

COMPILADORES 2019/2020

aula 0x01 - Introdução

20/02/2020

Pedro Patinho <pp@di.uevora.pt>

Universidade de Évora - Departamento de Informática



SUMÁRIO

1. Contextualização
2. Construção de um compilador
3. Análise Lexical
4. Análise Sintáctica
5. Mais à frente...

CONTEXTUALIZAÇÃO

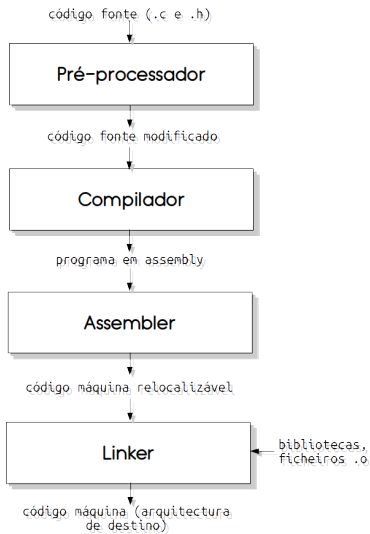
ALGUMA HISTÓRIA

- Compiladores inicialmente escritos em assembly
- Corrado Böhm (1951) desenvolve o primeiro compilador (linguagem, máquina e forma de tradução) na sua tese de doutoramento
- Donald Knuth (1965) inventou o *LR Parser* e provou que gramáticas livres de contexto $LR(k)$ podem ser transformadas em $LR(1)$ (determinismo)
- Frank DeRemer (1969) refinou o processo para SLR e LALR, pois um parser LR seria demasiado exigente para o hardware existente na época
- Isto levou a que as gramáticas para a maioria das linguagens de programação sejam definidas de forma a que possam ser analisadas por um *parser* LALR

O QUE É UM COMPILADOR?

- Não é um interpretador...
- É um *tradutor*.
- Mas preserva a equivalência semântica.
- Necessita de uma *gramática* (BNF ou EBNF)
- Pode fazer-se “à unha”, mas...

EXEMPLO: COMPILADOR DE C



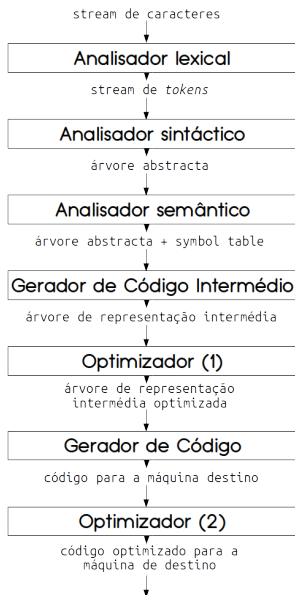
CONSTRUÇÃO DE UM COMPILADOR

COMO SE CONSTRÓI UM COMPILADOR?

Pipeline de processos independentes:

1. Analisador lexical
2. Analisador sintáctico
3. Analisador semântico
4. Geração de código intermédio
5. Optimizador (*machine-independent*)
6. Gerador de código
7. Optimizador (*machine-dependent*)

COMPONENTES DE UM COMPILADOR



Uso de “compiler compiler” tools:

- flex/bison
- JFlex/Cup
- ANTLR
- PyBison
- Muitas outras (**TPC: pesquisar**)

ANÁLISE LEXICAL

IDENTIFICAÇÃO DE SÍMBOLOS (LÍNGUA NATURAL)

- Agrupa letras em palavras (e sua classificação lexical)
- Numerais (literais numéricos)
- Pontuação
- Comentários (basicamente ignorados)

Tipos de tokens:

- Palavras reservadas (`if`, `return`, `for`, `int`, `char`)
- Identificadores/nomes (`a`, `printf`)
- Operadores e pontuação (`-` `==` `>` `;` `{` `}`)
- Literais (valores numéricos, strings)

EXEMPLO (PORTUGUÊS)

o gato comeu o rato
(artigo nome verbo artigo nome)

EXEMPLO (LINGUAGEM DE PROGRAMAÇÃO)

```
if a==b then x=1; else x=2;  
(kw var op var kw var op lit sep kw var op lit sep)
```

ANÁLISE SINTÁTICA

- Uso de gramática regular (BNF, EBNF, ...)

```
FRASE      := SUJEITO verbo PREDICADO
SUJEITO    := artigo nome | nome
PREDICADO  := artigo nome | nome | <vazio>
```

MAIS À FRENTÉ...

Análise do significado/sentido da frase:

O rato comeu o gato

```
a:string; if a==1 then print 'a is 1';
```

Também chamada *análise de nomes e tipos*.

REPRESENTAÇÃO INTERMÉDIA

- Reduzir o trabalho (do implementador) para metade
- Muitos compiladores \rightarrow mesma R.I.
- Mesma R.I. \rightarrow muitas máquinas destino

- Tornar mais rápido
- Usar menos memória
 - Uso de registos em vez de memória
 - Substituir variáveis por literais (quando são constantes)
- Gerar executáveis mais pequenos (por vezes, em detrimento das anteriores)

- Normalmente, código “low-level”
 - Assembly (MIPS, x86, x64, etc)
 - Bytecode (JVM, PHP/Zend, etc)
- Ou código “high-level” (descompiladores)
- Ou código em linguagens complementares (e.g., gerar SQL a partir de classes Java, etc.)