

## Фази на компајлер 2

Компајлери  
Миле Јованов

1

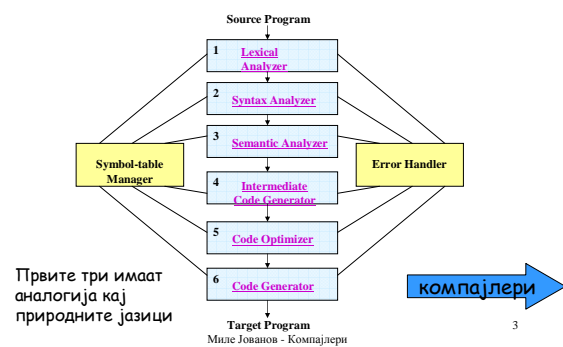
### Денес...

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

Миле Јованов - Компајлери

2

### Многуге фази на компајлерот



3

### Семантичка анализа

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

Миле Јованов - Компајлери

4

### Семантичка анализа во природен јазик

- Пример:  
*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?

5

Миле Јованов - Компајлери

### Семантичка анализа

- ПТЈ дефинираат правила за избегнување на двосмислености

```
{
    int Jack = 3;
    {
        int Jack = 4;
        cout << Jack;
    }
}
```
- Овој C++ код печати "4"; искористена е внатрешна дефиниција

Миле Јованов - Компајлери

6

### Уште семантичка анализа

- Компајлерите 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 - Компајлери

7

### 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

8

### Оптимизација

- Нема пандан во природниот јазик, едитирање...цензура ☺...
- Автоматска промена на програмите за тие да:
  - Се извршуваат побрзо
  - Користат помалку меморија
  - Во глобала, заштедат ресурс

Миле Јованов - Компајлери

9

### Пример за оптимизација

$X = Y * 0$  е исто што и  $X = 0$

НЕ!

Точно е за integer-и, но не за floating point number-и

Миле Јованов - Компајлери

10

### Примери за оптимизации во ПЈи

- Елиминација на мртов код
- Евалуација на повторливи изрази само еднаш
- Замена на израз со поедноствен, еквивалентен на него
- Пресметување на изрази за време на преведувањето
- Вадење на константните изрази надвор од циклус
- ...

Миле Јованов - Компајлери

11

### Генерирање на код

- (Најчесто) Произведува асембли код
  - $A := B + C$ 
    - LOAD B
    - ADD C
    - STORE A
- Преведување во друг јазик
  - Аналогно како природното преведување

Миле Јованов - Компајлери

12

## „Меѓу“ јазици

- Многу компајлери преведуваат помеѓу повеќе последователни јазични форми
  - Сите освен првиот и последниот се МЕЃУ јазици за внатрешна употреба
  - Најчесто има еден МЈ
- МЈи се подредени од повисоко кон пониско ниво на апстракција
  - Највисоко е source-от
  - Најниско е assembly-то

Миле Јованов - Компајлери

13

## Меѓу јазици

- МЈи се корисни оти пониските нивоа ги откриваат деловите скриени од погорните нивоа
  - регистри
  - Меморија
  - итн.
- Но, пониските нивоа го задржуваат значењето од повисоките (??)

Миле Јованов - Компајлери

14

## Компајлери

- Дизајнот на јазикот многу влијае на компајлерот
  - Одредува што лесно, а што тешко ќе се компајлира
  - Проектот: компромиси во дизајнот на јазикот

Миле Јованов - Компајлери

15

## Премини кај компајлерот

- Премин е целосно поминување преку изворната програма или преку некоја нејзина внатрешна репрезентација
- Премин може да одговара на фаза - ама не мора
- Понекогаш еден „премин“ одговара на повеќе фази кои се преклопени некако
- Кои и колку премини ќе направи компајлерот при компилацијата е многу значајно прашање

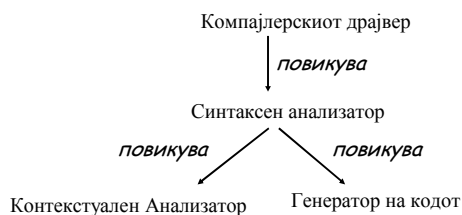
Миле Јованов - Компајлери

16

## Компајлер со еден премин (КЕТП)

Овој компајлер со еден премин го скенира, парсира текстот и истовремено генерира код.

Дијаграм на зависност на ваков компајлер:



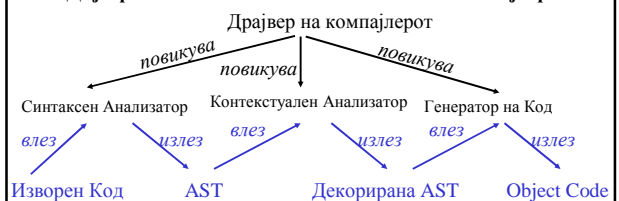
Миле Јованов - Компајлери

17

## Компајлер со повеќе премини (КТПП)

Овој компајлер прави повеќе премини. Излезот од еден премин се запаметува во некоја податочна структура за да може понатаму да се користи.

Дијаграм на зависност на типичен ваков компајлер:



Миле Јованов - Компајлери

18

## Дизајн на компајлери - прашања

	Еден премин	Повеќе премини
Брзина	подобро	полошо
Меморија	Подобро за поголеми програми	(ев.) подобро за мали програми
Модуларност	полошо	подобро
Флексибилност	полошо	подобро
“Глобална” оптимиз.	невозможно	возможно
Изворен јазик	КЕП не се можни за некои јазици	

Миле Јованов - Компајлери

19

## Јазик - прашања

### Пример Pascal:

Pascal е експлицитно дизајниран да лесно се имплементира со КЕП:

- Секој идентификатор мора да се декларира пред да се користи.

```
var n: integer;
procedure inc;
begin
  n = n + 1;
end;
var n: integer;
```

Миле Јованов - Компајлери

20

## Јазик - прашања

### Пример Pascal:

- Секој идентификатор мора да се декларира пред да се користи.
- Како се справуваме со врзаната рекурзија?

```
procedure ping(x: integer)
begin
  ... pong(x-1); ...
end;
procedure pong(x: integer)
begin
  ... ping(x); ...
end;
```

Миле Јованов - Компајлери

21

## Јазик - прашања

### Пример Pascal:

- Секој идентификатор мора да се декларира пред да се користи.
- Како се справуваме со врзаната рекурзија?

```
forward procedure pong(x: integer)
procedure ping(x: integer)
begin
  ... pong(x-1); ...
end;
procedure pong(x: integer)
begin
  ... ping(x); ...
end;
```

Миле Јованов - Компајлери

22

## Јазик - прашања

### Пример Java:

- идентификаторите може да се декларираат пред да се користат.
- затоа Java компајлерот има потреба од барем 2 премина

```
Class Example {
  void inc() { n = n + 1; }
  int n;
  void use() { n = 0; inc(); }
}
```

Миле Јованов - Компајлери

23

## Запаметете

Многу прашања влијаат на дизајнот на нов програмски јазик:

- Избор на парадигма
- Синтаксни преференци
- Имплементација на компајлерот
  - Пр. Број на премини
  - Расположливи алатки

Миле Јованов - Компајлери

24

## Запаметете

---

Многу прашања влијаат на дизајнот на нов компајлер:

- Бројот на премини
- Изворниот, целниот и јазикот за имплементација
- Распожливите алатки

## За следниот час:

---

- Една весела тема:
  - Bootstrapping