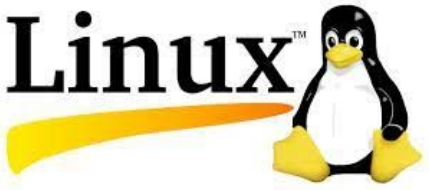
Introdução à linguagem C
Aula 01 - Introdução

Professor: Racyus Delano

E-mail: racyus@univicosa.com.br

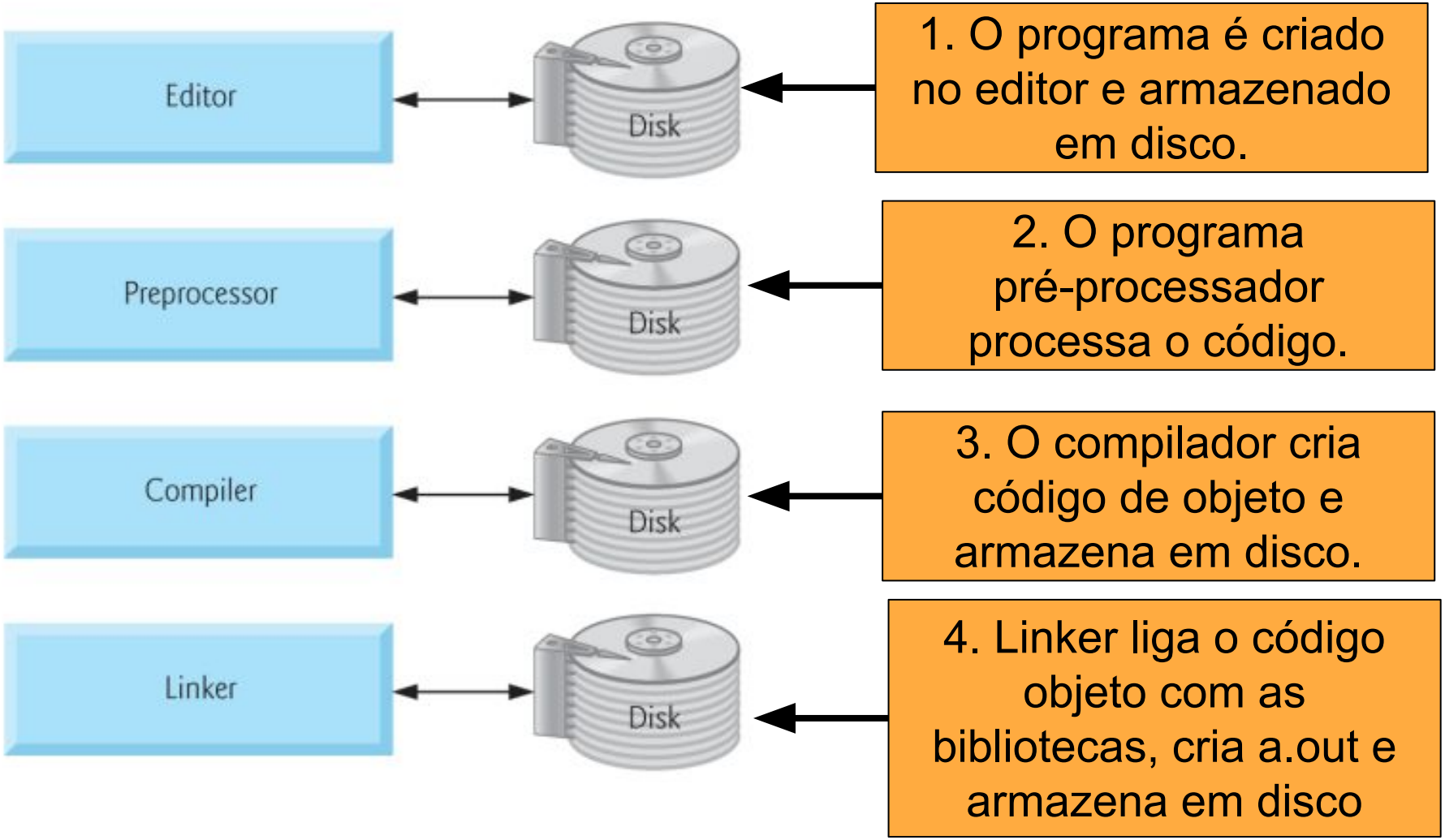
Introdução

- **Características**
  - Criada pelo laboratório Bell AT&T em 1970;
  - É uma linguagem independente do hardware;
    - Roda em diferentes plataformas de computadores;
    - Altamente portátil;
  - Fornece acesso à baixo nível.
    - Ex: Acesso à memória, leitura de registradores, etc….
  - Está presente no núcleo do Linux;



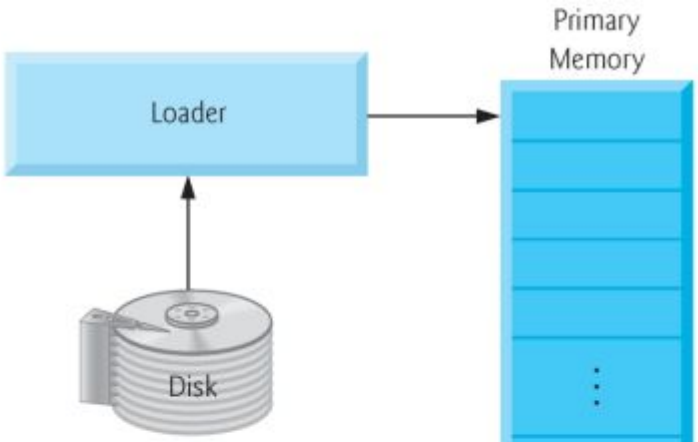
Ambiente de desenvolvimento C

- **Fases dos programas C**

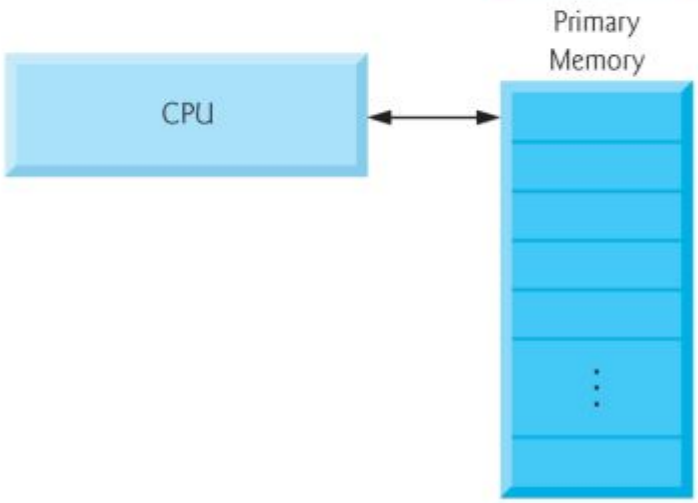


Ambiente de desenvolvimento C

- **6 fases dos programas C**

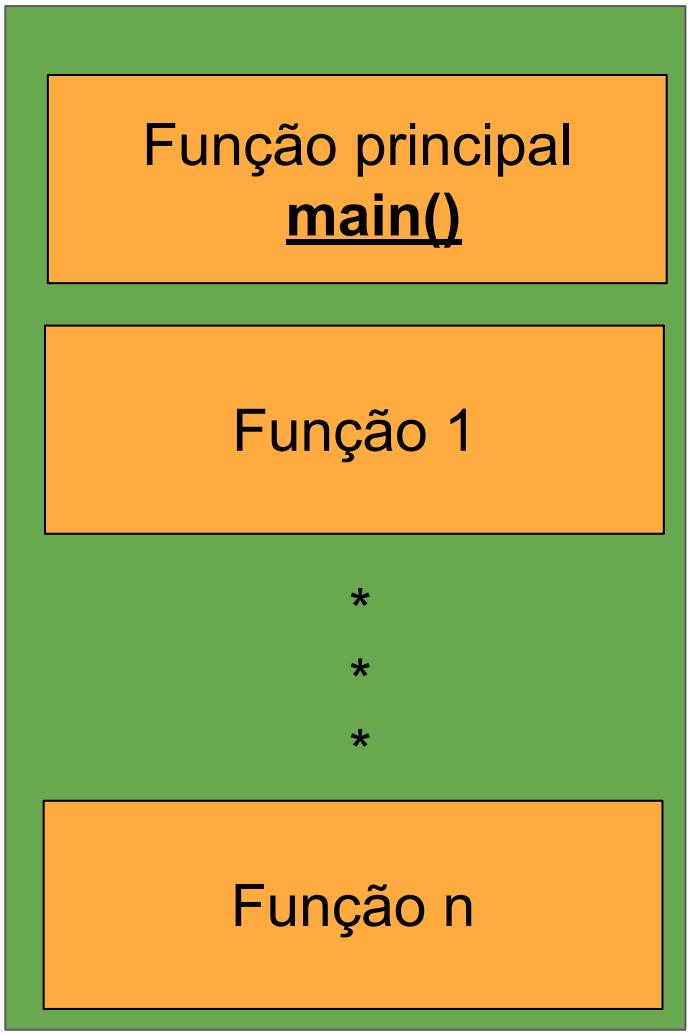


5. O loader coloca o programa na memória.



6. A CPU pega cada instrução e a executa, possivelmente armazenando novos valores de dados conforme o programa é executado.

Estrutura de um programa C

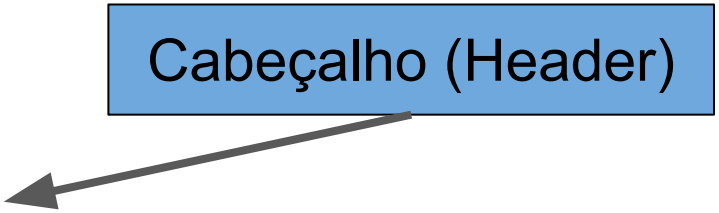


Programa

Todo programa C precisa ter uma função principal definida como **main()**.

Estrutura de um programa C

- **Função**
  - Cada função consiste de um cabeçalho (**header**) seguido de um bloco (**block**);



  - **Formato geral:**

    <Tipo do retorno> Nome da função (lista de argumentos)

    Bloco de código

    - **Tipo do retorno:** char, int, float, double, void.
    - **Nome da função:** main, hello_world, etc.
    - **Lista de argumentos:** nome das variáveis que serão passadas para função.
      - Exemplo: número, idade, quantidade ….

Estrutura de um programa C

- **Bloco de código**
  - Um bloco pode ter nenhuma ou várias instruções;
  - Ponto e vírgula (**;**) é utilizado para terminar as instruções;
  - Boa prática de programação: manter os blocos alinhados.
    - Cada bloco interior ao principal pode incluir declaração de variáveis.

```
{
    Declaração de variáveis
    Instruções executáveis
}
```

Estrutura de um programa C

- **Comentário**
  - Torna programas fáceis para ler e modificar (Documentação);
  - É ignorado pelo compilador C;
  - Duas formas:
    - Comentar linha usando **//**
      - A linha inteira será ignorada;
      - Exemplo: **// a = 2;**
    - Comentar bloco usando **/* */**
      - Todo o código entre **/* */** será ignorado;
      - Exemplo:

```
/*
  Nome do programa: hello_world.c
  Descrição: Meu primeiro programa C.
  Autor(es): Racyus Delano
  Data: 07/02/2023
*/
```

Estrutura de um programa C

- **Instrução de retorno**
  - Determina o fim da execução da função;
  - Retorna para a função que invoca a função processada.

```
int main()          //Cabeçalho
{                   //Início do bloco
                    // ……
   return 0;        //Programa terminado com sucesso
}                   //Fim do bloco
```

Estrutura de um programa C

- **Diretiva do pré-processador: #include**
  - Serve para importar bibliotecas e arquivos;
  - Uma linha com **#** é processada pelo compilador antes de começar a tradução do programa;
  - **Importando uma biblioteca**
    - **#include<stdio.h>**
  - **Importando um arquivo**
    - **#include<minha_funcao.h>**

Estrutura de um programa C

- **Entrada/Saída**
  - **Entrada de dados:** O programa lê um valor de uma variável digitado no teclado pelo usuário;
  - **Saída de dados:** O programa escreve um valor em uma variável e mostra na tela;
  - Na linguagem C, a biblioteca **<stdio.h>** contém as funções para entrada e saída de dados do programa.
    - Declarar #include<stdio.h> no início do programa;
    - **Funções de entrada:** scanf, fscanf, …
    - **Funções de saída:** printf, fprintf, …

Estrutura de um programa C

- **Formato da saída**
  - São usados alguns caracteres especiais para formatar a saída.
  - Eles são inseridos dentro do *printf*.

Caracteres	**Descrição**
\n	Nova linha
\t	Tab na horizontal
\r	Recuo
\a	Alerta
\\	Imprime uma \
\”	Usado para imprimir “

Meu primeiro programa C

- **Hello World! :)**

```c
/*
  Nome do programa: hello_world.c
  Descrição: Meu primeiro programa C.
  Autor(es): Racyus Delano
  Data: 20/02/2024
*/

#include<stdio.h>

int main()
{
    printf("Hello World!");
    return 0;
}
```

Compilar e executar um programa C

- **Windows**
  - **IDEs (Integrated Development Environment)**
    - Editor;
    - Compilador;
    - Debugger.

  - **Exemplos**
    - Dev C++
    - Code::Blocks
    - Visual code
    - Netbeans ou Eclipse
    - https://www.onlinegdb.com
    - ….

Compilar e executar um programa C

- **Linux**
  - **Compilar**
    - gcc <nome_do_programa.c> -o <nome_do_executavél>
      - **gcc hello_world.c -o hello_world.o**
  - **Executar**
    - ./<nome_do_executavél>
      - **./hello_world.o**
  - **Compilar e retornar mensagens de avisos (-WALL)**
    - **gcc -WALL hello_world.c -o hello_world.o**
  - **Compilar usando a biblioteca math.h (-lm)**
    - gcc programa.c -o programa.o -lm

Compilar e executar um programa C

- **Linux**
  - **Comandos auxiliares**
    - Criar um diretório: **mkdir nome_diretorio**
    - Entrar em um diretório: **cd nome_diretorio**
    - Voltar um diretório: **cd ../**
    - Criar um arquivo: **touch nome_arquivo**
    - Renomear um arquivo: **mv arquivo_original arquivo_novo**
    - Verificar o diretório atual: **pwd**
    - Listar diretório(s): **ls**
    - Exemplo
      - **cd /home/seu_usuario/**
      - **mkdir aula01**
      - **touch programa01.c**
      - **pwd** (Retorna /home/seu_usuario/aula01)
      - **ls** (Retorna programa01.c)

Compilar e executar um programa C

- **Windows**
  - **Instalação e configuração ambiente desenvolvimento**
    - **Compilador GCC - MinGW - Tutorial;**
    - **Code blocks - Tutorial;**
    - **Visual Code Studio - Tutorial.**

Exercícios sala de aula

1. C é uma linguagem independente do hardware criada pelo laboratório Bell AT&T em 1970 que está presente no núcleo do Linux. Sobre as fases dos programas C, relacione adequadamente as colunas a seguir.

1.Editor
2.Pré-processador
3.Compilador
4.Linker
5.Loader
6.CPU

( ) Cria código de objeto e armazena em disco.

( ) É criado no editor e armazenado em disco.

( ) Processa o código.

( ) Pega cada instrução e a executa, possivelmente armazenando novos valores de dados conforme o programa é executado.

( ) Coloca o programa na memória.

( ) Liga o código objeto com as bibliotecas, cria a.out e armazena em disco.

A sequência está correta em:

A) 2, 4, 6, 3, 1, 5
B) 1, 3, 5, 4, 2, 6
C) 6, 2, 4, 5, 3, 1
D) 3, 1, 2, 6, 5, 4
E) 4, 5, 1, 2, 6, 3

Exercícios sala de aula

1. C é uma linguagem independente do hardware criada pelo laboratório Bell AT&T em 1970 que está presente no núcleo do Linux. Sobre as fases dos programas C, relacione adequadamente as colunas a seguir.

1.Editor
2.Pré-processador
3.Compilador
4.Linker
5.Loader
6.CPU

( ) Cria código de objeto e armazena em disco.

( ) É criado no editor e armazenado em disco.

( ) Processa o código.

( ) Pega cada instrução e a executa, possivelmente armazenando novos valores de dados conforme o programa é executado.

( ) Coloca o programa na memória.

( ) Liga o código objeto com as bibliotecas, cria a.out e armazena em disco.

A sequência está correta em:

A) 2, 4, 6, 3, 1, 5
B) 1, 3, 5, 4, 2, 6
C) 6, 2, 4, 5, 3, 1
D) 3, 1, 2, 6, 5, 4
E) 4, 5, 1, 2, 6, 3

Exercícios sala de aula

2. Executar programa hello world em C.

```c
/*
  Nome do programa: hello_world.c
  Descrição: Meu primeiro programa C.
  Autor(es): Racyus Delano
  Data: 20/02/2024
*/

#include<stdio.h>

int main()
{
    printf("Hello World!");
    return 0;
}
```

Exercícios sala de aula

3. Escreva um programa que leia o número de alunos e de alunas de uma sala. Como saída, o programa deve apresentar o número de alunos e em seguida o de alunas.

Exercícios sala de aula

3. Escreva um programa que leia o número de alunos e de alunas de uma sala. Como saída, o programa deve apresentar o número de alunos e em seguida o de alunas.

```c
/*************************************************
  Nome do programa: exercicio3.c
  Descrição: Imprimir o número de alunos e alunas.
  Autor(es): Racyus Delano
  Data: 20/02/2024
*************************************************/
#include <stdio.h>

int main()
{
    int num_alunos, int_num_alunas;

    printf(“Digite o número de alunos: ”);
    scanf(“%d”, &num_alunos);

    printf(“Digite o número de alunas: ”);
    scanf(“%d”, &num_alunos);

    printf("O número de alunos é: %d\n", num_alunos);
    printf("O número de alunas é: %d\n", num_alunas);

    return 0;
}
```

Exercícios da lista 01

- **Data de entrega: 22/02/2024.**

1. Escreva um programa para receber 3 valores inteiros do usuário e mostrar a sua média (que pode não ser inteira).

2. Escreva um programa para:
   a. Receba do usuário um tempo em segundos, correspondente à duração de um evento qualquer (por exemplo, jogo de futebol);

   b. Calcule e mostre ao usuário o tempo equivalente em horas, minutos e segundos.