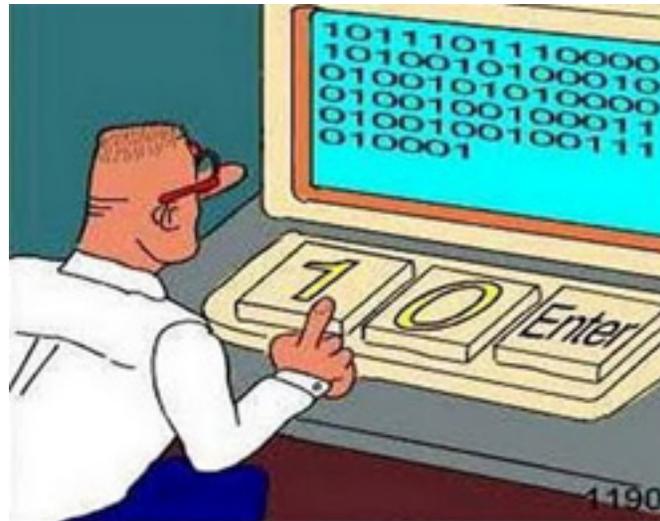


O que é um programa?

- Programa é uma lista de instruções que descrevem uma tarefa a ser realizada pelo computador.



Linguagem de Programação

- Uma linguagem de programação é um método padronizado para expressar instruções para um computador.
- É um conjunto de regras sintáticas e semânticas usadas para definir um programa de computador.

Como falar a mesma linguagem?

Calcule a soma dos
10 primeiros
números primos.



01001001001010010
01010100101010010
01010010100100111
???

Linguagem de Máquina

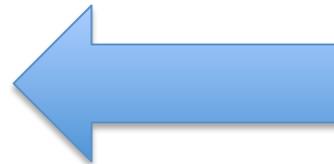
- Formada apenas por 0's e 1's. (Aula passada)
- Pode ser diretamente executada pelos circuitos do processador.
- No início da era da computação, os programas eram escritos diretamente em linguagem de máquina.



```
010100101010101  
010010101001010  
010101001010010
```

Linguagem de Máquina

- Um programa em linguagem de máquina é uma longa série de 0's e 1's:
 - códigos de instruções
 - dados a serem processados
 - endereço dos dados
- Código binário contém uma sequência de instruções a serem executadas



```
010100101010101  
010010101001010  
010101001010010
```

Codificar em linguagem de máquina?

- Difícil de ser escrito;
- Sujeito a muitos erros;
- Demanda muito tempo;
- Difícil de ser entendido para futuras modificações;
- Difícil de encontrar onde estão os erros;
- ...

Outros problemas

- Cada família de computadores possui sua própria linguagem de máquina.
- Um programa em linguagem de máquina é dedicado para um determinado computador.
- O código binário somente poderá ser executado em computadores da mesma família, ou seja, que sejam 100% compatíveis.

Linguagem de Montagem

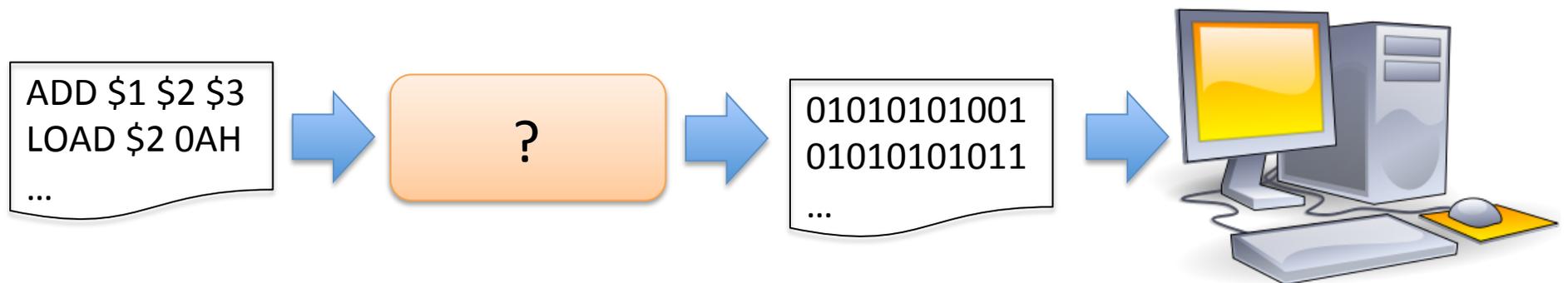
- Primeira idéia para resolver o problema
 - Criar uma linguagem em que os códigos numéricos fossem substituídos por mnemônicos.
- Exemplo:
 - LOAD = carregar e ADD = somar
- Mnemônicos se aproximam de palavras comuns da língua inglesa.
- As localizações dos dados (endereços na memória) fossem substituídas por referências simbólicas.
- Definição de regras de sintaxe de fácil memorização
 - Tornou a escrita e manutenção dos programas uma tarefa mais fácil, menos complexa.

Linguagem de Montagem

- Essa linguagem simbólica recebeu o nome de *Assembly Language*.
- Vantagens:
 - Desnecessário decorar os códigos numéricos que representavam as diferentes instruções e os endereços reais de armazenamento;
 - Conhecer apenas os mnemônicos para as instruções;
 - Definir nomes para as referências dos endereços.

Questões

- O computador não conhece apenas a linguagem binária?
- Usando mnemônicos, como fica a linguagem binária?
- Necessário um intermediário.



Tradutor

- Programa que recebe como entrada um programa escrito em uma linguagem de programação e produz como resultado as instruções deste programa traduzidas para linguagem de máquina.
- A tradução da linguagem *Assembly* para linguagem de máquina é feita pelo **Montador** (*Assembler*)

Linguagem de Montagem

- A linguagem *Assembly* facilitou o trabalho humano de programação, mas ainda exigia muito esforço.
- Necessário conhecimento profundo da máquina para a qual se estava programando.
- E então?

Linguagem de Programação

- Metas
 - Permitir que programadores tivessem uma maior produtividade;
 - Facilitar o entendimento humano;
 - Facilitar a correção de erros;
 - Tornar os programas menos dependentes de computadores ou ambientes computacionais específicos (portabilidade).

Linguagem de Programação

- Chamadas de linguagens de alto nível.
- Foram criadas diversas linguagens.



C#
Delphi
JAVA
PASCAL
COBOL
.NET
C++

Tradutor

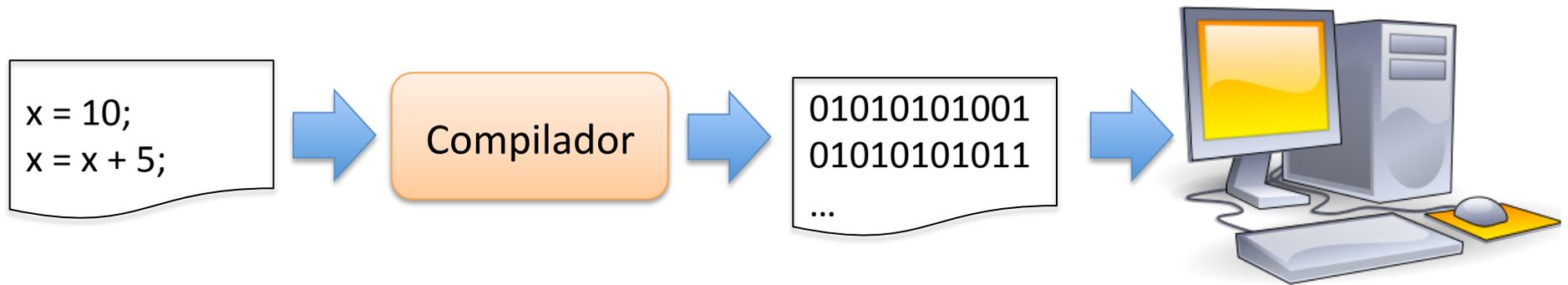
- Novamente precisamos da ajuda de um tradutor!
- Um programa escrito por um programador (chamado código fonte) em uma linguagem de alto nível é um conjunto de instruções que é clara para programadores, mas não para computadores.
- Programas em linguagem de alto nível também precisam ser traduzidos para linguagem de máquina.

Tradutores de Linguagem de Programação

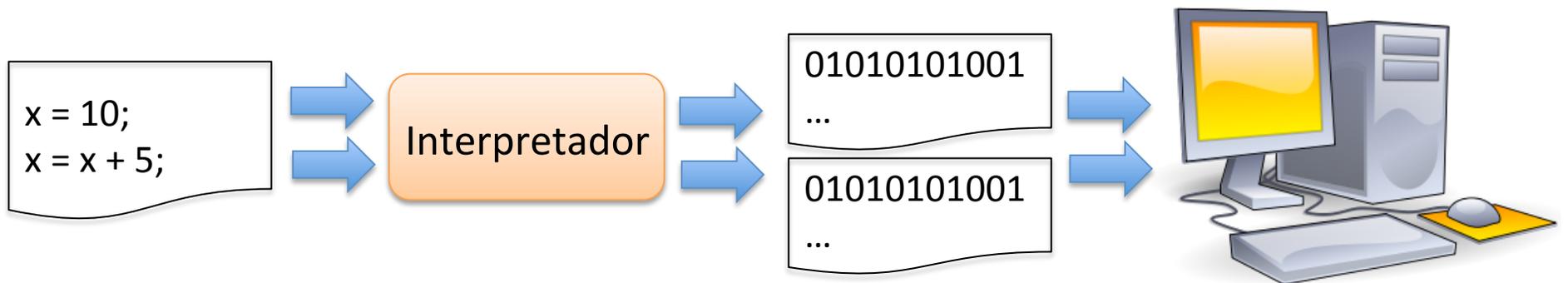
- **Compilador**
 - programa que traduz um programa escrito (código fonte) em uma determinada linguagem de programação para outra linguagem (linguagem destino).
- **Interpretador**
 - programa que traduz instrução por instrução de um programa em linguagem de máquina e imediatamente executa a instrução.

Compilação X Interpretação

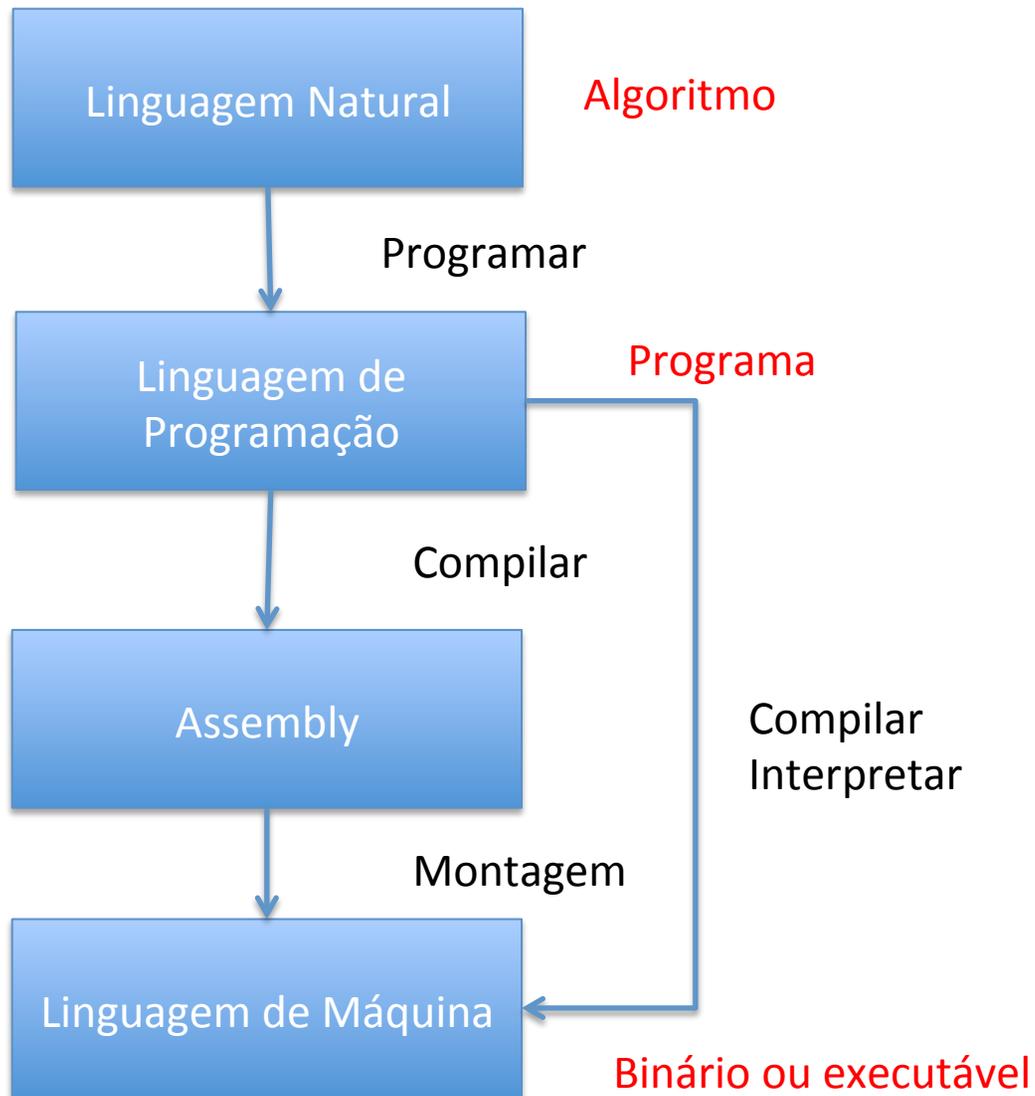
- Compilação



- Interpretação



Resumo



Comparação

- Tempo de execução
 - Uma vez compilado, enquanto o código fonte do programa não for alterado, ele poderá ser executado sucessivas vezes, sem necessidade de nova compilação.
 - Cada vez que um programa interpretado tiver que ser re-executado, todo o processo de interpretação deverá ser refeito, independentemente de ter havido ou não modificações no código fonte do programa desde sua última execução.

Comparação

- Consumo de Memória
 - O interpretador precisa permanecer na memória durante todo o tempo que durar a execução do programa, pois um programa necessita do interpretador para ter traduzidos cada um dos seus comandos, um a um, até o término de sua execução.
 - o compilador é carregado e fica na memória apenas durante o tempo de compilação, depois é descarregado. Quando o usuário for executar o programa, apenas o módulo de carga (código executável) é carregado e fica na memória durante a execução.
 - Desta forma, vemos que o método de interpretação acarreta um consumo de memória muito mais elevado durante a execução do programa.

Comparação

- Depuração de erros
 - na compilação, a identificação de erros durante a fase de execução fica sempre difícil, pois não há mais relação entre comandos do código fonte e instruções do executável.
 - na interpretação, cada comando é interpretado e executado individualmente, a relação entre código fonte e executável é mais direta e o efeito da execução (certa ou errada) é direta e imediatamente sentido. Quando a execução de um comando acarreta erro, quase sempre o erro pode ser encontrado no comando que acabou de ser executado.