

ALGORITMOS E PROGRAMAÇÃO DE COMPUTADORES

AULA 7

Fundamentos de Programação

Vandor Roberto Vilardi Rissoli

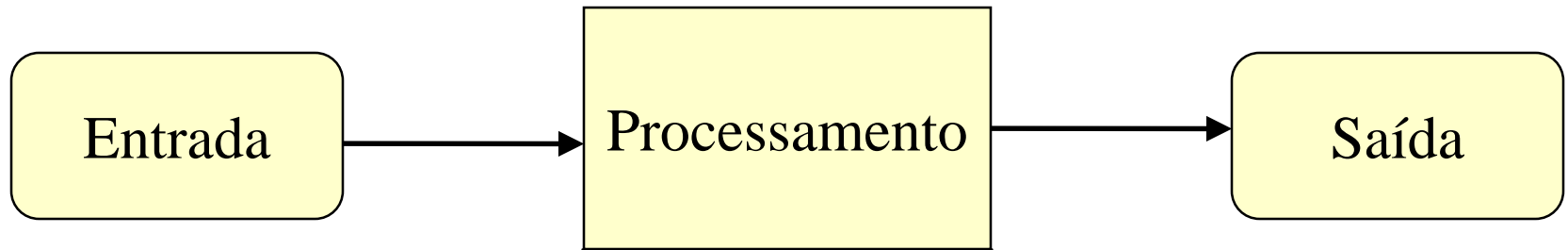


APRESENTAÇÃO

- Processamento de Dados
- Linguagem de Programação
 - Linguagem de Máquina (ou binária)
 - Linguagem de Montagem
 - Linguagem de Alto Nível
- Etapas de Desenvolvimento
- Referências



Processamento de Dados

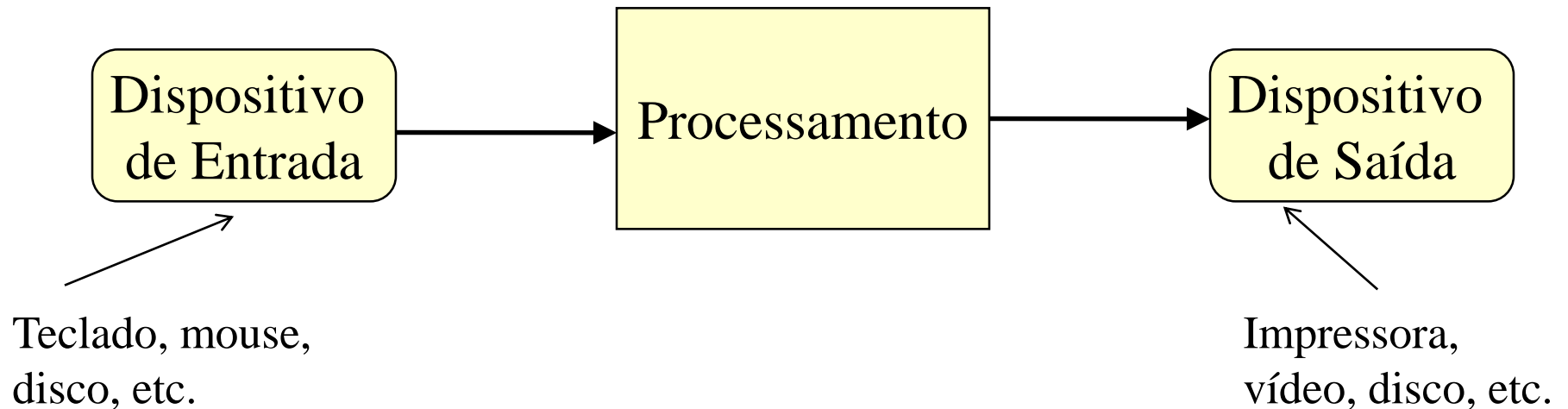


qualquer atividade ou conjunto de atividades ordenadamente realizadas que, utilizando informações (ou dados) básicas, efetua transformações para obter novas informações (ou dados) como resultado

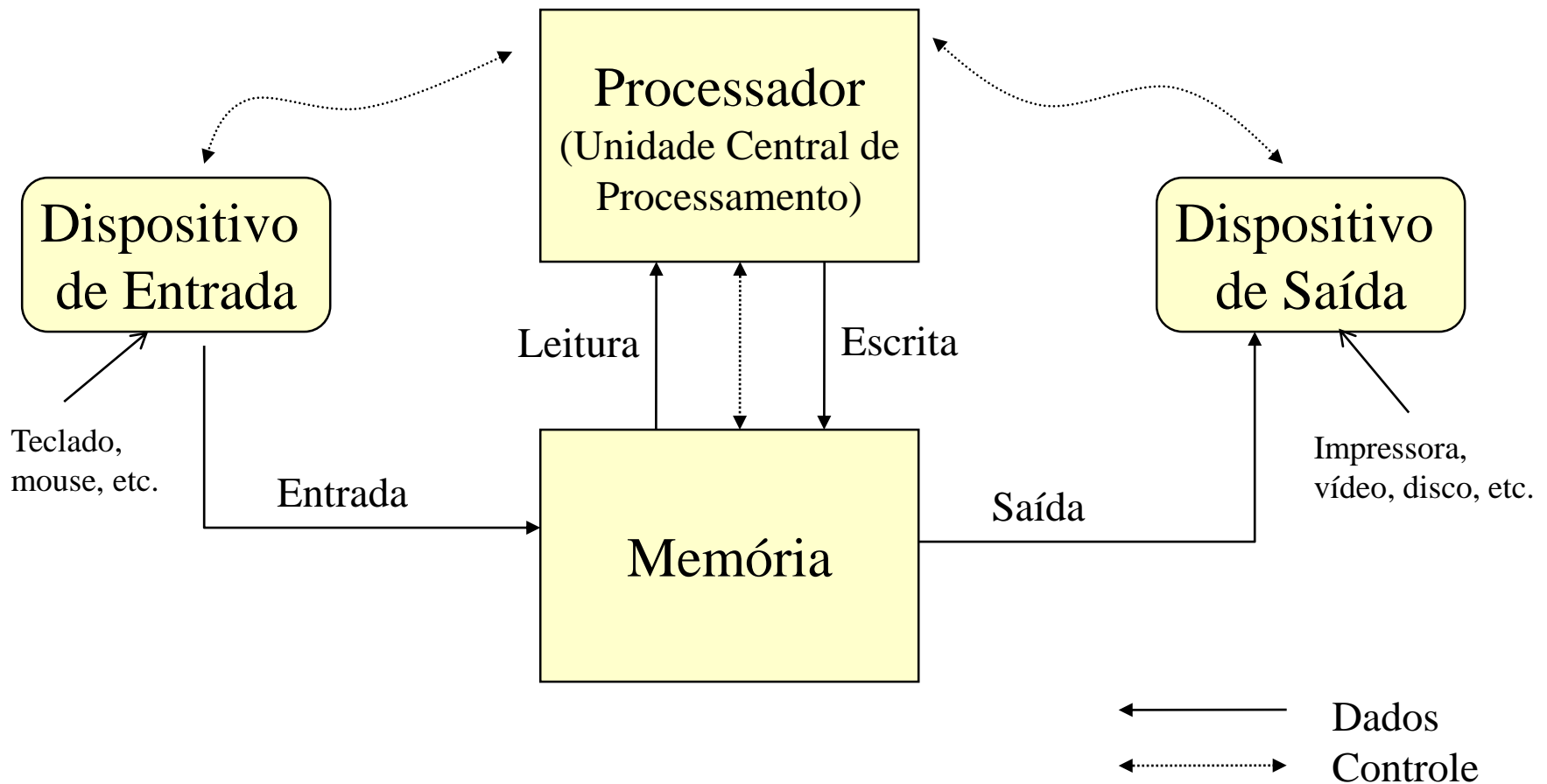


Processamento de Dados

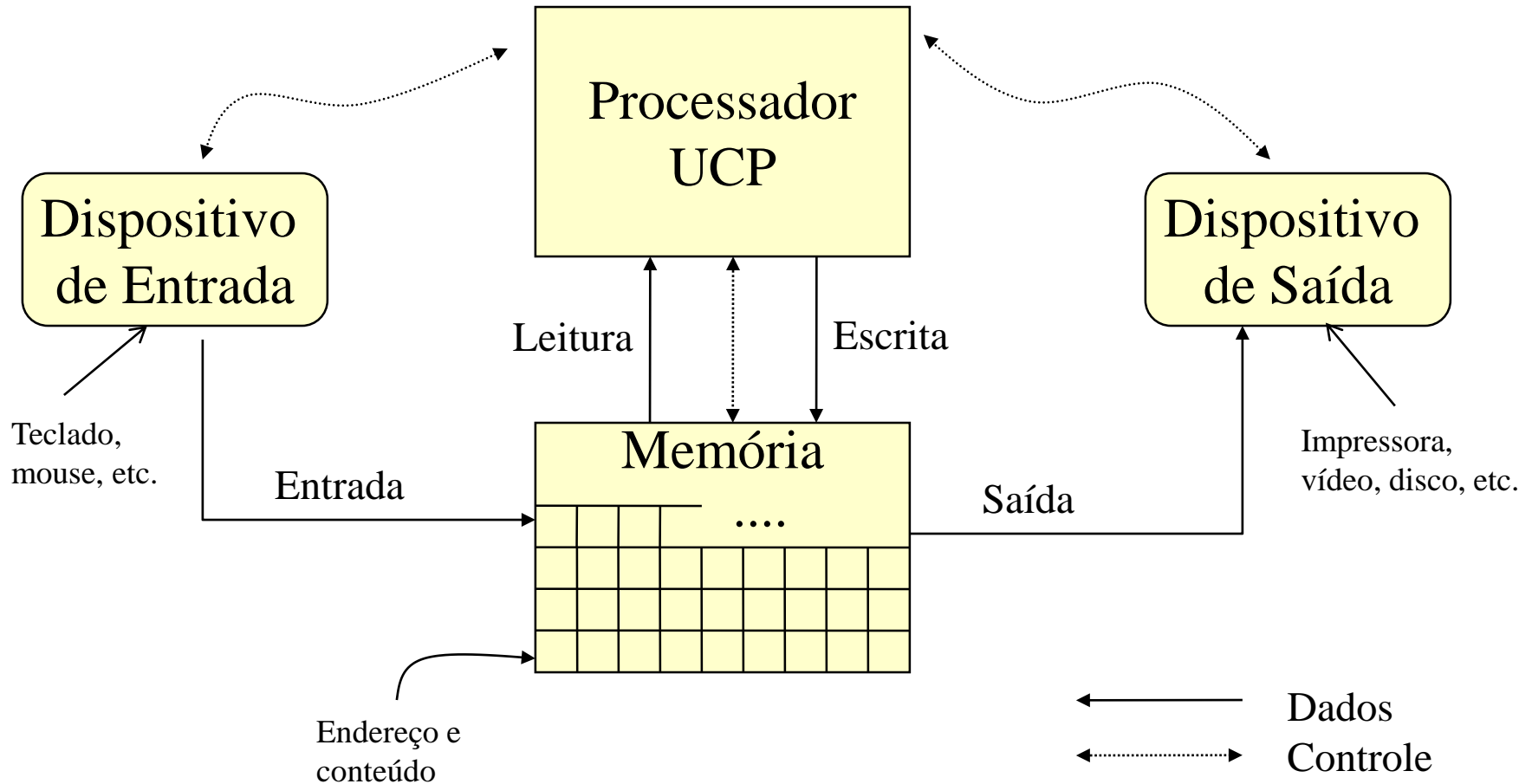
Componentes Principais ao Processamento de Dados



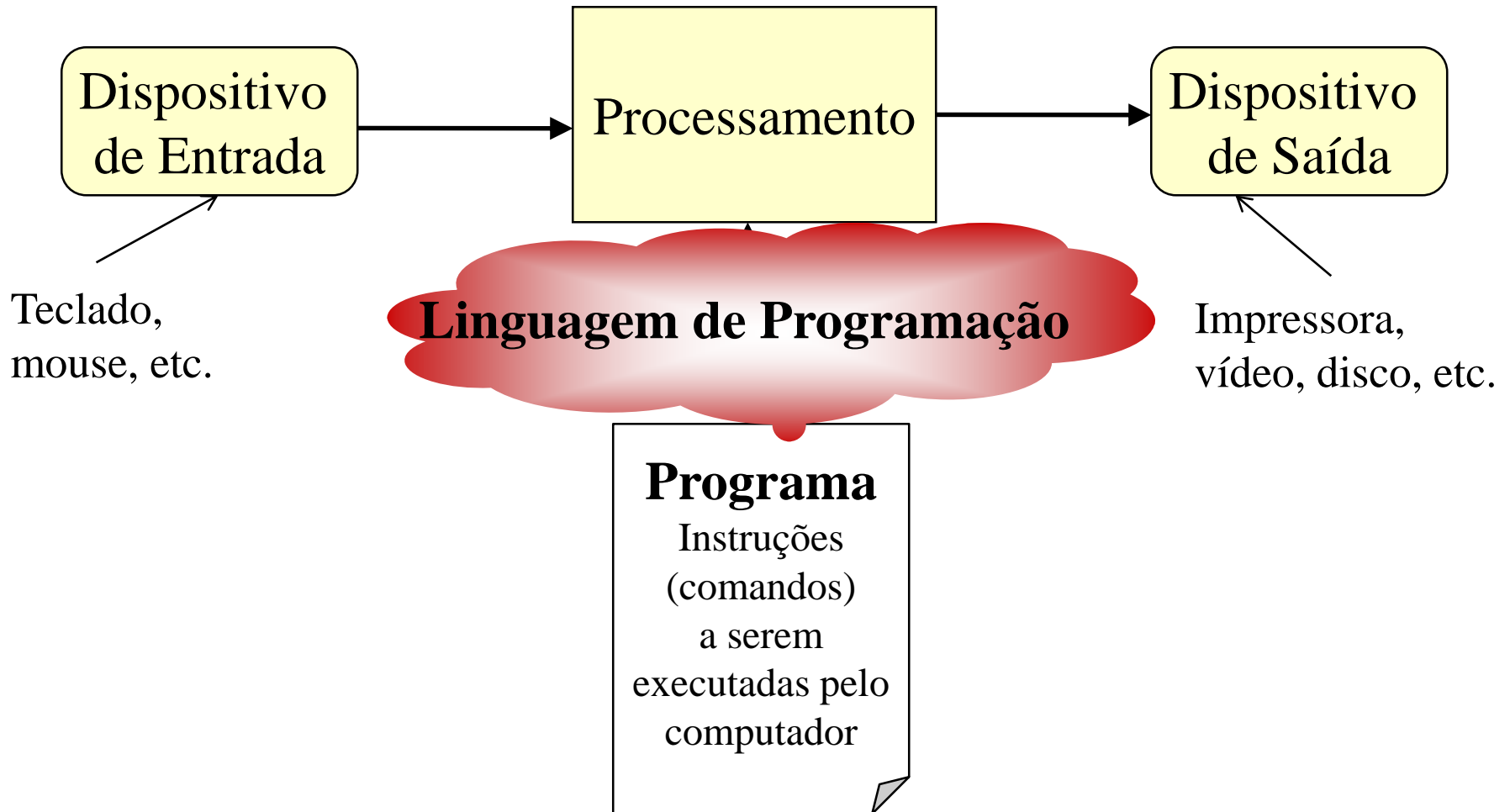
Processamento de Dados



Processamento de Dados

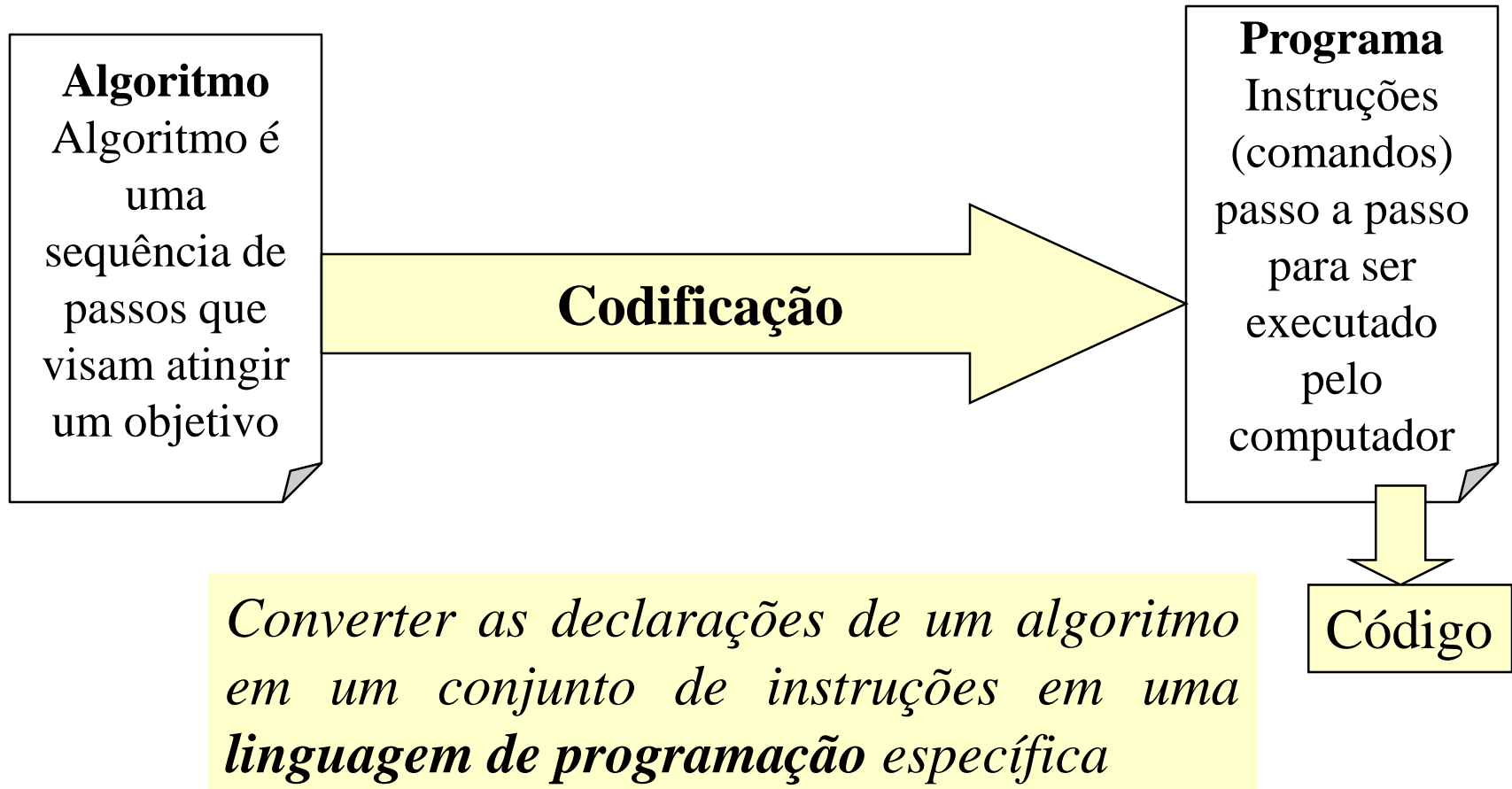


Processamento de Dados



Processamento de Dados

Algoritmo X Programa

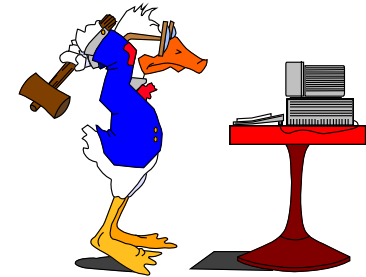


Linguagem de Programação

Linguagem de Alto Nível



Linguagem de Montagem



Linguagem de Máquina



Linguagem de Programação

Linguagem de Máquina

- código na forma binária (ou hexadecimal)
- necessidade de conhecer as instruções da máquina e seus respectivos códigos de operação
- necessidade de conhecer os componentes (registradores) do processador (UCP) e os endereços de memória
- Exemplo: *Programa para somar 23 + 100*

```
0010 0001 0000 0000 0001 0111  
0000 0001 0000 0000 0110 0100  
0000 1001
```

Trecho do programa em **linguagem binária**

Linguagem de Programação

Linguagem de Montagem

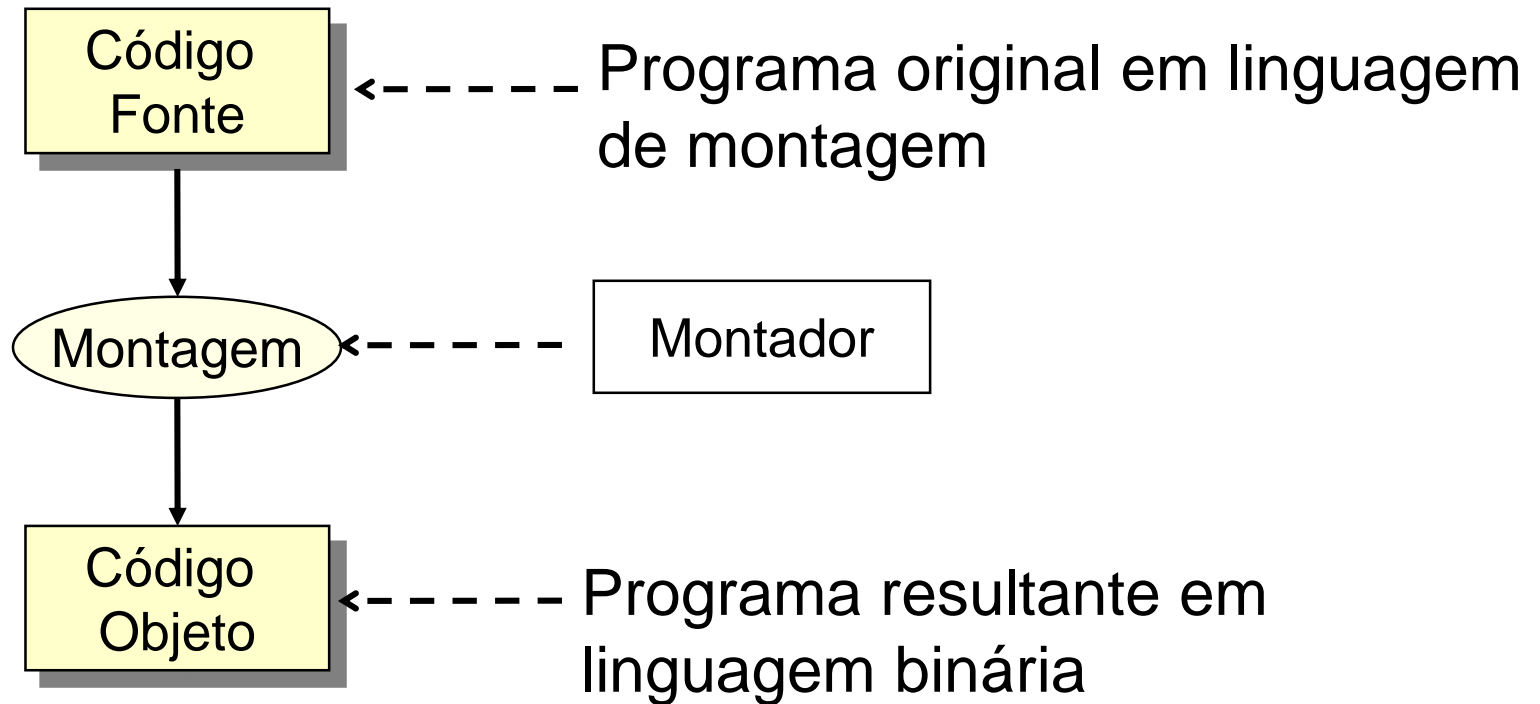
- “Assembler language”
- instruções escritas por meio de símbolos
- cada instrução corresponde a um único comando em linguagem de máquina
- necessidade de converter para linguagem de máquina
 - Processo chamado de **Montagem** realizado por um programa chamado **Montador** (ou Assembler)
- Exemplo: *Programa para somar 23 + 100*

```
LD HL, 23  
LD BC, 100  
ADD HL, BC
```

Trecho do programa em linguagem Assembler

Linguagem de Programação

Processo de Montagem



Linguagem de Programação

Funções do Montador

- Verificar a correção do programa
- Substituir códigos de operações simbólicas por valores numéricos
- Substituir nomes simbólicos de endereços por valores numéricos
- Reservar espaço de memória para o armazenamento de funções e dados
- Converter valores de números para valores binários



Linguagem de Programação

Linguagem de Alto Nível

- estruturadas de acordo com a compreensão do programador
- cada instrução pode corresponder a dezenas de instruções de máquina
- é orientada ao problema
- necessidade de converter o programa para linguagem de máquina:
 - Processo chamado de **Compilação** realizado por um programa chamado **Compilador**
- Exemplos de Linguagens de Alto Nível:
Pascal, COBOL, C, Java e várias outras.



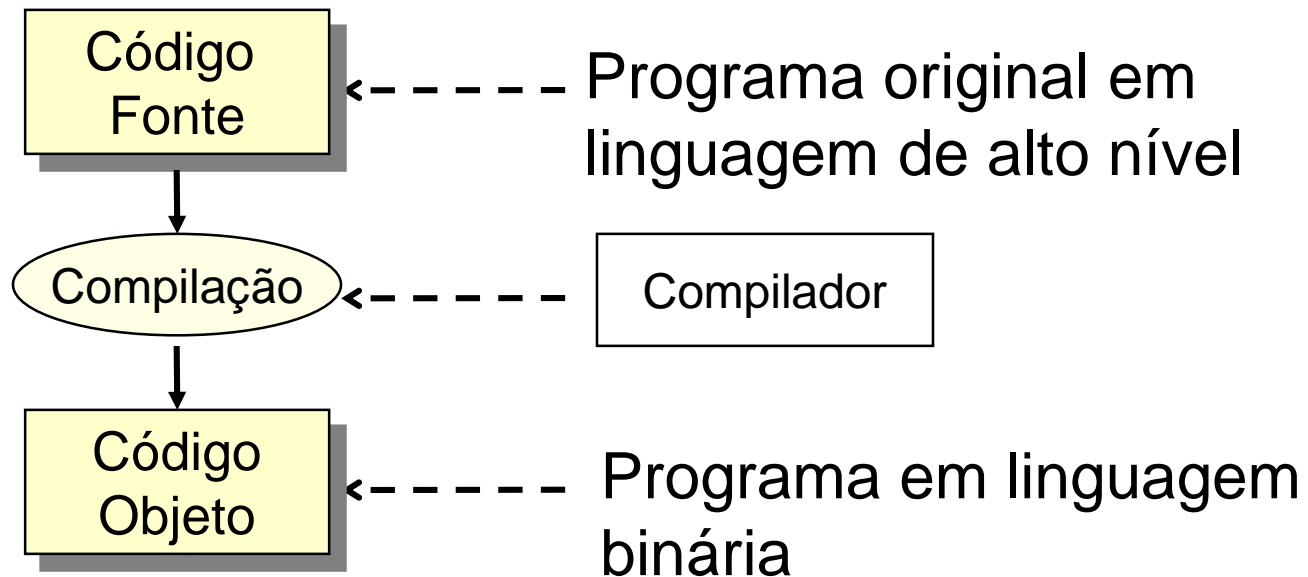
Linguagem de Programação

- Exemplo: *Programa para somar 23 + 100*

```
A = 23  
B = 100  
TOTAL = A + B
```

Trecho do Programa em Fortran

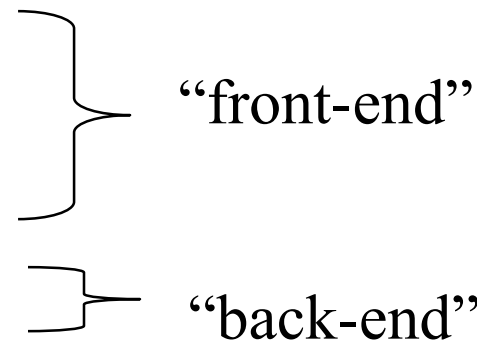
Processo de Compilação



Linguagem de Programação

Funções do Compilador

- Compilação é semelhante a montagem, mas é mais complexo e demorado (**diferente da relação 1:1 das linguagens de montagem**)
- A compilação também é chamada de **tradução**
- Realiza várias tarefas:
 - Análise léxica
 - Análise sintática
 - Análise semântica
 - Criação do código binário



Linguagem de Programação

Funções do Compilador

- **Análise léxica**
 - verificar se todos os símbolos (comandos ou não) utilizados são permitidos pela linguagem (ex: igualdade em Pascal é “:=“ em C é “=“)
- **Análise sintática**
 - verifica a estrutura de cada comando
- **Análise semântica**
 - análise semântica estática (regras de compilação) dos comandos
 - Exemplo: o desvio para uma linha que não existe na linguagem Basic
- O módulo “back-end” cria o código binário e aloca espaço de memória

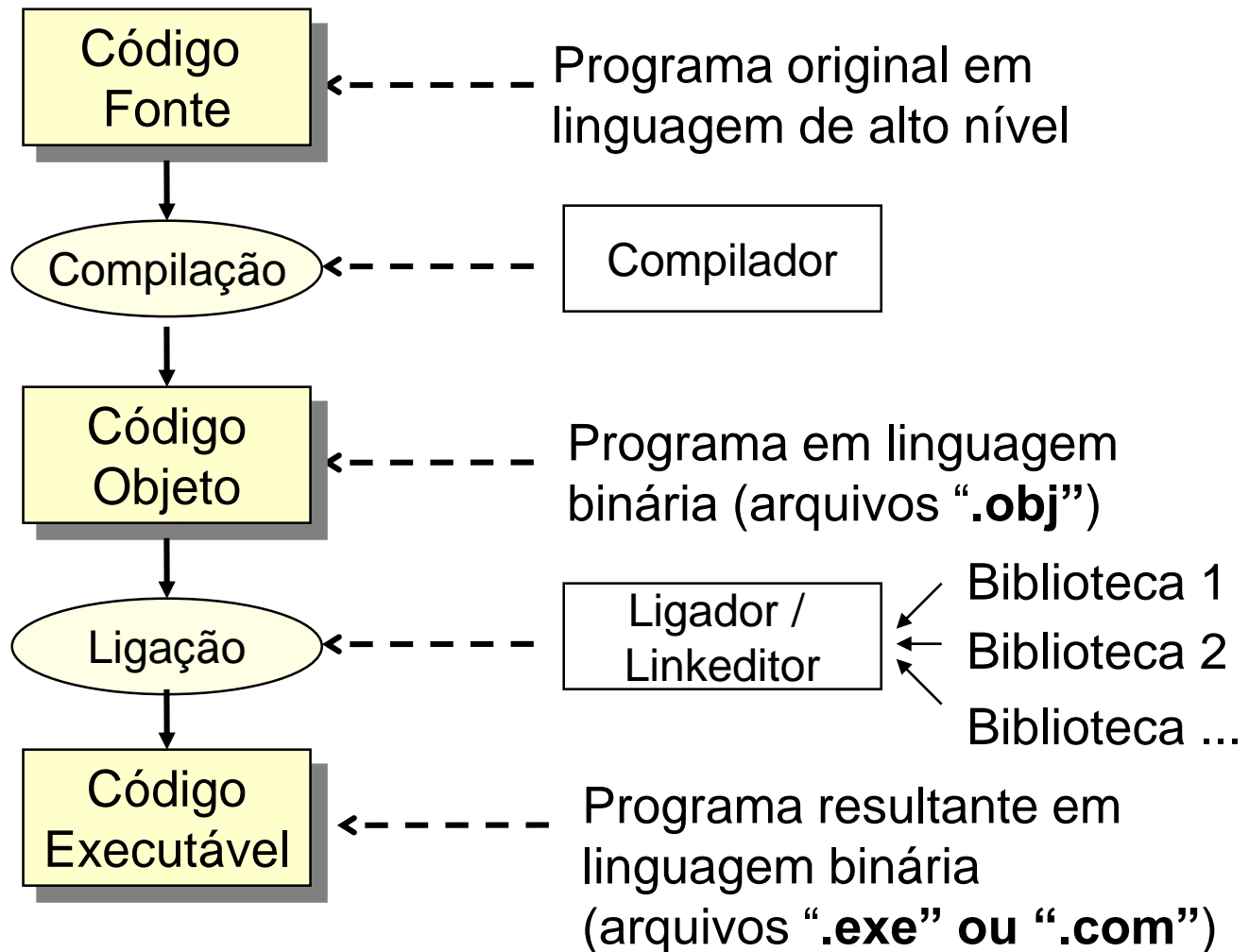
Linguagem de Programação

Processo de Ligação ou Linkedição

- Facilidade criada para evitar que o programador precise codificar todas as operações
- Algumas operações já são disponibilizadas (no sistema) em código binário pelas linguagens
- O código é buscado e integrado ao programa que está sendo feito
 - processo chamado **linkedição** (ou ligação) realizado por um **linkeditor** (ou ligador)
- Exemplo de códigos existentes:
 - impressão, raiz quadrada, comparação de dois nomes, etc.
- Esses códigos objetos (rotinas) são organizados em Bibliotecas
 - identificados pelo nome da rotina
 - incorporados ao programa por uma chamada à biblioteca

Linguagem de Programação

Processo de Linkedição



Linguagem de Programação

Funções do Linkeditor

- Compilador quando encontra uma chamada a rotina de biblioteca cria uma “referência externa não resolvida”
- Linkeditor analisa todo o código objeto procurando as “referência externa não resolvida”
- Para cada “referência externa não resolvida”
 - procura na biblioteca a rotina correspondente
 - quando encontra, substitui a linha de comando pelo código objeto encontrado na biblioteca



Linguagem de Programação

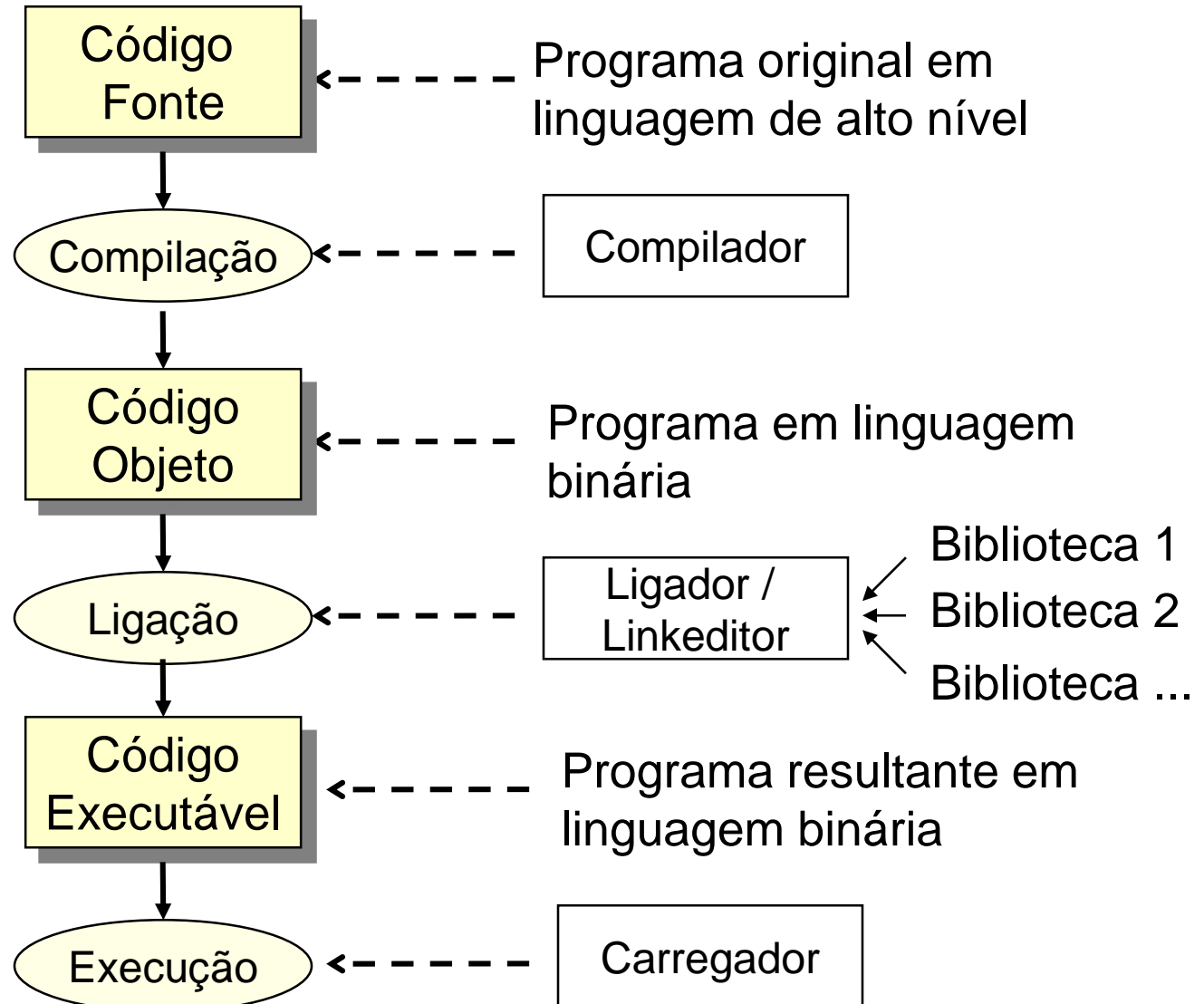
Processo de Execução

- Execução propriamente dita do código executável (do programa)
- Existem bibliotecas dinâmicas que são carregadas só na execução.
 - Processo realizado por um programa chamado **carregador** (ou *loader*)
- Existem carregadores (*loaders*) que fazem toda a linkedição e execução ao mesmo tempo (sem armazenar código executável, ou seja, o código é gerado, mas não armazenado)



Linguagem de Programação

Processo de Execução



Linguagem de Programação

Processo de INTERPRETAÇÃO

- Usado por linguagens de alto nível
- Execução das três fases (compilação, linkedição e execução) de uma só vez
 - processo chamado de **interpretação** realizado por um programa chamado **interpretador**
- ler comando → converte em código executável → executa (antes que o comando seguinte seja lido)
- Exemplo de linguagens: APL, ASP, ...

Linguagem de Programação

Compilação X Interpretação

COMPILAÇÃO

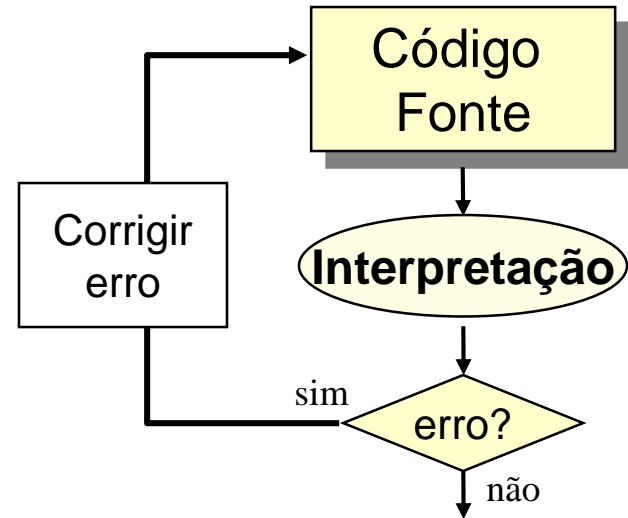
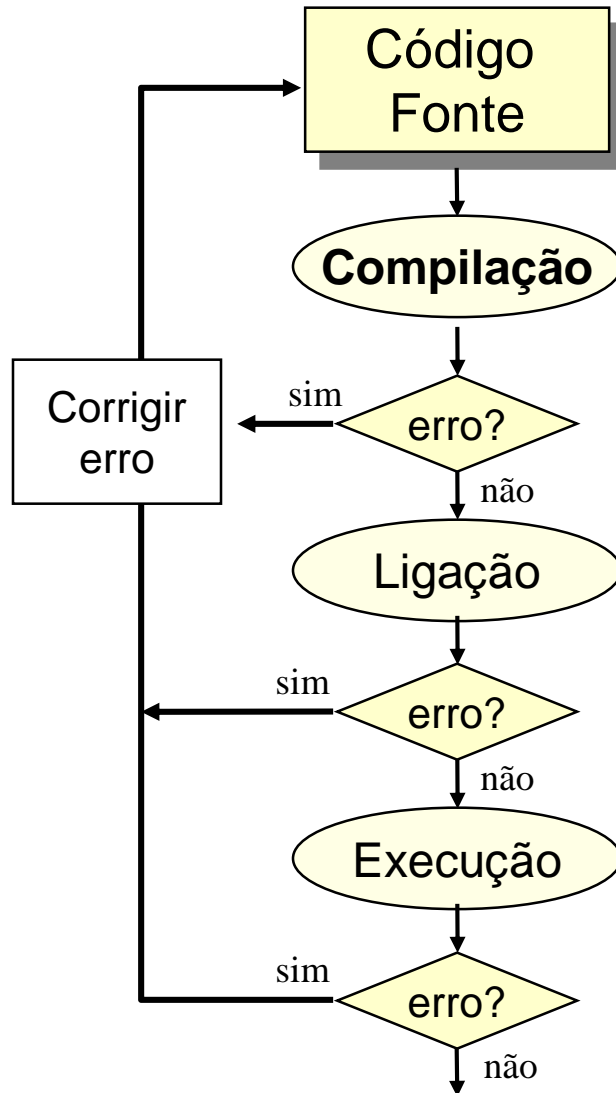
- menor consumo de memória
- transformação de partes comuns (rotinas de repetição) de uma única vez

INTERPRETAÇÃO

- maior consumo de memória
- necessidade de interpretação de partes comum (rotinas de repetição) várias vezes
- necessidade de carregar o código fonte na memória durante a execução
- execução mais lenta (necessidade de tradução, linkedição e execução)

Linguagem de Programação

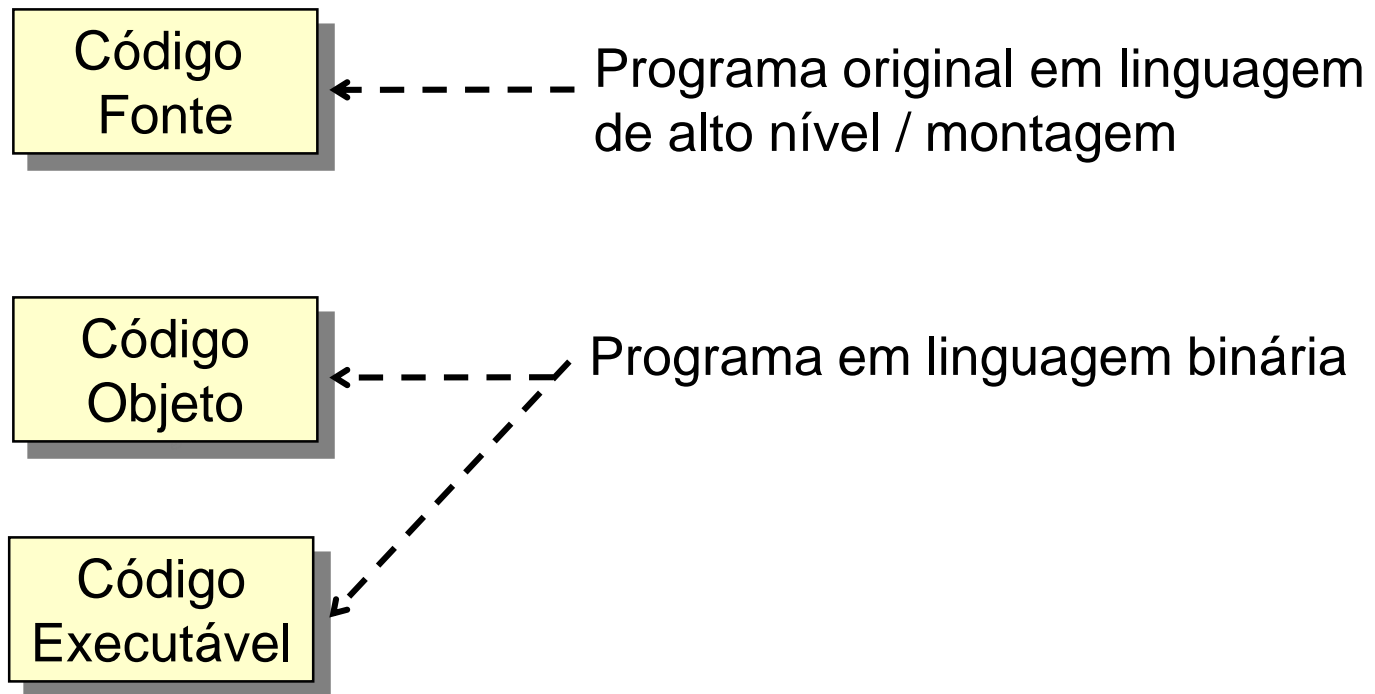
Processo de Depuração



Depurar ou debugger consiste no processo de procurar erros no programa e corrigi-lo.

Etapas de Desenvolvimento

Classificação dos Códigos



→ Também pode ser chamado de PROGRAMA Fonte, PROGRAMA Objeto e PROGRAMA Executável, ao invés de CÓDIGO.



Etapas de Desenvolvimento

- Desenvolver o algoritmo
 - compreender muito bem o problema
 - elaborar como solucionar passo a passo o problema, respeitando a “lógica de programação”
- Codificar o algoritmo
 - escrever o algoritmo em uma linguagem de programação
- INTERPRETAÇÃO
 - ou
 - COMPILAÇÃO {
 1. Compilar
 2. Linkeditar
 3. Executar



Referências de Criação para Apoio ao Estudo

Material para Consulta e Apoio ao Conteúdo

- FARRER, H. *et al.* Algoritmos Estruturados, Editora LTC, 3ª . edição, 1999.
 - Capítulo 0
- MANZANO, J.; OLIVEIRA, J. Algoritmos, Lógica para desenvolvimento de programação, Editora Ética, 1996.
 - Capítulos 1 e 2

