computadores

---

- Software: programas executados em um computador.
  - ▶ Editores de texto.
  - ▶ Sistema Operacional.
  - ▶ Planilhas.
  - ▶ Video players.
  - ▶ Jogos.
  - ▶ ...



# Computadores

---

- Custo do hardware diminuiu drasticamente. Os computadores se tornaram uma utilidade cotidiana.
- Custo do desenvolvimento vem crescendo: programas cada vez mais complexos para diferentes plataformas.
- Este curso foca no desenvolvimento de software.



# Desenvolvimento

---

- Veremos as técnicas por trás de uma boa programação estruturada que potencialmente podem reduzir os custos de desenvolvimento e acelerar o processo de lançamento de aplicações poderosas e de alto nível.
- Mas para projetar um bom software, precisamos entender o hardware minimamente.





# Sumário

---

## 2 Organização



# Organização de Computadores

---

- Independentemente da aparência física, computadores podem ser divididos em unidades lógicas:
  - ▶ Unidade de entrada.
  - ▶ Unidade de saída.
  - ▶ Unidade de memória.
  - ▶ Unidade Lógica e Aritmética.
  - ▶ Unidade de Controle.
  - ▶ CPU.
  - ▶ Unidade de armazenamento secundário.



# Organização de Computadores

---

## Unidade de Entrada

- Responsável por obter informação de dispositivos de entradas e disponibilizá-las para outros dispositivos.
- Dispositivos de entrada: teclado, mouse, dispositivos USB, CDs, discos rígidos, Internet,...



# Organização de Computadores

---

## Unidade de Saída

- Recebe a informação processada e disponibiliza a mesma em um formato apropriado para os dispositivos de saída.
- Dispositivos de saída: monitores, caixas de som, internet, discos, CDs, USB, ...



# Organização de Computadores

---

## Unidade de Memória Primária

- Provê um meio de armazenamento temporário e rápido.
- Memória Volátil.
- Informações não temporárias devem ser armazenadas em dispositivos de armazenamento secundário.



# Organização de Computadores

---

## Unidade Lógica e Aritmética

- Parte da CPU que desempenha operações lógicas e aritméticas.
- Adições, subtrações, multiplicações, comparações,...



# Organização de Computadores

---

## Unidade de Controle

- Parte da CPU que desempenha funções de controle.
- Determina a próxima instrução a ser executada, busca a próxima instrução de memória, instrui a ULA a fazer determinada operação, instrui o armazenamento do dado da ULA para outro dispositivo,...



# Organização de Computadores

---

## Unidade de Processamento Central (CPU)

- Composto da ALU, Unidade de controle e registradores.
- Computadores modernos possuem diversas CPUs.





# Organização de Computadores

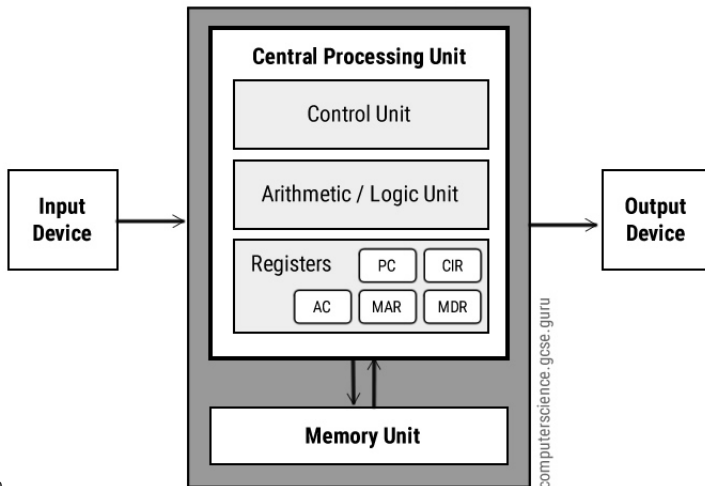
---

## Unidade de Memória Secundária

- Provê um meio de armazenamento persistente e permanente.
- Mais lenta que a memória primária.
- Discos, CDs DVDs, Pendrives, ...



# Organização de Computadores





# Sumário

---

## 3 Computação Pessoal



## Computação Pessoal, Distribuída e Cliente/Servidor

---

- Em 77, a Apple popularizou os computadores pessoais.
- Com a entrada da IBM com no ramo da computação pessoal com o IBM PCem 81, computadores começaram a ficar mais baratos e acessíveis.
- Com a popularização das LANs, foi dado origem à computação distribuída dentro de organizações empresariais.
- Em vez de toda a computação ser feita em um computador central, ela poderia ser feita nos computadores pessoais.
- A comunicação entre os computadores era feita via rede.



## Computação Pessoal, Distribuída e Cliente/Servidor

---

- Devido ao aumento de poder computacional e facilidade de comunicação, surgiram diferentes paradigmas ao longo do tempo.
- Um paradigma bastante utilizado nos dias de hoje é o Cliente/Servidor.
- Nele, temos os servidores que oferecem serviços aos clientes.
- Arquivos, bases de dados, servidores WEB, ...



# Sumário

---

## 4 Internet



# Internet

---

- A Internet, ou rede mundial de computadores, remete à década de 60 quando projetos financiados pelo departamento de defesa dos EUA e por outros órgãos de outros países trabalhavam em redes de comutação por pacotes.
- ARPANET, CYCLADES, NPL Network dentre outras redes surgiram neste tempo.



# Internet

---

- Com a introdução da WEB, ficou possível pesquisar e localizar documentos multimídia de maneira bem simples.
- Isto ocasionou toda uma mudança de paradigmas, na qual aplicações são escritas de modo a se comunicarem com outras.
- Possibilitou severas mudanças de comunicação e no próprio funcionamento da sociedade.





# Internet

---

- Hoje é possível:
  - ▶ Pesquisar preços on-line e fazer compras.
  - ▶ Propagar informações sobre o seu negócio na rede.
  - ▶ Participar de grupos de discussões e redes sociais.
  - ▶ Comunicação instantânea através de aplicativos.
  - ▶ ...



# Sumário

---

## 5 Linguagens



# Linguagens

---

- Programadores escrevem programas via **Linguagens de Programação**.
- Algumas linguagens conseguem ser interpretadas diretamente por computadores enquanto outras requerem traduções.
- Hoje em dia: milhares de linguagens de programação.
- Geralmente divididas em:
  - ▶ Linguagens de máquina.
  - ▶ Linguagens assembly.
  - ▶ Linguagens auto-nível.



# Sumário

---

## 5 Linguagens

- Linguagens de Máquina
- Linguagens Assembly
- Linguagens Alto Nível
- A Linguagem C
- A Biblioteca Padrão do C
- C++
- Java
- FORTRAN, COBOL, Pascal e Ada
- BASIC, VB, Visual C++, C# e .NET



# Linguagens de Máquina

---

- Os computadores entendem diretamente a sua linguagem de máquina.
- Definida pela sua arquitetura.
- Também referenciada como código objeto.
- Consistem de 0s e 1s.
- Sequências específicas de 0s e 1s correspondem à instruções a serem executadas.



# Linguagens de Máquina

---

- Geralmente estas linguagens são dependentes de arquitetura.
- Uma linguagem de máquina específica só pode ser utilizada em um computador específico ou em uma família de computadores específica.
  - ▶ Temos uma infinidade de dispositivos hoje em dia, cada um com decisões de projeto de hardware diferentes.


$$2 + 2 = 4$$

001000010000 11110000  
101000010000 11110000  
101000010000 11110000  
THE UNIVERSITY OF ALABAMA  
TALLAHASSEE, FLORIDA 32307  
ALABAMA POWER  
TALLAHASSEE, FLORIDA 32307  
ALABAMA POWER

```

0010000010000 11110010
0010000011000 11110010
0010000010011 11111100
0010000010011 11011100
0010100011000 11110010
11110010 01110111 0111
11110010 01111011 1101
11111100 01011010 1101
11011100 01011010 1111
11110010 01110011 1101

```



# Linguagens de Máquina

---

- O desenvolvimento diretamente em linguagem de máquina era lento, tedioso e propenso a erros.
- Em vez de utilizar diretamente números binários ou hexadecimais, programadores começaram a usar abreviações em inglês para cada sequência de dígitos que correspondia a uma instrução.
- Estas abreviações formaram a base das chamadas **Linguagens Assembly**.





# Sumário

---

## 5 Linguagens

- Linguagens de Máquina
- **Linguagens Assembly**
- Linguagens Alto Nível
- A Linguagem C
- A Biblioteca Padrão do C
- C++
- Java
- FORTRAN, COBOL, Pascal e Ada
- BASIC, VB, Visual C++, C# e .NET



# Linguagens Assembly

---

- As linguagens assembly tornaram-se bem mais legíveis e amigáveis para humanos.
- No entanto, elas eram incompreensíveis para os computadores.
- Programas de tradução, denominados **assemblers**, foram desenvolvidos para traduzir um programa escrito em Assembly para o seu equivalente em linguagem de máquina.



# Linguagens Assembly

i = j + k;	1	ILOAD j // i = j + k	0x15 0x02
if (i == 3)	2	ILOAD k	0x15 0x03
k = 0;	3	IADD	0x60
else	4	ISTORE i	0x36 0x01
j = j - 1;	5	ILOAD i // if (i < 3)	0x15 0x01
	6	BIPUSH 3	0x10 0x03
	7	IF_ICMPEQ L1	0x9F 0x00 0x0D
	8	ILOAD j // j = j - 1	0x15 0x02
	9	BIPUSH 1	0x10 0x01
	10	ISUB	0x64
	11	ISTORE j	0x36 0x02
	12	GOTO L2	0xA7 0x00 0x07
	13 L1:	BIPUSH 0 // k = 0	0x10 0x00
	14	ISTORE k	0x36 0x03
	15 L2:		

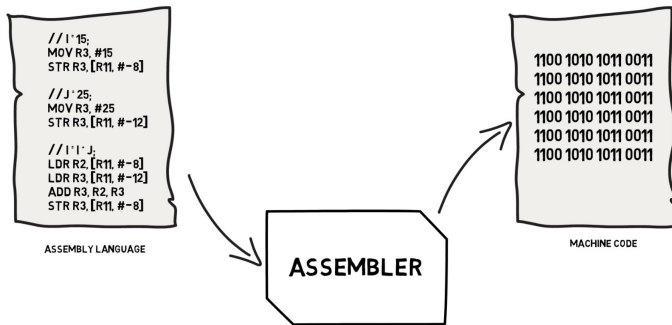
(a)

(b)

(c)



# Linguagens Assembly



ANDROID AUTHORITY



# Linguagens Assembly

---

- Com a utilização das linguagens assembly o desenvolvimento tornou-se menos moroso e menor propenso a erros.
- Mais ainda era difícil construir programas de maneira simples.
- Para aumentar a facilidade do desenvolvimento, **Linguagens de Alto Nível** foram criadas.



# Linguagens Alto Nível

---

- Linguagens de Alto Nível possibilitaram escrever programas utilizando muito menos primitivas.
- Mais próximas do inglês e da notação matemática utilizada no dia a dia.



# Linguagens Alto Nível

---

- Compiladores: traduzem programas em linguagens de alto nível para linguagem de máquina.
- Geralmente o processo de compilação pode requerer uma razoável quantidade de tempo e processamento.
- Interpretação: executa programas alto nível diretamente sem precisar compilar (geralmente mais lentos)
- Exemplos de Linguagens Alto Nível: C, C++, Java, Python, Ruby, Smalltalk, Haskell, Prolog, C#, Visual Basic, Pascal, ...



# Compiladores

High-level Language

```
temp = v[k];  
v[k] = v[k+1];  
v[k+1] = temp;
```

```
TEMP = V(K)  
V(K) = V(K+1)  
V(K+1) = TEMP
```

C/Java Compiler

Fortran Compiler

Assembly Language

```
lw $t0, 0($2)  
lw $t1, 4($2)  
sw $t1, 0($2)  
sw $t0, 4($2)
```

MIPS Assembler

Machine Language

```
0000 1001 1100 0110 1010 1111 0101 1000  
1010 1111 0101 1000 0000 1001 1100 0110  
1100 0110 1010 1111 0101 1000 0000 1001  
0101 1000 0000 1001 1100 0110 1010 1111
```





# Sumário

---

## 5 Linguagens

- Linguagens de Máquina
- Linguagens Assembly
- Linguagens Alto Nível
- **A Linguagem C**
- A Biblioteca Padrão do C
- C++
- Java
- FORTRAN, COBOL, Pascal e Ada
- BASIC, VB, Visual C++, C# e .NET



# A Linguagem C

---

- A linguagem C evoluiu de duas linguagens anteriores: BCPL e B.
- BCPL: desenvolvida em 1967 por Martin Richards como uma linguagem para escrever softwares de sistemas e compiladores.
- B: desenvolvida por Ken Thompson em 70 após ser inspirada no BCPL. Utilizada no desenvolvimento de versões preliminares do UNIX.



# A Linguagem C

---

- A linguagem C evoluiu da linguagem B.
- Desenvolvida por Dennis Ritchie nos laboratórios da Bell e implementada inicialmente em um computador DEC PDP-11 em 1972.
- Utiliza vários conceitos do B e do BCPL e adiciona novos conceitos como tipagem de dados e outras características poderosas.



# A Linguagem C

---

- A linguagem C ficou conhecida como a linguagem de desenvolvimento do UNIX.
- Basicamente os principais sistemas operacionais modernos são escritos em C ou C++.



# A Linguagem C

---

- C é considerada uma linguagem portátil e está disponível na maioria das plataformas e computadores.
- Com um pouco de cuidado, é possível fazer programas em C que, quando compilados, rodem em qualquer computador.



# A Linguagem C

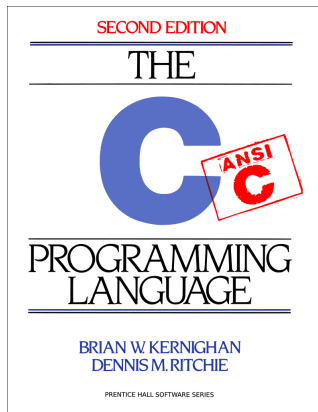
---

- No fim da década de 70, C evoluiu no que é agora referenciado como o C “tradicional”.
- Em 78 o livro “The C Programming Language” por kernighan e Ritchie chamou mais atenção ainda para a linguagem.
- Um dos livros mais conhecidos em Ciência da Computação.



# A Linguagem C

---





# A Linguagem C

---

- A rápida expansão do C em diferentes plataformas levou a muitas variações que eram similares, mas incompatíveis.
- Isso era um problema para programadores que queriam usar o mesmo código em várias plataformas.
- Ficou claro que um Padrão para C era necessário.
- Surge o ANSI C.
- C99: padrão ANSI revisado que refina e expande as capacidades do C.
- Utilizaremos o ANSI C.





# Sumário

---

## 5 Linguagens

- Linguagens de Máquina
- Linguagens Assembly
- Linguagens Alto Nível
- A Linguagem C
- A Biblioteca Padrão do C
- C++
- Java
- FORTRAN, COBOL, Pascal e Ada
- BASIC, VB, Visual C++, C# e .NET



# A Biblioteca Padrão do C

---

- Um programa C consiste de peças denominadas de **funções**.
- Você pode programar todas as funções que você precisa em um programa em C.
- Mas isso é reinventar a roda, muitas coisas já estão prontas e organizadas na **Biblioteca Padrão do C**.
- <http://pt.cppreference.com/w/c>



# A Biblioteca Padrão do C

---

- Geralmente quando programamos em C utilizamos a seguintes construções:
  - ▶ Funções da biblioteca padrão.
  - ▶ Funções implementadas por você mesmo.
  - ▶ Funções que outras pessoas criaram e disponibilizaram.



## A Biblioteca Padrão do C

---

- A vantagem de programar as funções é que você saberá exatamente como elas funcionam, mas consome muito tempo.
- É melhor utilizar a biblioteca do C que provê várias funcionalidades e gastar os esforços na tarefa a ser resolvida.
- Possivelmente você não irá conseguir implementar uma função de maneira mais otimizada do que uma que existe na Biblioteca Padrão.



# Sumário

---

## 5 Linguagens

- Linguagens de Máquina
- Linguagens Assembly
- Linguagens Alto Nível
- A Linguagem C
- A Biblioteca Padrão do C
- C++
- Java
- FORTRAN, COBOL, Pascal e Ada
- BASIC, VB, Visual C++, C# e .NET



# C++

---

- O C++ foi desenvolvido pelo Bjarne Stroustrup nos laboratórios da Bell.
- Foi inspirado em C, mas é um paradigma de programação diferente: programação orientada a objetos, o que é diferente da programação procedural utilizada pelo C.
- Levar vícios de programação procedural para uma programação orientada objetos não é uma boa prática de programação.



# Sumário

---

## 5 Linguagens

- Linguagens de Máquina
- Linguagens Assembly
- Linguagens Alto Nível
- A Linguagem C
- A Biblioteca Padrão do C
- C++
- **Java**
- FORTRAN, COBOL, Pascal e Ada
- BASIC, VB, Visual C++, C# e .NET



# Java

---

- Na década de 90, microprocessadores estavam impactando o design inteligente de dispositivos eletrônicos.
- A Sun em 1991 fundou um projeto interno que resultou no desenvolvimento de uma linguagem inspirada em C++.
- Esta linguagem foi batizada de Oak.
- Mas já existia uma linguagem chamada Oak. Então, quando alguns empregados da Sun visitaram um café, o nome Java foi sugerido.





# Java

---

- O mercado para dispositivos eletrônicos inteligentes no entanto não cresceu o tanto que a Sun havia previsto.
- O projeto estava na iminência de ser cancelado.
- Em 93 a WEB explodiu, e a Sun viu uma oportunidade para utilizar o Java para adicionar conteúdos dinâmicos, interatividade e animações a páginas WEB.
- O Java foi anunciado em 95 e ganhou atenção mundial juntamente com o aumento da popularidade da WEB.



# Java

---

- Hoje em dia o Java é utilizado em diversas plataformas:
  - ▶ Aplicações WEB.
  - ▶ Smartphones.
  - ▶ Tablets.
  - ▶ Aplicações Desktop.



# Sumário

---

## 5 Linguagens

- Linguagens de Máquina
- Linguagens Assembly
- Linguagens Alto Nível
- A Linguagem C
- A Biblioteca Padrão do C
- C++
- Java
- FORTRAN, COBOL, Pascal e Ada
- BASIC, VB, Visual C++, C# e .NET



# Linguagens Alto Nível

---

- Várias Linguagens alto nível foram propostas, mas poucas tornaram-se popular.
- Vamos examinar algumas delas?



# FORTRAN

---

- FORTRAN (FORmula TRANslator) foi desenvolvida pela IBM na década de 50 para uso de aplicações científicas e de engenharia.
- Após várias reformulações, continua sendo utilizada nessas aplicações (principalmente pelos Físicos).



# COBOL

---

- COBOL: COmmon Business Oriented Language.
- Desenvolvida em meados da década de 50.
- Utilizada em aplicações comerciais que requerem manipulação precisa e eficiente de grandes volumes de dados.
- Boa parte do código legado em sistemas bancários.



# Programação Estruturada

---

- Na década de 60, muito do desenvolvimento de software estava envolto em dificuldades.
- Pesquisa na década de 60 resultou no desenvolvimento da programação estruturada, que possibilitava a escrita de programas de larga escala mais fáceis de testar, depurar e modificar.



# Pascal

---

- Pascal é uma linguagem estruturada, mas não possuía muitas das características requeridas no mercado, limitando sua popularidade a ambientes acadêmicos.





# Ada

---

- Ada foi uma linguagem financiada pelo departamento de defesa dos EUA.
- Uma das importantes características era a chamada multiprogramação, que permitia que muitas atividades ocorressem em paralelo.
- Influenciou várias linguagens de programação.



# Sumário

---

## 5 Linguagens

- Linguagens de Máquina
- Linguagens Assembly
- Linguagens Alto Nível
- A Linguagem C
- A Biblioteca Padrão do C
- C++
- Java
- FORTRAN, COBOL, Pascal e Ada
- BASIC, VB, Visual C++, C# e .NET



# BASIC

---

- Desenvolvida na década de 60 com objetivo de criar programas bem simples.
- Visual Basic foi introduzido pela MS na década de 90 para simplificar o desenvolvimento de aplicações Windows.
- Por conta da popularidade do S.O, tornou-se momentaneamente uma linguagem altamente popular e requisitada.



# .NET

---

- Recentemente a MS concentrou seus esforços em integrar aplicações WEB e desktop.
- A estratégia foi criar a plataforma .NET: que provê requisitos que desenvolvedores precisam para criar aplicações distribuídas.
- Baseada em Visual Basic, Visual C++ e Visual C#.



# Sumário

---

## 6 Ambiente



# Ambiente de Desenvolvimento C

---

- Sistemas em C consistem de várias partes:
  - ▶ A linguagem.
  - ▶ O ambiente de desenvolvimento de programas.
  - ▶ A biblioteca padrão.
- Geralmente sistemas em C passam por 6 etapas.



# Ambiente de Desenvolvimento C

---

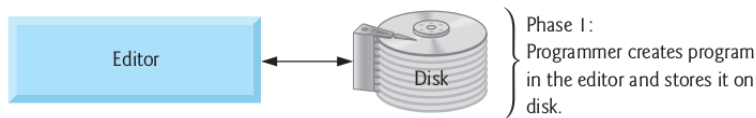
## Etapa 1: Criação do Programa

- Através de um editor de texto ou de uma IDE, o programador escreve o programa em C.
- Armazenados em arquivos no disco rígido.
- Exemplos de editores ou IDEs: Code::Blocks, Sublime, Atom, Clion, Vim, Emacs, nano, gedit, geany, Eclipse, MS Visual Studio, ...



# Ambiente de Desenvolvimento C

---







# Ambiente de Desenvolvimento C

---

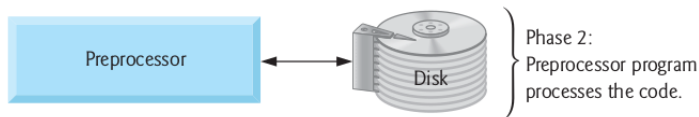
## Etapa 2: Pré-Processamento

- Ao invocar o compilador, a primeira coisa que é feita é a invocação do pré-processador C.
- O pré-processador é responsável pela manipulação do programa antes da compilação propriamente dita.
- As diretivas de pré-processamento especificam que manipulações devem ser realizadas.
- As manipulações geralmente consistem de: inclusão de outros arquivos e substituição de textos.



# Ambiente de Desenvolvimento C

---





# Ambiente de Desenvolvimento C

---

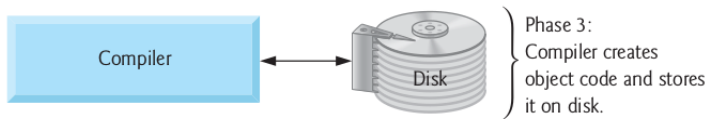
## Etapa 3: Compilação

- A compilação transforma o programa manipulado pelo pré-processador em linguagem de máquina.



# Ambiente de Desenvolvimento C

---





# Ambiente de Desenvolvimento C

---

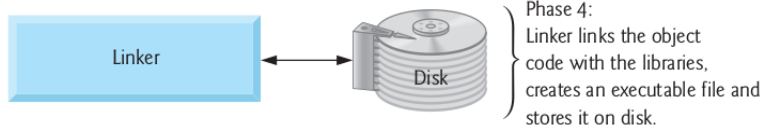
## Etapa 4: Ligação

- Os programas em C contêm referências para as funções definidas em outros arquivos, como nas bibliotecas padrões ou bibliotecas do programador.
- O código objeto produzido pelo compilador possui então “buracos”.
- O ligador liga o código objeto do código compilado com o código objeto das funções definidas em outros lugares.
- Após a ligação temos o **executável**.



# Ambiente de Desenvolvimento C

---





# Ambiente de Desenvolvimento C

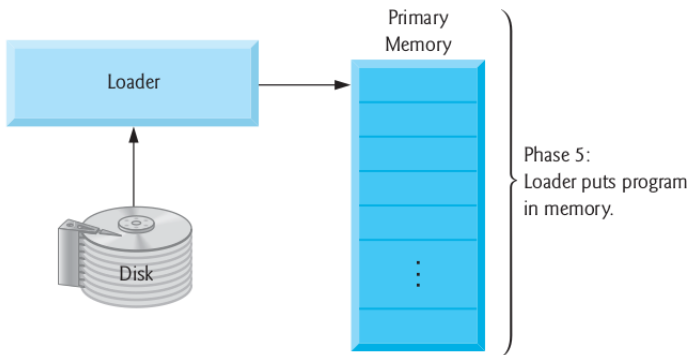
---

## Etapa 5: Carregamento

- O carregamento insere o programa em memória.
- Realizado pelo loader: transfere a imagem do executável do disco para a memória.
- Componentes adicionais de bibliotecas dinâmicas (shared) são carregadas nesta etapa.



## Ambiente de Desenvolvimento C







# Ambiente de Desenvolvimento C

---

## Etapa 6: Execução

- Após o carregamento, o programa pode ser executado.



## Ambiente de Desenvolvimento C

