

Програмски јазици и компајлери

Компајлери
Миле Јованов

1

Денес...

- Кратка историја на програмските јазици
- Критериуми за дизајн на ПЈ
- Имплементација на ПЈ

Миле Јованов - Компајлери

2

Разлика помеѓу говорен и програмски јазик

- Компјутерскиот јазик нема двосмисленост и нејасност
- На македонски: *Го видов човекот со телескопот.* (Кај кој е телескопот?) или *Земи троа сол* (Колку е тоа „троа“?)
- Во ПЈ реченицата или значи едно нешто или не значи ништо

Миле Јованов - Компајлери

3

Нивоа на ПЈ

„Виша“ програма

```
class Triangle {  
    ...  
    float surface()  
    return b*h/2;  
}
```

„Ниска“ програма

```
LOAD r1,b  
LOAD r2,h  
MUL r1,r2  
DIV r1,#2  
RET
```

Извршен машински код

```
0001001001000101  
0010010011101100  
10101101001...
```

Миле Јованов

4

(Кратка) Историја на вишите ПЈ

- 1953 IBM го развива 701
- Се програмира во assembly
- Проблем: Трошоците за софтвер ги надминуваат хардверските
- John Backus: "Speedcoding"
 - Интерпретер (интерпретатор, толкувач)
 - 10-20 пати побавен од рачно пишуваниот assembly

Миле Јованов - Компајлери

5

FORTRAN I

- 1954 IBM го развива 704
- John Backus
 - Идеја: преведување на Виш код во assembly
 - Многумина велат - невозможно
 - Веќе неколку неуспешни проекти
- 1954-7 FORTRAN I проект
- До 1958, >50% од целиот софтвер е во FORTRAN
- Времето на развој е драматично скратено
 - (2 недели ! 2 часа)

Миле Јованов - Компајлери

6

FORTRAN I

- Првиот компајлер
 - Произведува код добар скоро како и рачно пишувањето
 - Големо влијание врз информатиката (CS)
- Модерните компајлери ги имаат зачувано аспектите на FORTRAN I

Миле Јованов - Компајлери

7

Историја на идеи: Апстракција

- Апстракција = занемарување на конкретни детали
- Неопходна е за изградба на софтверски системи
- Видови на апстракција:
 - Преку јазици/компајлери:
 - Виш код, малку машински зависности
 - Преку потпрограми
 - Апстрахирање на интерфејсот преку однесувањето
 - Преку модули
 - Експорт на интерфејс, криење на имплементацијата
 - Преку апстрактни податочни типови
 - Податоците и операциите - „на купче“

Миле Јованов - Компајлери

8

Историја на идеи : Типови

- На почеток, неколку типови
 - FORTRAN: скалари, низи
 - LISP: без дефинирање на статички типови
- Реализација: Типовите
 - Овозможуваат програмерот да ја изрази апстракцијата
 - Овозможуваат компајлерите да ги увидат многуте чести грешки
 - Некогаш до точка во која програмите се гарантирано „безбедни“

Миле Јованов - Компајлери

9

Историја на идеи : Reuse (Преупотреба)

- Reuse = искористување на заедничките облици во софтверските системи
- Цел: продукција на софтверски компоненти за масовна употреба
- Преупотребата е напорна
- Два популарни пристапи (комб. во C++)
 - Параметризација на типови (List(int), List(double))
 - Класи и наследување: C++ изведени класи
- Наследувањето овозможува
 - Специјализација на постоечката апстракција
 - Проширување, модификација, криење на однесувањето

Миле Јованов - Компајлери

10

Зошто толку многу јазици?

- Апликациите имаат различни (конфликтни) потреби
- На пример:
 - Научни пресметки: брзина
 - Бизнис: генерирање на извештаи
 - ВИ: симболички пресметки
 - Системско програмирање: Ниж(низок) пристап
 - Јазици со специјална намена

Миле Јованов - Компајлери

11

Тема: Дизајн на јазик

- Нема универзално мерило
- “Добар јазик е оној кој го користат луѓето” ?
- НЕ !
 - Дали BASIC е најдобриот јазик?
- Тешко се постигнува добар јазичен дизајн

Миле Јованов - Компајлери

12

Критериуми за оцена на јазик

Карактеристики	Критериуми		
	Читливост	Пишување	Стабилност
Едноставност	*	*	*
Податочни типови	*	*	*
Синтаксен дизајн	*	*	*
Апстракција		*	*
Изразливост		*	*
Контрола на типови			*
Exception handling			*

Миле Јованов - Компјалери

13

Критериуми за добар јазичен дизајн

- **Пишување** („Пишуивливост“): Квалитет на ПЈ што овозможува на програмерот да го изрази пресметувањето јасно, точно, кратко и брзо.
- **Читливост**: Квалитет на ПЈ што овозможува на програмерот да ја разбере и сфати природата на пресметувањето лесно и точно
- **Ортогоналност**: Квалитет на ПЈ кој дадените програмски можности ги прави со малку рестрикции и комбинирливи на секој можен начин
- **Стабилност**: Квалитет на ПЈ кој обезбедува извршување на програмата ТОЧНО КАКО Е ПРЕДВИДЕНО и неможност за разорно однесување
- **Одржливост**: Квалитет на ПЈ кој овозможува пронаоѓање и поправка на грешките и додавање на нови можности

Миле Јованов - Компјалери

14

Критериуми за добар јазичен дизајн 2

- **Униформност**: Квалитет на ПЈ кој овозможува сличните можности да изгледаат и да се однесуваат слично
- **Проширливост**: Квалитет на ПЈ кој овозможува основни механизми на корисникот да додаде нови конструкции на јазикот
- **Стандардност**: Квалитет на ПЈ кој овозможува програмите да се пренесуваат од еден на друг компјутер без значајни промени во структурата на јазикот
- **Имплементабилност**: Квалитет на ПЈ кој обезбедува дека за него постои преведувач. Ова се однесува на соженоста на дефинирањето на јазикот.

Миле Јованов - Компјалери

15

Што од изучување на ПЈ и компјалери?

- Зголемена моќ на изразување
- Подобро разбирање на однес. на програмата
- Зголемена способност за учење на нови ПЈ
- Да се научи да се изгради голем стабилен систем
- Да се согледа како функционираат многу информатички концепти

Миле Јованов - Компјалери

16

Трендови

- Дизајн на јазици
 - Многу специјално наменети јазици
 - Задржување на популарните јазици
- Компјалери
 - Попотребни и посложени
 - Водени од се поголемата разлика помеѓу:
 - Новите јазици
 - Новите архитектури

Миле Јованов - Компјалери

17

Како се имплементирани јазиците?

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

Миле Јованов - Компјалери

18

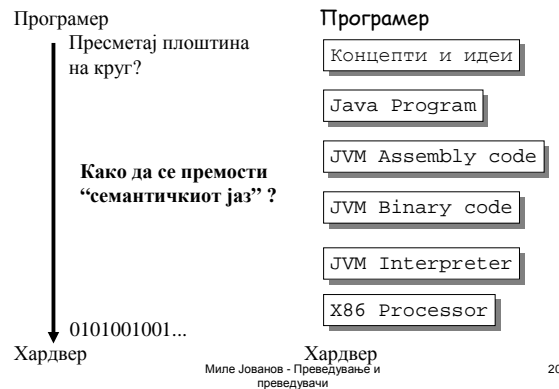
Имплементација на јазици

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

Миле Јованов - Компајлери

19

Зошто имаме потреба од јазични процесори?



Миле Јованов - Преведување и преведувачи

20

За следниот час:

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

Миле Јованов - Компајлери

21