

Compiladores



Introdução a Computação

Aula 06

Prof. Ma. Camilla

Linguagem

- As regras dividem-se em:

- Regras sintáticas

- Definem as formas corretas

- Regras semânticas

- Definem as condições para que as frases sintaticamente corretas possam ser interpretadas.

Linguagens naturais X Linguagens formais

- Naturais

- Utilizadas para as pessoas se comunicarem



- Formais/Artificial

- Comunicação Homem/Máquina

```
; Hello World for Intel Assembler (MSDOS)

mov ax,cs
mov ds,ax
mov ah,9
mov dx, offset Hello
int 21h
xor ax,ax
int 21h

Hello:
    db "Hello World!",13,10,"$"
```

Como se especifica uma linguagem

- Linguagens artificiais

- Têm a sua sintaxe e semântica rigorosamente definidas á custa de regras apropriadas.

- Regras sucinta e objetiva, para garantir a sua interpretação com facilidade e sem ambiguidade.

Processadores de Linguagens

Processadores de Linguagens

- Linguagens de Programação são notações para descrever processamentos computacionais, tanto para serem compreendidas por pessoas quanto por máquinas.
- Antes que possa rodar um programa precisa primeiro ser traduzido para um formato que permita ser executado por um computador.
- Essas linguagens são criadas com o objetivo de permitir que os programadores expressem algoritmos e instruções de forma precisa e estruturada, de modo que possam ser traduzidos em código executável pelos computadores.
- As linguagens de programação podem variar em termos de sintaxe, semântica e paradigmas, mas todas têm em comum essa função de servir como uma forma de comunicação entre os programadores e as máquinas.

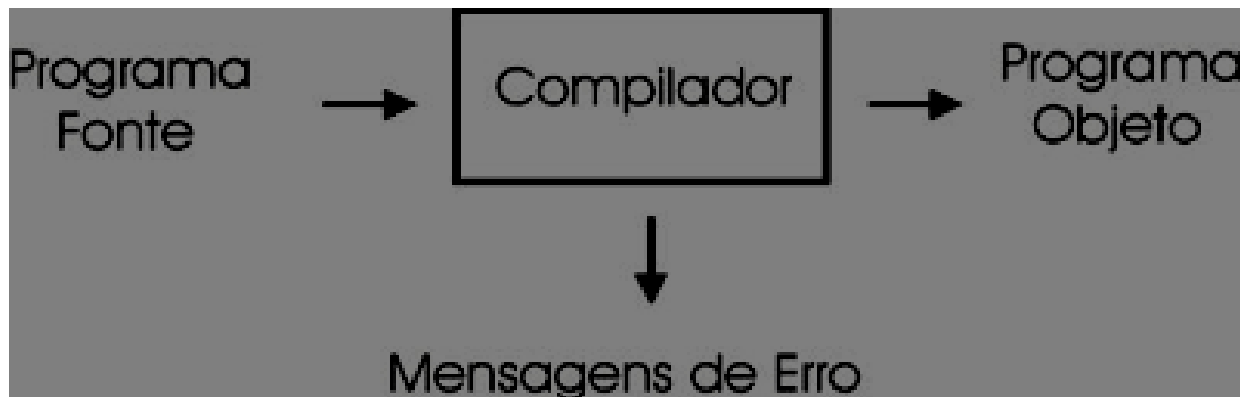
Processadores de Linguagens

—Antes que possa rodar um programa precisa primeiro ser traduzido para um formato que permita ser executado por um computador.

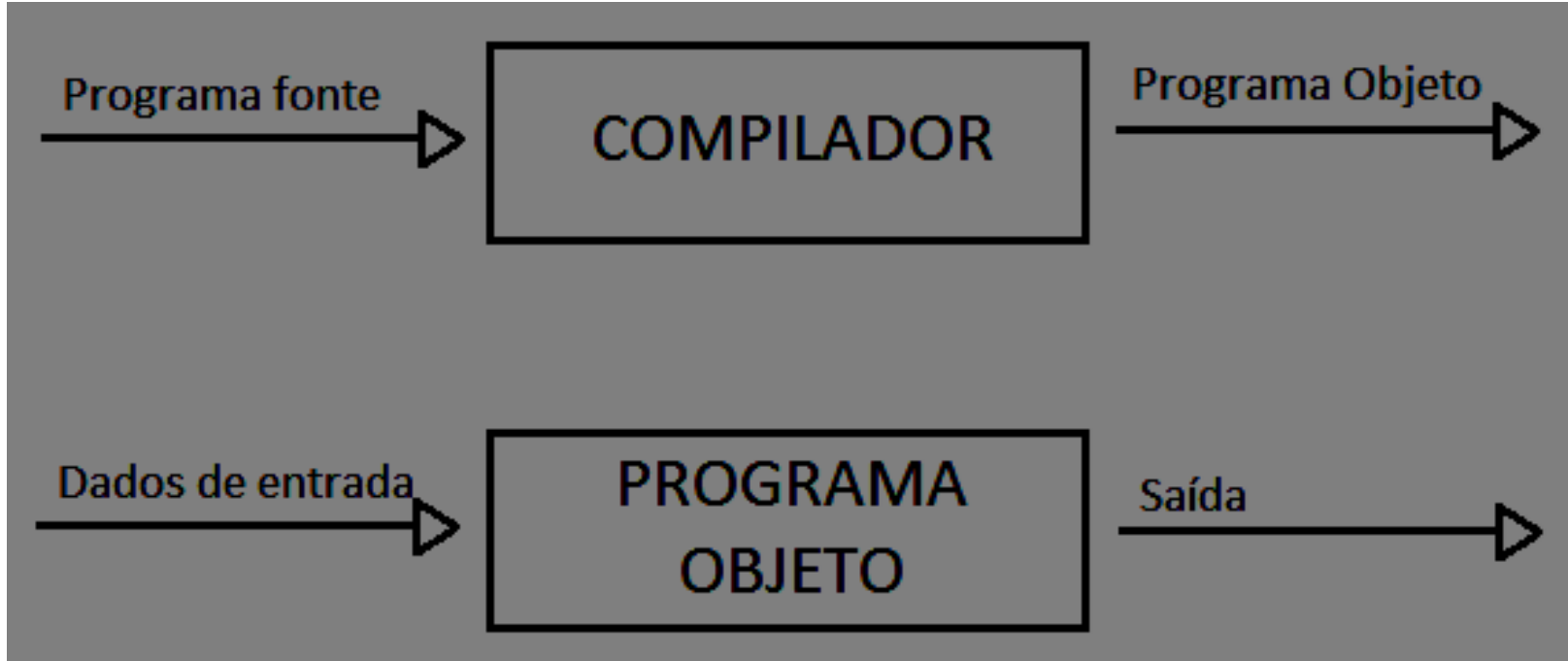
—Os sistemas de software que fazem a tradução são denominados **COMPILADORES**

Processadores de Linguagem

- Um compilador é um software que recebe como entrada um programa em uma linguagem de programação (**linguagem fonte**) e o traduz para um programa equivalente em outra linguagem (**linguagem objeto**).
- O compilador é um software separado projetado para traduzir o código-fonte de um programa em linguagem de programação para um código executável, para linguagem de máquina.



Processadores de Linguagem



Se o programa objeto for um programa em uma linguagem de máquina executável, ele é processado: entradas e produzir saídas

Compilação

Passa por 2 fases:

- Análise léxica (ou tokenização);
- Análise sintática (ou parsing).

Após essas duas fases, outros estágios da compilação incluem a análise semântica, a geração de código intermediário, a otimização e, por fim, a geração do código de máquina ou código executável.

Análise léxica (tokenização)

- o código-fonte do programa é dividido em unidades léxicas, conhecidas como tokens.
- Os tokens podem ser palavras-chave, identificadores, operadores, números, símbolos, entre outros.
- A análise léxica remove espaços em branco, comentários e formatação desnecessária do código-fonte, criando uma sequência de tokens (junção do código) bem definidos que serão usados nas etapas subsequentes.

Análise sintática (parsing)

- Após a análise léxica, os tokens são organizados em uma estrutura hierárquica conhecida como árvore sintática ou árvore de análise.
- Essa etapa verifica se a estrutura do programa está de acordo com a gramática da linguagem de programação.
- É realizado o reconhecimento de padrões e a verificação da correta organização dos tokens, aplicando regras gramaticais definidas pela linguagem.
- Essa análise sintática é fundamental para garantir que o código esteja sintaticamente correto antes de prosseguir para a próxima fase.

Análise Semântica

- É uma fase da compilação que ocorre após a análise sintática, tendo como objetivo garantir que o programa esteja semântica e logicamente correto, identificando erros como uso incorreto de variáveis, incompatibilidades de tipos, referências indefinidas, entre outros problemas.
- Nessa etapa, são realizadas verificações sobre o significado e a consistência das construções presentes no programa, sendo aplicada várias verificações: Declarações e escopo; Tipagem; Coerência de tipos; Regras específicas da linguagem.

Essa etapa é essencial para evitar erros e comportamentos indesejados durante a execução do programa compilado.

Etapa das Verificações da Sêmantica

- **Declarações e escopo:** Verificar se as declarações de variáveis, funções e outros elementos estão corretas e se estão sendo usadas de acordo com suas definições e escopos.
- **Tipagem:** Verificar se as operações estão sendo realizadas em tipos de dados compatíveis. Por exemplo, se uma operação de soma está sendo feita entre dois inteiros ou se uma variável está sendo usada corretamente em relação ao seu tipo.
- **Coerência de tipos:** Verificar se os tipos estão sendo usados corretamente nas expressões, atribuições e chamadas de função.
- **Regras específicas da linguagem:** Verificar se o código segue as regras e restrições específicas da linguagem de programação em uso.

Processadores de Linguagem

- Um *Interpretador* é outro tipo comum de processador de linguagem.
 - Executa diretamente as operações especificadas no programa fonte sobre as entradas fornecidas pelo usuário.
 - Um interpretador de linguagem é um programa de software responsável por executar o código-fonte de um programa de forma direta, eles só precisam escrever, testar, corrigir, escrever, testar e distribuir, sem a necessidade de uma etapa prévia de compilação.
 - Ao contrário de um compilador, que traduz o código-fonte em linguagem de máquina antes da execução, o interpretador lê e executa o código linha por linha, interpretando cada instrução em tempo real.

Em muitos casos, o interpretador lê linha a linha e converte em código objeto a medida de que vai executando.



Compilador X Interpretador

O **programa objeto** produzido por um compilador é muito mais rápido no mapeamento de entradas para saídas do que um Interpretador. Impossibilita ou pelo menos dificulta ser quebrado e visualizado o código-fonte original. Compila apenas se o código estiver sem erros.

Um Interpretador frequentemente oferece um melhor diagnóstico de erro do que o compilador, pois executa o **programa fonte** instrução por instrução. Correções e alterações são mais rápidas de realizar. Consomem menos memória.

Desvantagens

Compilador: O código passa por diversos níveis de compilação. Processo de correção, necessita que o código seja compilado novamente. Converte o programa de uma maneira a aperfeiçoar o uso da memória. Porém, ele transforma a linguagem fonte, em outras até chegar em Assembly, o que lhe custa certo tempo

Interpretador: Execução é mais lenta do programa. Necessita sempre ter lido o código original para ser executado. Apenas executa.

Com isso, observa-se que a linguagem compilada é mais rápida, do que a interpretada. Pois seu código já está compilado e 100% pronto para sua execução.

Compilador X Interpretador

- Linguagens que precisam de compiladores

- C; C++; BASIC; Cobol; Delphi; Fortran; Pascal; Visual Basic e entre outras.

- Linguagens que precisam ser interpretadas

- ASP; C#; Java; JavaScript; Lisp; Lua; PHP; Python; Ruby; VBScript e entre outras. Cada uma dessas linguagens possui um interpretador específico que pode ser usado para executar o código-fonte escrito nelas. Por exemplo, o Python possui o interpretador CPython, o Java possui a Máquina Virtual Java (JVM) e o JavaScript é interpretado pelos navegadores da web.

E o HTML?

HTML

HTML (HyperText Markup Language) não é considerado uma linguagem de programação, e sim uma linguagem de marcação.

Usado para estruturar e organizar o conteúdo de uma página da web, definindo elementos e sua semântica.

Não é interpretado no sentido tradicional, **pois não possui lógica de programação ou instruções para serem executadas.**

O HTML é interpretado pelos navegadores da web.

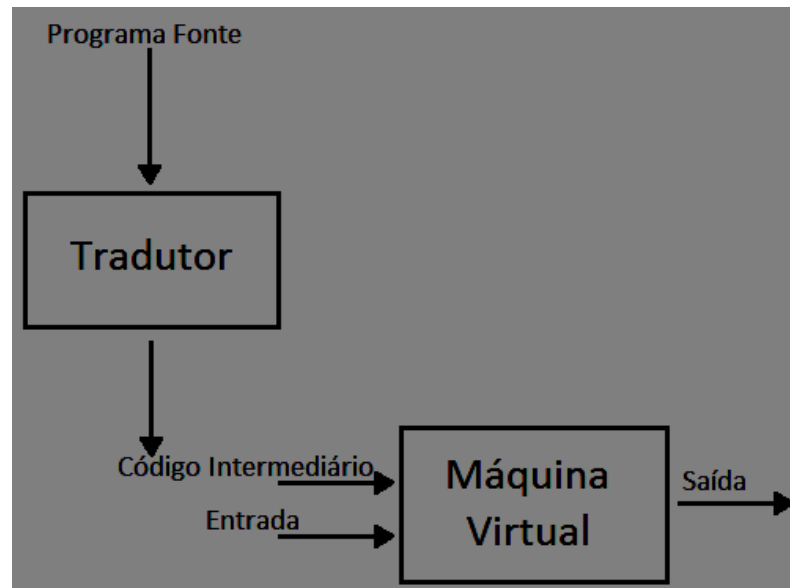
Quando um navegador encontra uma página HTML, ele interpreta o código e renderiza (transformar o código fonte) o conteúdo da página de acordo com as marcações HTML.

O navegador interpreta as tags HTML para exibir o texto, imagens, links e outros elementos na página.

No entanto, é importante destacar que, junto com o HTML, é comum usar o CSS, JavaScript, que é uma linguagem de programação interpretada pelos navegadores

Compilador Híbrido

- Os processadores da linguagem Java combinam compilação.
- Pode ser compilado para uma forma intermediária **bytecode** antes de ser interpretado pela JVM e interpretação.



Compilador Híbrido

Quais são os benefícios de um compilador Híbrido?

Compilador Híbrido

Os Bytecodes compilados em uma máquina podem ser interpretados em outra máquina, talvez por meio de uma rede.

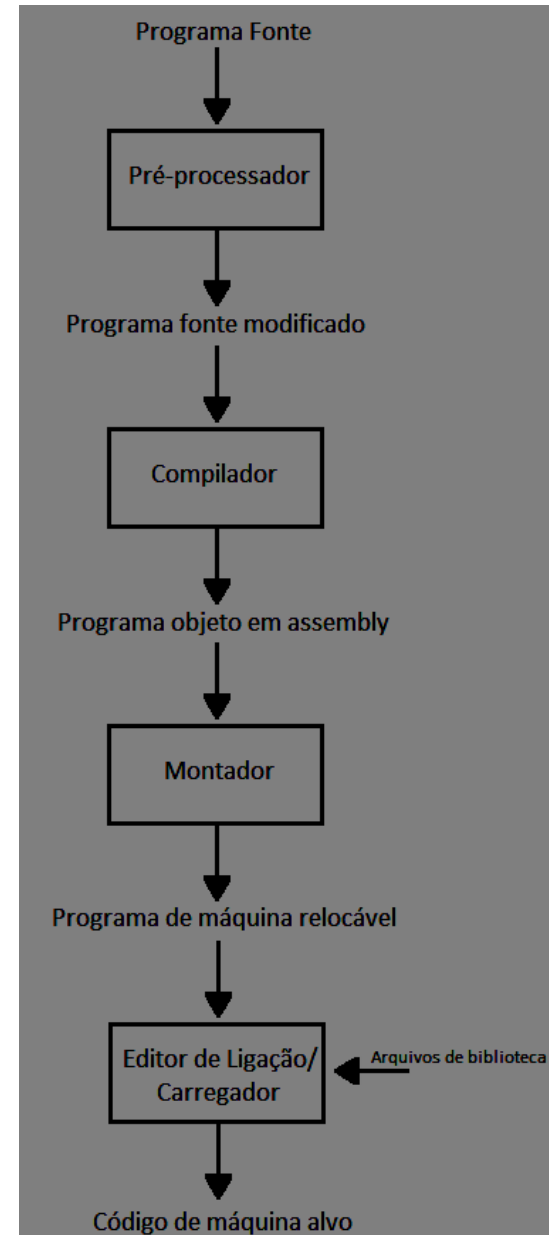
Onde entra a Máquina Virtual?

Um sistema de processamento de Linguagem

- Vários outros programas podem ser necessários para a criação de um programa objeto executável.
- Programa fonte pode ser subdividido em módulos de armazenamento em arquivos separados

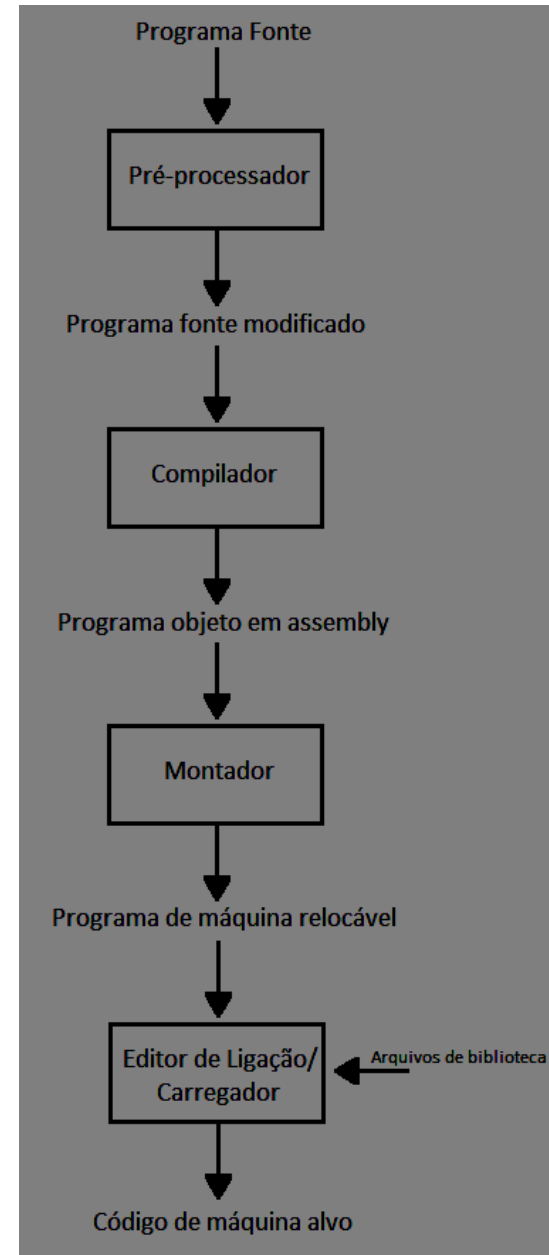
Um sistema de processamento de Linguagem

- O preprocessor coleta o programa fonte e modifica-os em partes para ser compilado.
- O compilador recebe na entrada o fonte modificado e pode produzir como saída um programa em uma linguagem simbólica (assembly)



Um sistema de processamento de Linguagem

- A linguagem simbólica é processada por um programa montador (assembler), considerada mais fácil de ser gerada como saída e mais fácil de depurar.
- Editor de Ligação resolve os endereços de memória externos.



Exercícios

Individual/Visto/Chamada

1. Qual é a diferença entre um compilador e um interpretador?
2. Quais são as vantagens de (a) um compilador em relação a um interpretador e (b) um interpretador em relação a um compilador?