

Linguagens de Programação

Processos de Tradução

Carlos Bazilio

carlosbazilio@id.uff.br

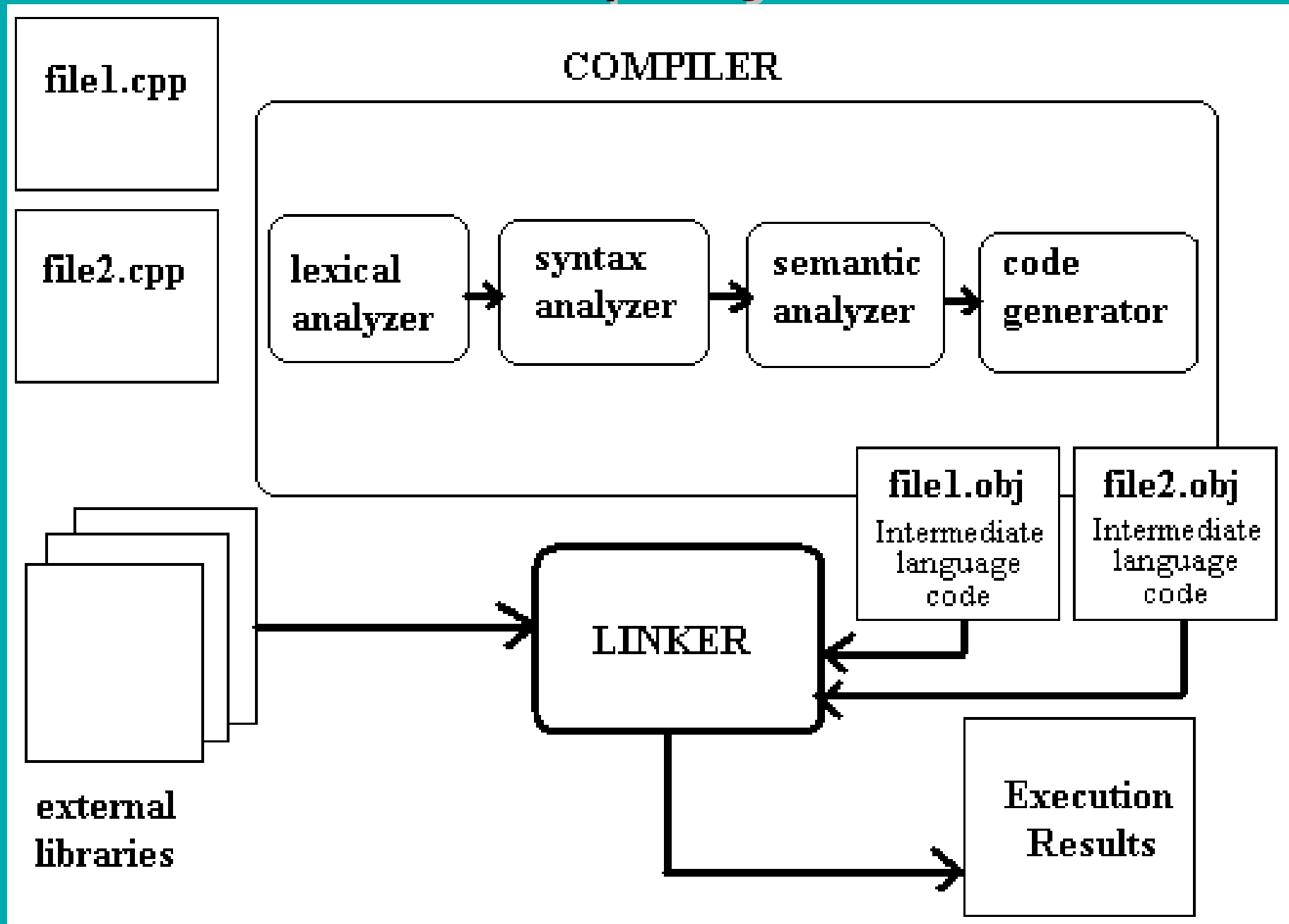
<http://www.ic.uff.br/~bazilio/cursos/lp>

Processos de Tradução

```
[>>> def alo():
[...     print("Alo mundo!")
[...     + Nova playlist
[>>> dis.dis(alo)
2          0 LOAD_GLOBAL          0 (print)
          2 LOAD_CONST          1 ('Alo mundo! ')
          4 CALL_FUNCTION        1
          6 POP_TOP
          8 LOAD_CONST          0 (None)
         10 RETURN_VALUE
```

- Um **programa** escrito numa linguagem é apenas um **arquivo texto** (um amontoado de palavras)
- Naturalmente, este amontoado tem seu valor reconhecido quando é **executado** efetivamente
- Chamamos de **processo de tradução** às etapas que vão desde o amontoado de **palavras** até um amontoado de **bits** (linguagem entendida pela máquina)

Compilação



Tipos de Bibliotecas

- Estática
 - São incluídas durante o processo de compilação
 - Ex.: bibliotecas .lib em C
- Dinâmica
 - Carregadas apenas durante a execução de uma aplicação e somente quando são referenciadas
 - Ex.: .dll no Windows



Tipos de Bibliotecas

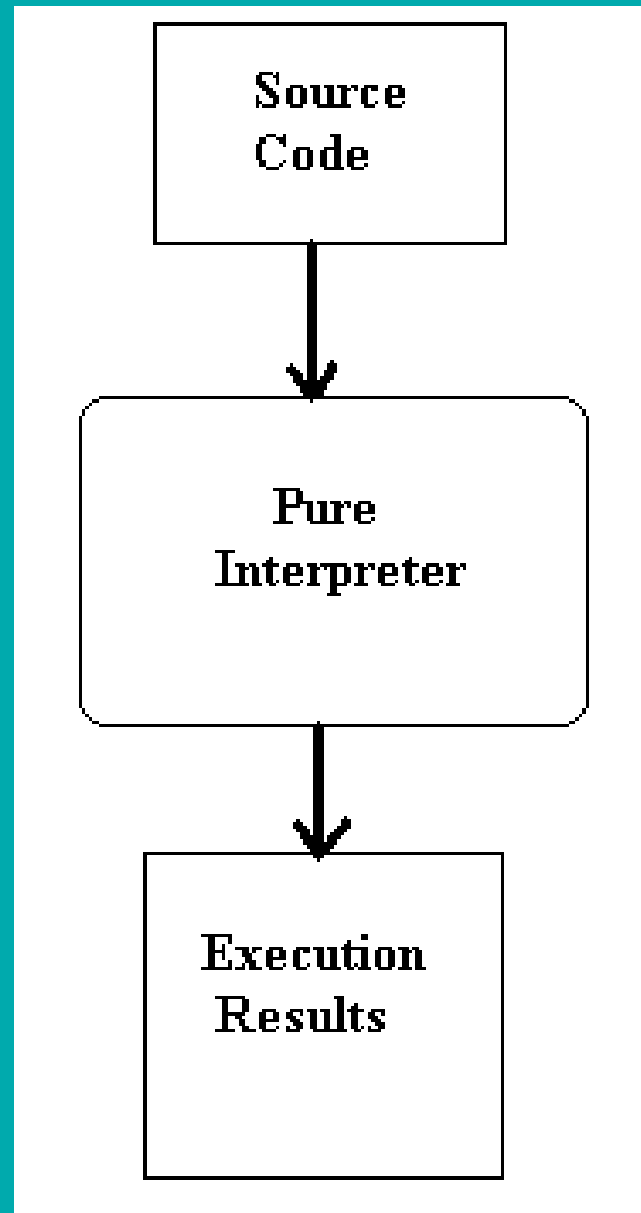
- Compartilhada
 - São acionadas também durante a execução e possuem versionamento, o que evita o conflito compartilhamento de bibliotecas por diferentes aplicações
 - Ex.: bibliotecas com extensão .so (shared objects) no Linux



Versionamento de Bibliotecas

- Suponha que uma biblioteca L, na sua versão 1.0, possui uma função: *int f(int)*;
- Em seguida, os desenvolvedores de L percebem que *f* poderia ser aprimorada adicionando-se um 2o. Parâmetro;
- Geram uma versão 1.1 de L com a seguinte versão de *f*: *int f(int, double)*;
- Quem usa a versão 1.0 de L pode atualizar seu ambiente para a versão 1.1?

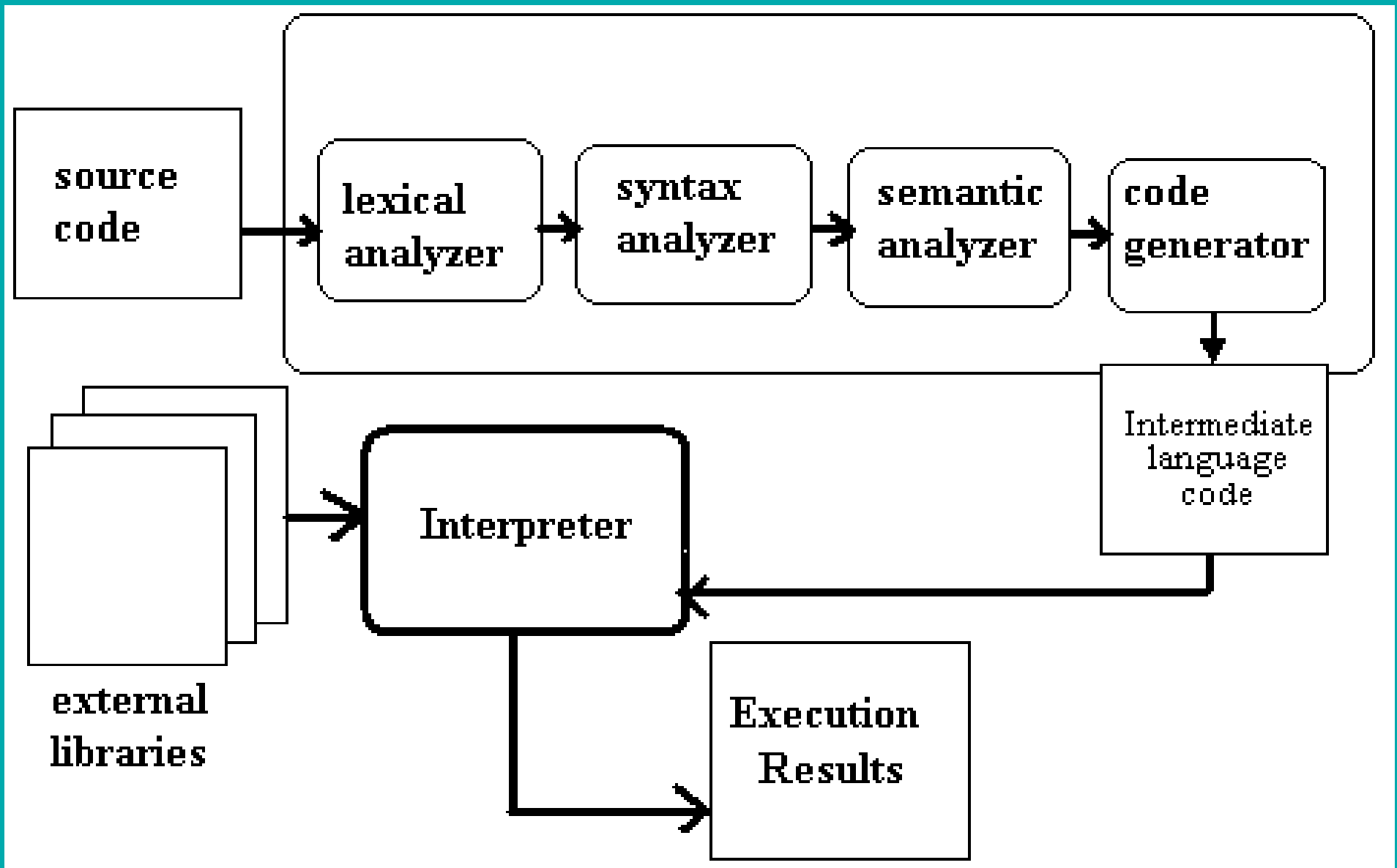
Interpretação



O código abaixo em C pode ser **interpretado?**

```
int x, y, z;  
x = 10;  
y = 20;  
z = x + y;
```


Tradução Mista



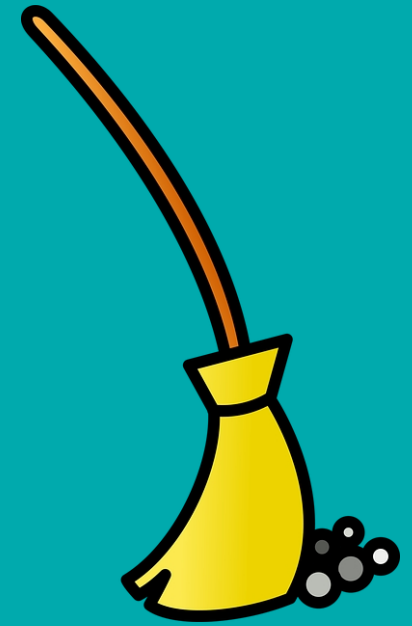
JIT (Just In Time) Compilation

- A interpretação usualmente é mais lenta que a compilação
- Uma maneira de torná-la mais eficiente é identificar trechos de código mais frequentemente utilizados e compilá-los durante a execução
- A esse mecanismo se dá o nome de Just In Time Compilation



Pré-processamento

- Etapa realizada antes da compilação/interpretação
- É utilizada para diversos fins:
 - Remoção de comentários (interpretação)
 - Reestruturação do código, quando possível, visando facilidade na análise (interpretação)
 - Processamento de diretivas de compilação, macros, etc (compilação)
 - http://en.wikipedia.org/wiki/C_preprocessor



Pré-processamento

```
1 // Programa Alo Mundo em C
2 // Autor: Prof. Bazilio
3
4 #include "stdio.h"
5
6 int main(void) {
7     printf("Alô mundo!\n");
8 }
```

```
1 #include "stdio.h"
2 int main(void){printf("Alô mundo!\n");}
```

```
[(base) Carlos-MacBook-Pro-3:C carlosbazilio$ ls -l alomundo*
-rwxr-xr-x  1 carlosbazilio  staff  8432 28 Abr 13:06 alomundo
-rw-r--r--@ 1 carlosbazilio  staff   117 28 Abr 13:05 alomundo.c
-rw-r--r--@ 1 carlosbazilio  staff    60 28 Abr 13:06 alomundopreproc.c
```

Otimização de Código



- Conjuntos de ações tomadas visando aumento de desempenho
- Podem ser executadas automaticamente pelos compiladores ou manualmente pelos programadores
- Programadores podem, por exemplo, otimizar loops que executam de forma demorada
- Compiladores podem otimizar tanto na geração de código intermediário, por exemplo, eliminando redundâncias do código fonte, como no nível de instruções de máquina

Tradução/Transformação de Código

- Processo de tradução de uma linguagem em outra
- Usualmente adequado para tratamento de sistemas legados
- Diferentemente dos processos de compilação, a transformação de código se dá entre linguagens com mesmo nível de abstração
- Ex. de linguagens: Babel, XSLT e TXL



Perguntas?

- Como são traduzidas as linguagens que você trabalha?
- Conhece algum outro exemplo de tradução de alto nível?
- Consegue pensar numa outra analogia para comparar Compilação e Interpretação?