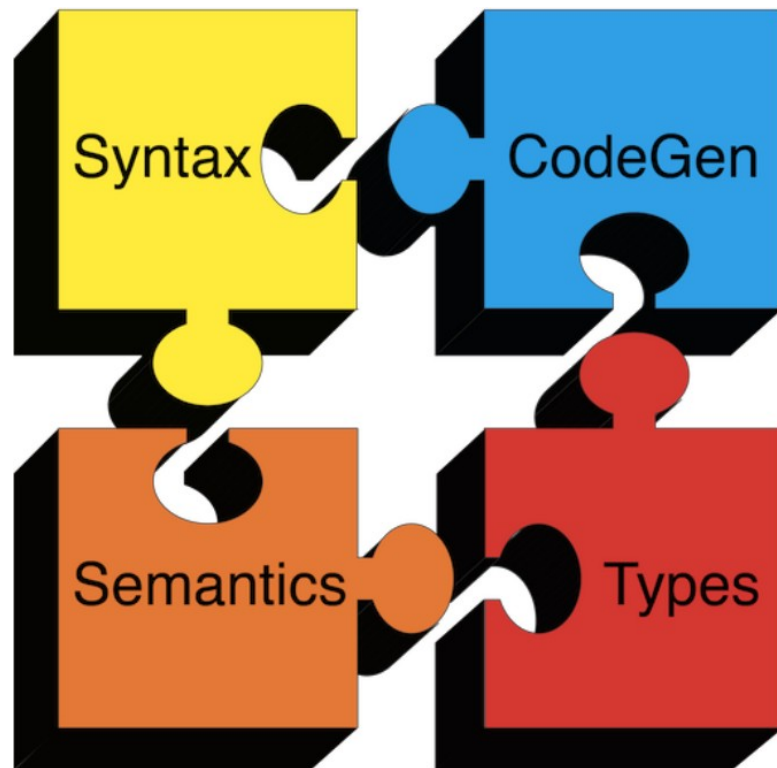


Kompajleri

preuzeto sa:

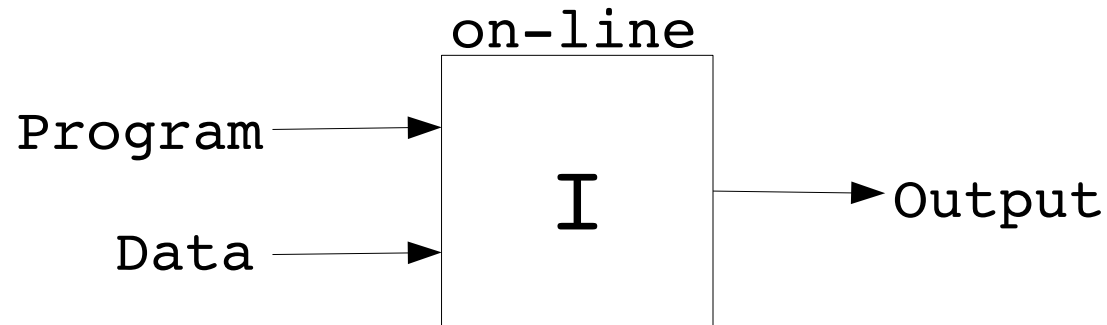
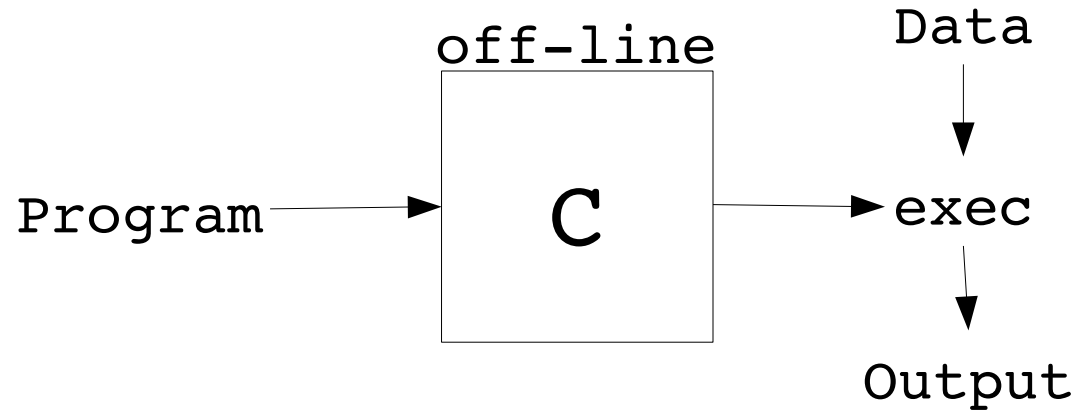
<https://www.coursera.org/course/compilers>

by Alex Aiken



pripremila:
Zorica Suvajdžin Rakić

Kompajler nasuprot interpreteru



Kompajleri - istorija

- ❑ Priča počinje 1954.
- ❑ računarnom IBM 704

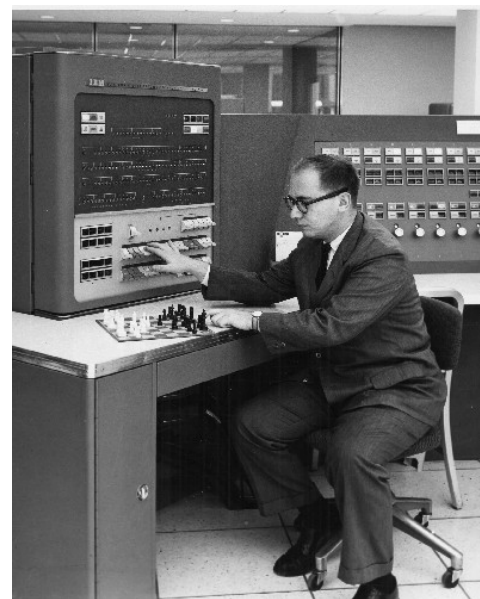


Kompajleri - istorija

- MUSIC – prvi kompjuterski muzički program by Max Mathews
- 1962., *Bell Labs*
 - fizičar *John Larry Kelly, Jr*
 - sintetizovao govor, pesma *Daisy Bell*
 - Arthur C. Clarke, *2001: A Space Odyssey*
 - HAL 9000 computer to iskoristio



- *Ed Thorp*
 - teorija igara



Kompajleri - istorija

- IBM 704
 - njihov prvi komercijalno uspešan računar
 - troškovi softvera >> troškovi hardvera
 - kako programiranje napraviti produktivnijim

Kompajleri - istorija

- Prethodni pokušaj povećanja produktivnosti se zvao „*speedcoding*“
 - *John Backus* razvio 1953.
 - daleki rođak interpretera
 - prednosti:
 - programi su se pravili mnogo lakše i mnogo brže
 - mane:
 - programi 10-20x sporiji od ručno pisanih programa
 - *speedcode* interpreter je zauzimao 300 bajta memorije
 - = 30% ukupne memorije računara IBM 704

Kompajleri - istorija

- *speedcoding* nije postao popularan
 - ali je *John Backus* pokrenuo novi projekat
 - u to vreme su bila bitna računanja u nauci (formule)
- FORTRAN I
 - ideja: da se formule ne interpretiraju već prevode
 - bile bi brže
 - **FOR**mulas **TRAN**slated
 - projekat trajao: 1954 – 1957
 - pp: 1 godina
 - projekat je bio uspešan:
 - do 1958. – 50% programa na FORTRAN-u
 - prvi uspešan programski jezik visokog nivoa
 - prvi kompajler

Kompajleri - istorija

- ❑ FORTRAN I - veliki uticaj na računarske nauke
 - doveo do ogromnog teoretskog rada (teorija + praksa)
 - moderni kompajleri zadržavaju osnovne osobine FORTRAN I
- ❑ Proporcije kompleksnosti faza su se izmenile od FORTRAN-a

L	P	S	GK	O
---	---	---	----	---

L	P	S	GK	O
---	---	---	----	---

Ekonomija programskih jezika

- videli smo kako se jezici implementiraju
 - bilo bi lepo da smo i dizajnirali neki mali jezik
 - kako jezici rade u stvarnom svetu
 - zašto se neki jezici koriste a drugi ne
1. Zašto postoji toliko mnogo programskih jezika?
 - lista programskih jezika na wikipedia sajtu
http://en.wikipedia.org/wiki/List_of_programming_languages
 2. Zašto se pojavljuju novi programski jezici?
 3. Šta je dobar programski jezik?

Zašto postoji toliko mnogo programskih jezika?

- domeni primene imaju različite / konfliktne potrebe
 - teško je dizajnirati jedan jezik koji bi
 - radio sve
 - u svim situacijama
 - za sve programere
- računarstvo u nauci: *scientific computing* (npr: simulacije)
 - dobra FP podrška
 - dobre operacije nad nizovima **FORTRAN**
 - paralelizam
- poslovne aplikacije: *business applications*
 - pouzdanost
 - generisanje izveštaja **SQL, Java**
 - analiza podataka
- sistemsko programiranje: *system programming*
 - kontrola nad resursima **C / C++**
 - *real time* ograničenja

Zašto se pojavljuju novi programski jezici?

- Ako već postoji toliko pj, zašto se pojavljuju novi?
- Tvrdnja: Alex Aiken
 - obuka programera je dominantni deo cene/koštanja u životu programskog jezika
 - ne samo novac (knjige, kursevi), već i
 - slobodno vreme programera u kom on sam uči
 - x100.000 ili x1.000.000 programera
 - dizajn jezika: tipično 1-2 čoveka
 - pravljenje kompajlera: 10-20 ljudi za jako veliki kompajler

Zašto se pojavljuju novi programski jezici?

1. Široko korišćeni programski jezici se sporo menjaju
 - i za malu izmenu treba puno vremena i skupo je
2. Jednostavno je započeti novi jezik
 - troškovi su minimalni, mali broj korisnika
 - brzo se menjaju, adaptiraju potrebama
- Kako programer bira između ova dva?
 - produktivnost > troškova obuke
3. Novi jezici se usvajaju da bi popunili prazninu
 - informaciona revolucija još uvek traje
 - nastaju novi domeni, nova vrsta programiranja - na par godina
 - mobilne aplikacije, nove mobilne tehnologije
- novi jezici imaju tendenciju da liče na stare jezike
 - to smanjuje troškove obuke programera
 - brže i lakše ga nauče
 - Java vs. C++

Šta je dobar programski jezik?

- Ne postoji **nijedna** univerzalno prihvaćena metrika za dizajn jezika
 - ljudi se ne slažu oko toga šta čini dobar jezik
 - kako naći dobru metriku, npr:
 - dobar jezik je onaj koji ljudi koriste?
 - mera je popularnost jezika: jasna mera
 - ako više ljudi koristi neki jezik, u nekom smislu je to bolji jezik
 - => VB je najbolji jezik?
 - da li je jezik u širokoj upotrebi
 - tehničke osobine možda nisu najbitnije
 - npr: namenjen domenu aplikacija za koji nema boljeg jezika

Ekonomija programskih jezika

- Domeni primene imaju konfliktne potrebe
 - teško je dizajnirati jedan sistem za sve
- Obuka programera je dominantni deo cene programskog jezika

- Ove dve stavke objašnjavaju zašto imamo nove jezike:
 - stare jezike je teško menjati
 - kada postoji prilika (novi domen) pravi se novi jezik
 - pre nego da se stari jezik i svi njegovi korisnici prilagođavaju novom domenu