

Фази на компајлер

Компајлери
Миле Јованов

1

Денес...

- Јазични процесори
- Многуге фази на компајлерот
 - Лексичка анализа
 - Синтаксна анализа (Парсирање)
 - Семантичка анализа (?)

Миле Јованов - Компајлери

2

Како се имплементирани јазичите?

- Две основни стратегии:
 - Интерпретери (постари, помалку проучувани)
 - Компајлери (понови, мн. повеќе проучувани)
- Интерпретерите ја извршуваат програмата "as is (како што е)"
 - Малку или нималку препроцесирање
- Компајлерите прават темелно препроцесирање

Миле Јованов - Компајлери

3

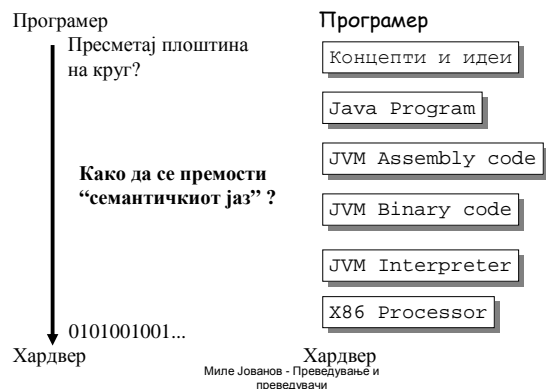
Имплементација на јазичи

- Доминираат системите за watch преведување
 - пр., gcc
- Некои јазичи примарно се интерпретираат
 - пр., Java bytecode
- Некои околии (Lisp) нудат
 - Интерпретер за развој
 - Компајлер за продукција

Миле Јованов - Компајлери

4

Зошто имаме потреба од јазични процесори?



5

Јазични процесори: Што се тие?

Процесор на програмски јазик е секој систем (софтверски или хардверски) кој манипулира со програмите.

Примери:

- Едитори
- Преведувачи (транслатори)*
 - Компајлери
 - Интерпретери

***Можете ли да ме преведете преку улица?**

Миле Јованов - Компајлери

6

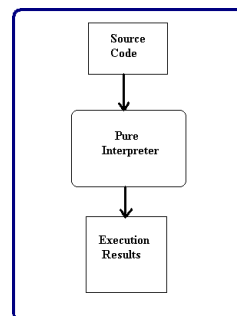
Процесирање на ПЈи

- Секој систем за процесирање на ПЈи, нивно извршување или подготовка за извршување се вика јазичен процесор
- Јазичните процесори вклучуваат *транслатори* и дополнителни алатки како синтаксно-водени едитори.
- Транслатор кој директно ја извршува програмата – **интерпретер**, додека транслатор која ја менува програмата во форма соодветна за извршување – **компајлер**
- Питоинаку кажано, интерпретацијата е процес во еден чекор во кој програмата и влезот му се достапни на интерпретерот, а излез е резултат на интерпретацијата:

Миле Јованов - Компајлери

7

Интерпретер

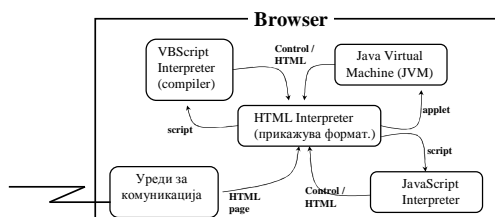


Миле Јованов - Компајлери

8

Интерпретерите секојдневно се употребуваат

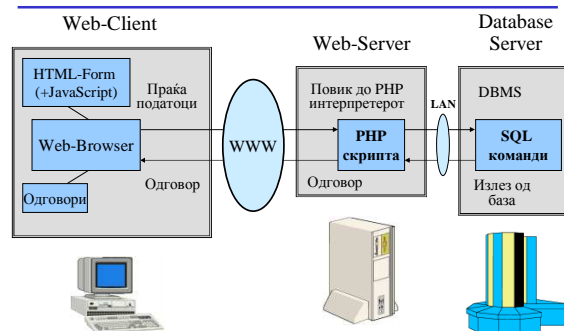
Пример:



Миле Јованов - Компајлери

9

И на web исто така:

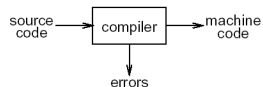


Миле Јованов - Компајлери

10

Компилација (преведување)

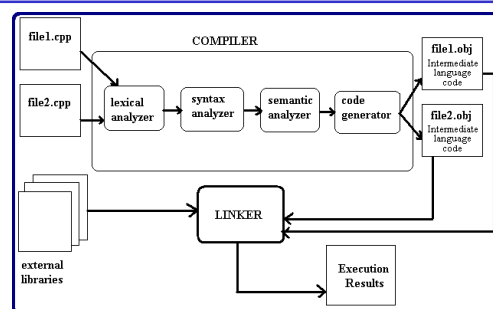
- **Компилацијата** е процес во барем 2 чекора, во кој, оригиналната програма (source) се внесува во компајлерот, а нова програма (target program) е излез од компајлерот.



Миле Јованов - Компајлери

11

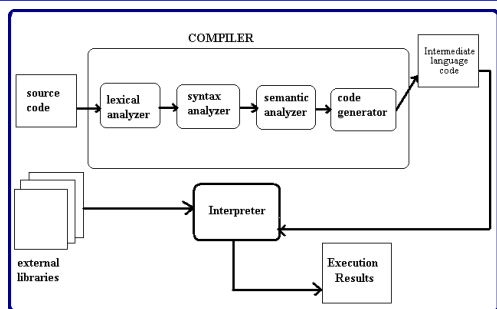
Компајлер



Миле Јованов - Компајлери

12

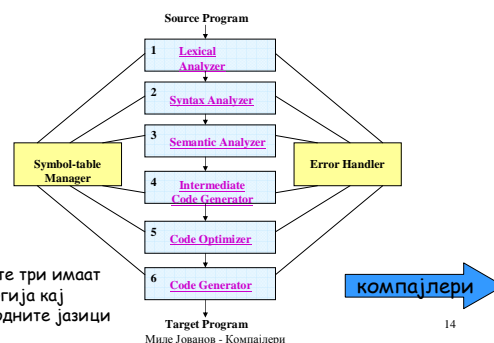
Хибрид: компјалер/интерпретер



Миле Јованов - Компјалери

13

Многуге фази на компјалерот



Првите три имаат аналогија кај природните јазици

Миле Јованов - Компјалери

14

Лексичка Анализа

- Прв чекор: препознај ги зборовите.
 - Најмала единица над буквите

Ова е реченица.

- Забележете
 - Голема буква "O" (симбол за реченичен старт)
 - Бланко " " (одделувач на зборови)
 - Точка "." (симбол за реченичен крај)

Миле Јованов - Компјалери

15

Уште лексичка анализа

- Таа не е тривијална. Пример:
ако на тртпрт ке стр она
- И програмските јазици се типично покриптирачки од природните:

*p->f ++ = -.12345e-5

Миле Јованов - Компјалери

16

И уште Лексичка анализа

- Лексичкиот анализатор (скенерот) го дели текстот на програмата на "зборови" или "белези (tokens)"

if x == y then z = 1; else z = 2;

- Делови:

if, x, ==, y, then, z, =, 1, :, else, z, =, 2, ;



Миле Јованов - Компјалери

17

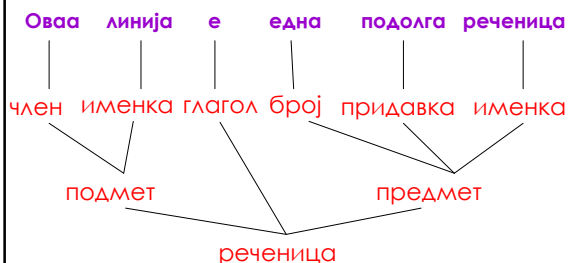
Парсирање

- Откако ќе ги разбереме зборовите треба да ја разбереме структурата на реченицата
- Парсирање = Дијаграмирање на речениците
 - Дијаграмот е дрво

Миле Јованов - Компјалери

18

Дијаграмирање



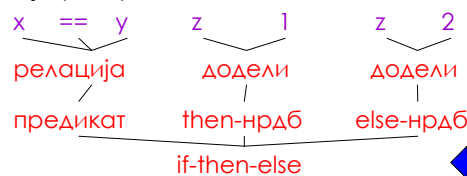
Миле Јованов - Компјалери

19

Проблеми при парсирање

- Парсирањето на изрази е исто
- Разгледуваме:

$$\text{If } x == y \text{ then } z = 1; \text{ else } z = 2;$$
- Дијаграмирано:



Миле Јованов - Компјалери

20

Семантичка анализа

- Кога ќе ја разбереме структурата, се обидуваме да го разбереме „значењето“,
 - Но, значењето е тешко за компјалерите
- Компјалерите изведуваат само ограничена анализа за пронаоѓање на неконзистентност
- За да се подобри изведбата на програмата некои прават поопсежна анализа

Миле Јованов - Компјалери

21

Семантичка анализа во природен јазик

- Пример:

$$\text{Го видов човекот со телескопот.}$$
 Кај кој е телескопот?
- Или:
 Виктор рече Виктор си ја заборави домашната?
 Колку Виктори има?
 Кој си ја заборави домашната?
 Домашната што?

Миле Јованов - Компјалери

22

Семантичка анализа

- ПЈ дефинираат правила за избегнување на двосмислености
 - Овој C++ код печати "4"; искористена е внатрешна дефиниција
- ```

{
 int Jack = 3;
 {
 int Jack = 4;
 cout << Jack;
 }
}

```

Миле Јованов - Компјалери

23

## Уште семантичка анализа

- Компјалерите прават и други проверки освен едначењето на променливите
- Пример:  
 Виктор ја остави нејзината домашна дома.
- "Промашен ТИП"  
 - Помеѓу неа и Виктор; тоа се различни луѓе  
 - Под претпоставка дека Виктор е машко

Миле Јованов - Компјалери

24

### Пример на семантички проверки во ПЈи

---

- Дефинирање на променлива пред употреба
- Дефинирање на променлива еднаш
- Компатибилност на типови
- Соодветни аргументи за функциите
- Константите не се променети
- Нема повратни врски во наследувањето
- ...



Миле Јованов - Компајлери

25

### За следниот час:

---

- Многуге фази на компајлерот
  - Семантичка анализа (?)
  - „Меѓукод“
  - Оптимизација
  - Генерирање на код
- Премини кај еден компајлер

Миле Јованов - Компајлери

26