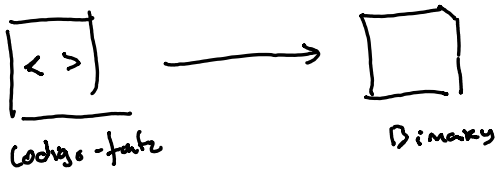
Como funcionam compiladores

1. Introdução
   - Breve apresentação do tema
2. Objetivos
   - Perceber o que é um compilador
   - Distinguir os tipos de compiladores
   - Entender as fases de compilação
3. O que é um compilador?
   - converte código fonte para binário.

Código → Binário (também podem ser objetos de código)

4. Tipos de compiladores
   - Native compiler
   - Bytecode compiler
   - Just-in-time compiler
   - Source-to-source compiler
   - Hardware compiler

5. Native compiler

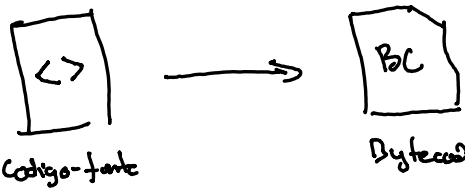
- 

Código-fonte Binary

  - Código Nativo específico da arquitetura do processador (ex: x86, ARM, MIPS, ...)

- Exemplos de Native compilers
  - GNU compiler Collection (GCC)
  - Microsoft Visual C++ (MSVC)
  - Clang + LLVM
  - ....

6. Bytecode compiler

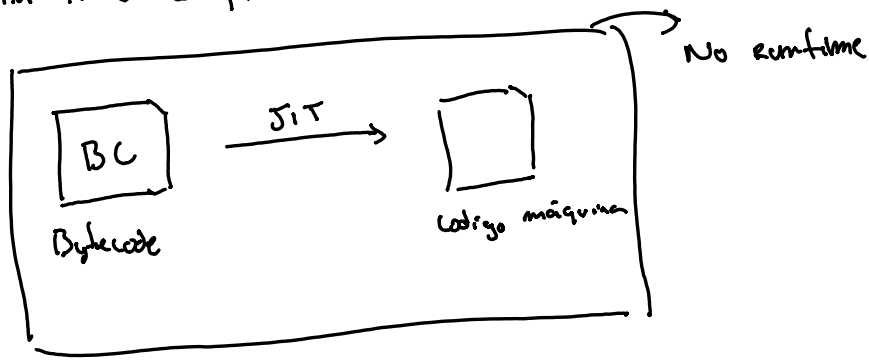
- 

Código-fonte Bytecode

  - Código Bytecode corre numa máquina virtual que interpreta estas instruções

- Exemplo Bytecode compilers
  - Roslyn C#
  - Java compiler
  - CPython
  - ...

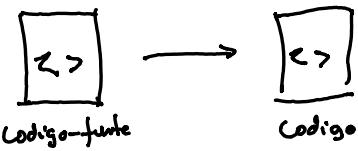
7. Just-in-time compilers



- JIT compilers correm em runtime
- Compilam pedaços ou todo o código
- Costumam optimizar rotinas usadas muito frequentemente.

- Exemplos de JIT compilers
  - PyPy Python
  - Java HotSpot
  - Lua JIT
  - ...

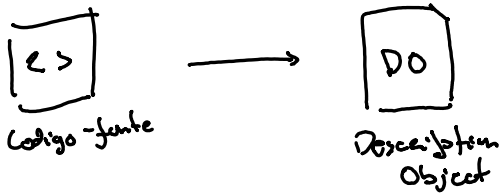
8. Source-to-source compilers



- Também conhecido por transpilador
- Converte código de uma linguagem para código de outra linguagem
- ...

- Exemplo de source to source compiler
  - Typescript para Javascript

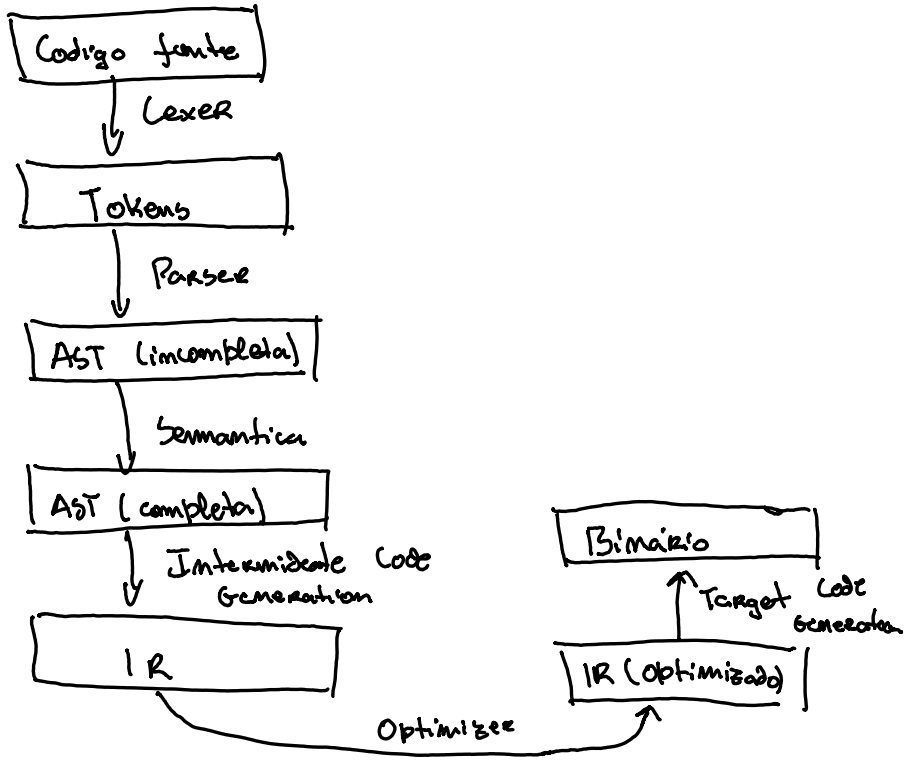
9. Hardware compiler



- Usado para programar FPGAs ou construir ASICs.

- Exemplo de um Hardware Compiler
  - GHDL

10. Fases de compilação



11. Lexer / Scanner

- Código-fonte → Tokens

12. Parser

- Tokens → Abstract Syntax Tree (AST)

13. Análise da semântica

- AST incompleta → AST completa
- Step mais complexo do compilador

14. Intermediate Code generation

- AST completa → IR (código intermédio)

15. Optimizer

- IR → IR (optimizado)
- Fase opcional

16. Target Code Generation

- IR → Assembly

17. Conclusão

- Recapitular os objetivos

18. Referências

- Livro Dragon compilers e Introduction to compilers