Como funcionam compiladores

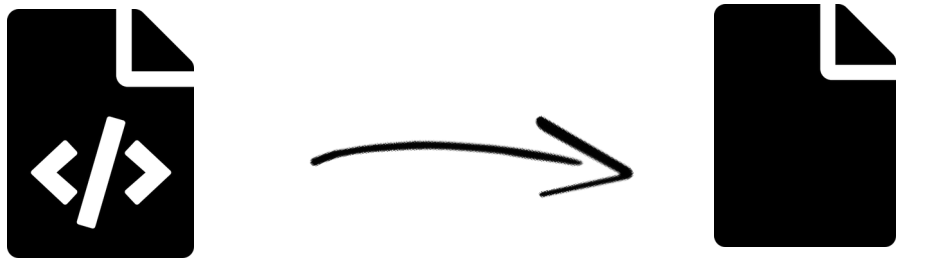
Desde o código-fonte ao binário

POLITÉCNICO DE LEIRIA

Objetivos

- Perceber o que é um compilador
- Distinguir os varios tipos de compiladores
- Entender sucintamente as várias fases de compilação

O que é um compilador?



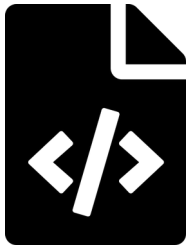
Código-fonte

Binário / Código

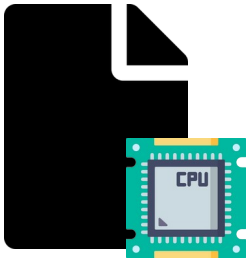
Tipos de compiladores

- Native compiler
- Bytecode compiler
- Just-in-time (JIT) compiler
- Source-to-source compiler
- Hardware compiler

Native compiler



Código-fonte



Binário Nativo

GNU Compiler Collection (GCC)

Clang + LLVM

Microsoft Visual C++ (MSVC)

Bytecode compiler

Código-fonte

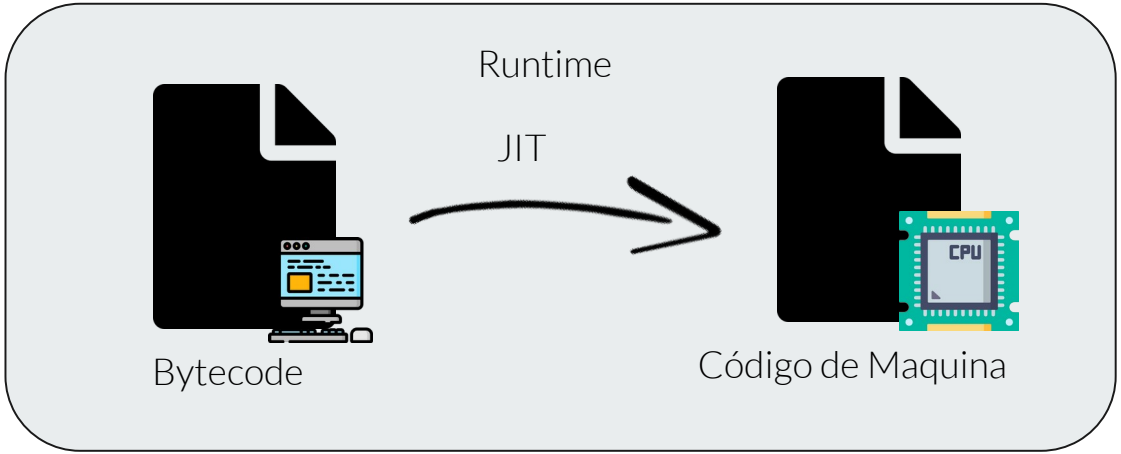
Bytecode

Javac
Compiler

Roslyn C#
Compiler

CPython Bytecode
Compiler

Just-in-time (JIT) compiler

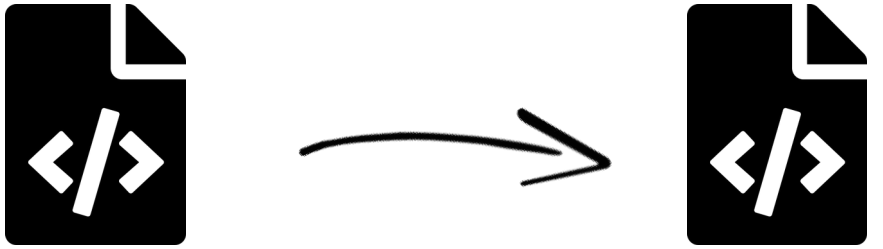


PyPy Python
Compiler

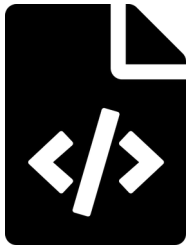
Java HotSpot
Compiler

LuaJIT
Compiler

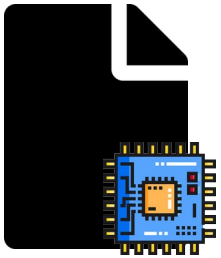
Source-to-source compiler



Hardware compiler



Código-fonte



Description Object

GHDL

Fases de compilação

Lexer / Scanner

```
# This program prints Hello,
world!
print('Hello, world!')
```

Código-fonte

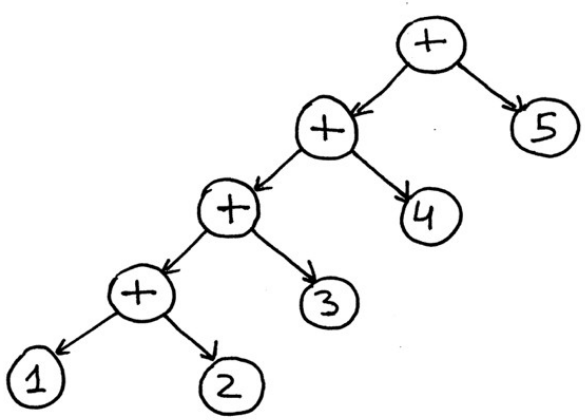
```
COMMENT
IDENTIFIER LPAREN STRING_LITERAL
RPAREN
```

Tokens

Parser

```
1 + 2 + 3 + 4 + 5
```

Expressão
(Em código-fonte)



Abstract Syntax Tree
(https://ruslanspivak.com/lsbasi-part7/)

Análise Semantica

```
auto foo(int bar)
{
    int foobar = 5;
    return foobar + bar;
}
```

Função
(Em código-fonte)

Nome	Tipo	*Scope*
`foo`	função, inteiro	global
`bar`	inteiro	parametro
`foobar`	inteiro	local

Symbol Table

Intermediate Code Generation

```
int foo(int bar)
{
      return bar + 5;
}
```

Função
(Em código-fonte)

```
%2 = alloca i32, align 4
store i32 %0, i32* %2, align 4
...
%4 = add nsw i32 %3, 5
ret i32 %4
```

Código Intermédio

Optimizer

```
%2 = alloca i32, align 4
store i32 %0, i32* %2, align 4
...
%4 = add nsw i32 %3, 5
ret i32 %4
```

Código Intermédio

```
...
%2 = add nsw i32 %0, 5
ret i32 %2
```

Código Intermédio Optimizado

Target Code Generation

```
...
%2 = add nsw i32 %0, 5
ret i32 %2
```

Código Intermédio Optimizado

```
lea eax, [rdi + 5]
ret
```

Código Máquina

Conclusão

- Importante saber como funcionam os compiladores
- Perceber as diferenças entre os vários tipos e as suas fases

Referências

- Aho, Sethi, Ullman, Compilers: Principles, Techniques, and Tools, Addison-Wesley, 1986. ISBN 0-201-10088-6
- Parsons, Thomas W., Introduction to Compiler Construction, Computer Science Press, 1992. ISBN 0-7167-8261-8