



# LINGUAGENS FORMAIS E AUTÓMATOS

## INTRODUÇÃO À COMPILAÇÃO

**Artur Pereira** <artur@ua.pt>

DETI, Universidade de Aveiro

# SUMÁRIO

1 PROCESSAMENTO DE LINGUAGENS

2 GERAÇÃO DE CÓDIGO

3 ESTRUTURA DE UM COMPILADOR

# PROCESSAMENTO DE LINGUAGENS

- Processador de linguagem
  - Processa *texto* numa linguagem dada



- Compiladores, interpretadores e assembladores são processadores de linguagens
- É deles que vamos *falar*

# PROCESSAMENTO DE LINGUAGENS

- Compilador

- Converte programa-fonte num programa-alvo

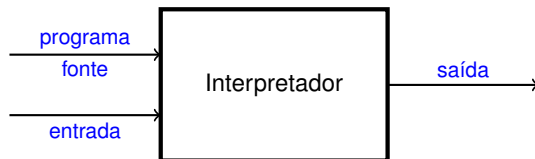


- O compilador deve reportar os erros que encontra durante a tradução
    - Programa alvo pode ser: código-máquina, código-intermédio, programa noutra linguagem de alto nível, ...
    - Se for código-máquina



# PROCESSAMENTO DE LINGUAGENS

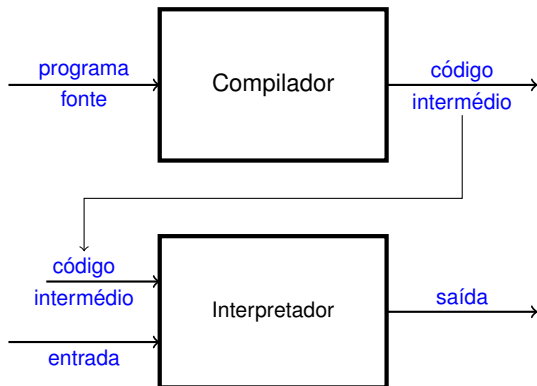
- Interpretador
  - Executa o programa-fonte



- O interpretador deve reportar os erros que encontra durante a tradução
- A execução é, em geral, muito mais lenta que usando código máquina.

# PROCESSAMENTO DE LINGUAGENS

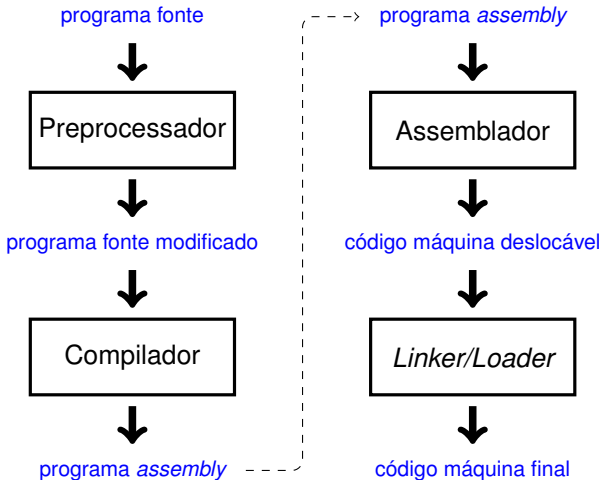
- Híbrido compilador-interpretador
  - Conversão para código intermédio e execução deste



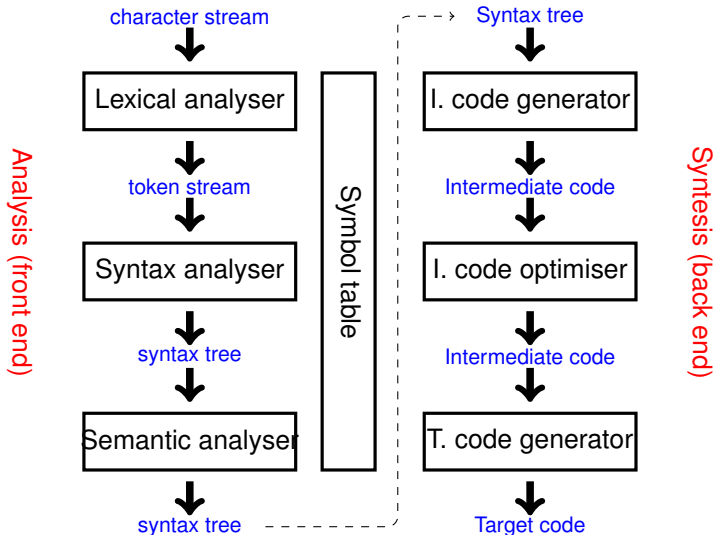
- O compilador deve reportar os erros que encontra durante a tradução
- compiladores JIT podem ser usado aquando da execução.

# GERAÇÃO DE CÓDIGO

- Fases da geração de código



# ESTRUTURA DE UM COMPILADOR





# ESTRUTURA DE UM COMPILADOR

- Lexical analysis
  - Convert the stream of characters into a sequence of lexemes/tokens
  - A lexeme/token is a tuple <token-name, attribute-value>
  - token-name is an abstract symbol representing a type of input
  - attribute-value represents the actual value of that element

- Example:

`pos = pos + vel * 5;`

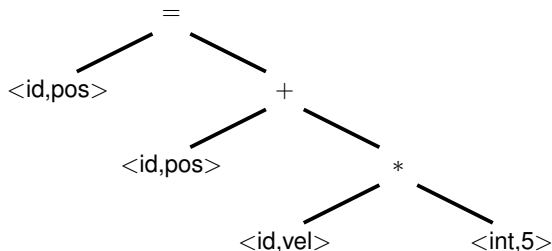
is converted to

`<id,pos> <=,> <id,pos> <+,> <id,vel>  
<*,> <int,5>`

- Typically, blanks are discarded by the lexical analyser

# ESTRUTURA DE UM COMPILADOR

- Syntax analysis
  - Converts the stream of tokens (first components) into a tree-like representation (syntax tree)
  - The tree captures the grammatical structure of the token stream
  - Syntax errors are detected and reported
  - Example:



# ESTRUTURA DE UM COMPILADOR

- Semantic analysis

- Checks semantic consistency based on syntax tree and symbol table
- Can apply type conversions
- Reports semantic errors
- Example:

