

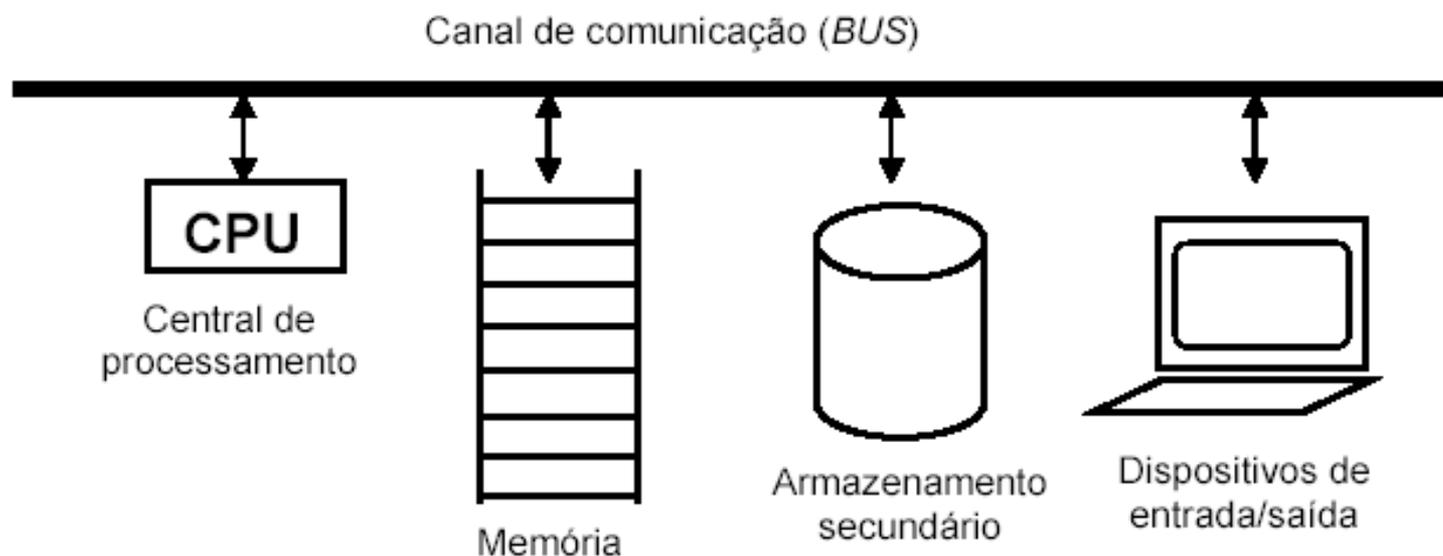


# INF 1005 – Programação I

## Aula 02 – Algoritmos e Ciclo de Desenvolvimento

Edirlei Soares de Lima  
<elima@inf.puc-rio.br>

# Modelo de um Computador



# Linguagem de Máquina

- Um processador executa **instruções de máquina**
- Instruções de máquina são muito simples, mas são executadas em altíssima velocidade
- Um conjunto típico de instruções inclui operações aritméticas, lógicas e de desvio
- Exemplo de seqüência típica de instruções:
  1. Carregue o conteúdo da posição de memória 40 no registrador 1 (R1);
  2. Carregue o valor 100 no registrador 2 (R2);
  3. Se o conteúdo de R1 for maior do que o conteúdo de R2 prossiga com a instrução armazenada na posição de memória 240;

# Linguagem de Máquina

- Instruções de máquina são representadas por seqüências de **dígitos binários**

- Exemplo:

Adicionar R1 e R2 e armazenar o resultado em R6

```
[ op | rs | rt | rd |shamt| funct]
  0  1  2  6  0   32
000000 00001 00010 00110 00000 100000
decimal
binary
```

Desviar para a instrução armazenada no endereço de memória 1024

```
[ op | target address ]
  2           1024
000010 00000 00000 00000 10000 000000
decimal
binary
```

# Linguagens de Alto-Nível

- **Programar em linguagem máquina** é uma tarefa entediante e propensa a erros
- A partir de meados dos anos 50 várias **linguagens de alto nível** foram criadas
- Possuem **nível de abstração** relativamente elevados
- Elas são mais próximas das linguagens utilizadas pelos seres humanos

# Linguagens de Alto-Nível

- **Exemplos de linguagens de alto-nível:**
  - FORTRAN (1957)
  - COBOL (1960)
  - PASCAL (1970)
  - C (1972)
  - C++ (1983)
  - JAVA (1995)

# Exemplo de Programa em C

```
int main(void)
{
    int a = 3, b = 7, c;

    c = a + b;
    if (c%2 == 0)
        printf("o resultado e par\n");
    else
        printf("o resultado e impar\n");

    return 0;
}
```

# Compilação

- Programas escritos em linguagens de alto nível **não são executados** diretamente pelo processador
- Antes de executá-lo é preciso **traduzir os comandos** de alto-nível para instruções de máquina
- Esta tarefa (**compilação**) é realizada por um programa chamado de **compilador**.
- Linguagem Compilada X Linguagem Interpretada

# Ciclo de Desenvolvimento



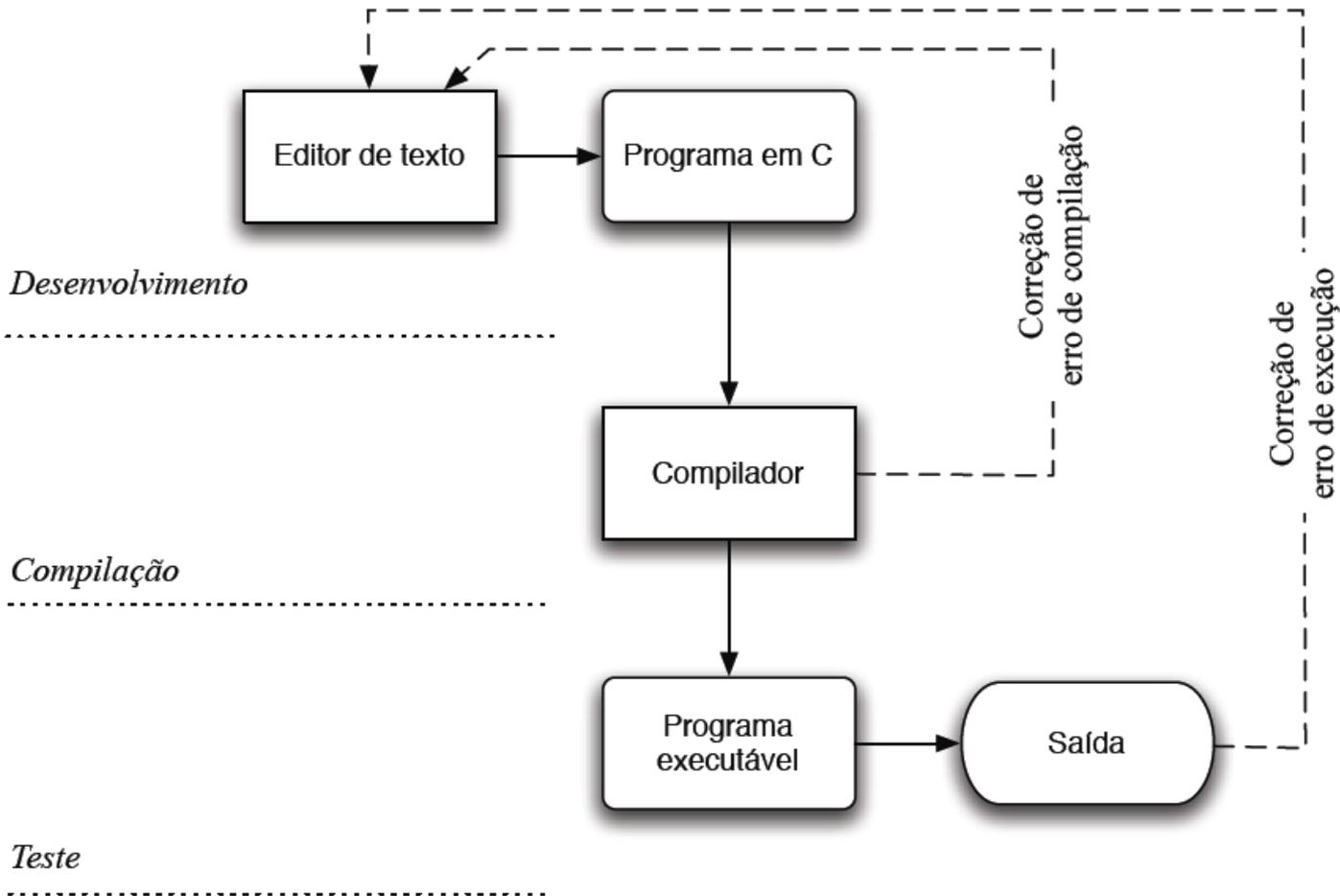
# Ciclo de Desenvolvimento

- Um programa escrito em C deve respeitar a **sintaxe da linguagem**
- Um computador não é capaz de executar uma sequência de instruções escritas em C
  - ou em qualquer outra linguagem de alto nível
- Para executar um programa em C, antes é necessário **converter o programa** em C para um programa em linguagem de máquina
  - processo chamado de **COMPILAÇÃO**
  - processo realizado por um **COMPILADOR**

# Ciclo de Desenvolvimento

- **Escolha da linguagem** de programação depende da aplicação que será desenvolvida
- Na disciplina INF1005, a escolha da linguagem C está relacionada a:
  - eficiência
  - facilidade de uso
  - base para muitas outras linguagens
  - mais usada atualmente
  - base para Programação II

# Ciclo de Desenvolvimento



# Definição de Problemas

- **Problema:** Andando pelo campus da PUC, alguém lhe pergunta:

*“Como faço para ir ao centro da cidade?”*

- O que você responde?

# Definição de Problemas

- **Faltam detalhes:**
  - meio de transporte
    - envolve tempo e dinheiro
  - endereço específico
  - quando?
    - envolve trajeto devido a engarrafamento
  - número de pessoas
- Fica difícil responder sem saber as reais necessidades de quem perguntou

# Solução de Problemas

- **Objetivo:** encontrar uma sequência de passos que permitam que o problema possa ser resolvido de maneira automática e repetitiva
- **Descrição da solução** (como fazer)
  - clara
  - não ambígua
  - sequencial
- **Atores envolvidos**
  - criador da solução
  - executor da solução
  - usuário da solução
- Linguagem utilizada na solução deve ser **compreendida** pelo executor da solução

# Solução de Problemas

- Definir o objetivo do problema
- Entender a proposta do problema
- Levantar dados sobre a situação que ele descreve
- Gerar alternativas para a solução (idéias)
- Avaliar as alternativas e chegar à solução
- Refletir acerca dessa solução
- Fazer com que ela seja entendida e aceita

# Algoritmos

- **Lógica de Programação** é a técnica de criar sequências lógicas de ações para atingir determinados objetivos.
- **Sequências Lógicas** são instruções executadas para atingir um objetivo ou solução de um problema.
- **Instruções** são uma forma de indicar ao computador uma ação elementar a executar.
- Um **Algoritmo** é formalmente uma sequência finita de instruções que levam a execução de uma tarefa.

# Algoritmos

- Até mesmo tarefas simples, podem ser descritas por **sequências lógicas**:

## *“Chupar uma bala”*

- 1) Pegar a bala;
- 2) Retirar o papel;
- 3) Chupar a bala;
- 4) Jogar o papel no lixo;



## *“Somar dois números”*

- 1) Escreva o primeiro número no retângulo A.
- 2) Escreva o segundo número no retângulo B.
- 3) Some o número do retângulo A com número do retângulo B e coloque o resultado no retângulo C.



# Escrevendo Algoritmos

- Os algoritmos podem ser escritos diretamente em uma linguagem de programação ou simplesmente descritos em pseudocódigo.
- **Pseudocódigo** é uma forma genérica de escrever um algoritmo.
- **Linguagens de programação** são formas padronizadas de comunicar instruções para o computador. São conjuntos de regras sintáticas e semânticas usadas para definir um programa de computador.

# Escrevendo Algoritmos

## Processo Geral de um Algoritmo



- **Entrada:** O algoritmo recebe os dados de entrada.
- **Processamento:** Os procedimentos para se chegar ao resultado final são executados.
- **Saída:** O resultado final é mostrado.

# Escrevendo Algoritmos

- **Exemplo:** “Ler duas notas e calcular a média”

## Algoritmo descritivo:

Leia a primeira nota e armazene ela em **nota1**;  
Leia a segunda nota e armazene ela em **nota2**;  
Some a **nota1** com a **nota2** e divida o **resultado** por 2;  
Mostre o **resultado**;

## Pseudocódigo:

### **variáveis**

```
nota1, nota2, media : real;
```

### **início**

```
escreva("Digite a nota 1");  
leia(nota1);  
escreva("Digite a nota 2");  
leia(nota2);  
media = (nota1 + nota2) / 2;  
escreva(media);
```

### **fim**

# Exercício

- 1) Crie um programa que faça a multiplicação dois números digitados pelo usuário
  - a) Escreva o algoritmo de forma descritiva
  - b) Escreva o algoritmo em pseudocódigo

# Exercício

2) Crie um programa que calcule a área de um triângulo

$$Area = \frac{b * h}{2}$$

- a) Escreva o algoritmo de forma descritiva
- b) Escreva o algoritmo em pseudocódigo