اصول طراحی کامپایلر

درس ۱

مقدمه‌ای بر کامپایلرها

An Introduction to Compilers

کاظم فولادی قلعه

دانشکده مهندسی، پردیس فارابی

دانشگاه تهران

http://courses.fouladi.ir/compiler

کامپایلر

اتیمولوژی: کامپایل

compile: (early 14c.), Middle English,
from O.Fr. (Anglo-French) *compiler*,
from L. ***compilare*** "to snatch together, plunder, heap,"
from ***com***- "together" + ***pilare*** "to compress, ram down."

from the Online Etymology Dictionary

ربودن، چپاول و غارت کردن، گردآوردن
گردآوردن + خلاصه‌کردن ≈ تألیف، گردآوری

ترمینولوژی: کامپایل

compile

FUNCTION: transitive verb

INFLECTED FORM(S): com.piled; com.pil.ing

1 : to compose out of materials from other documents

2 : to collect and edit into a volume

3 : to build up gradually <*compiled* a record of four wins and two losses>

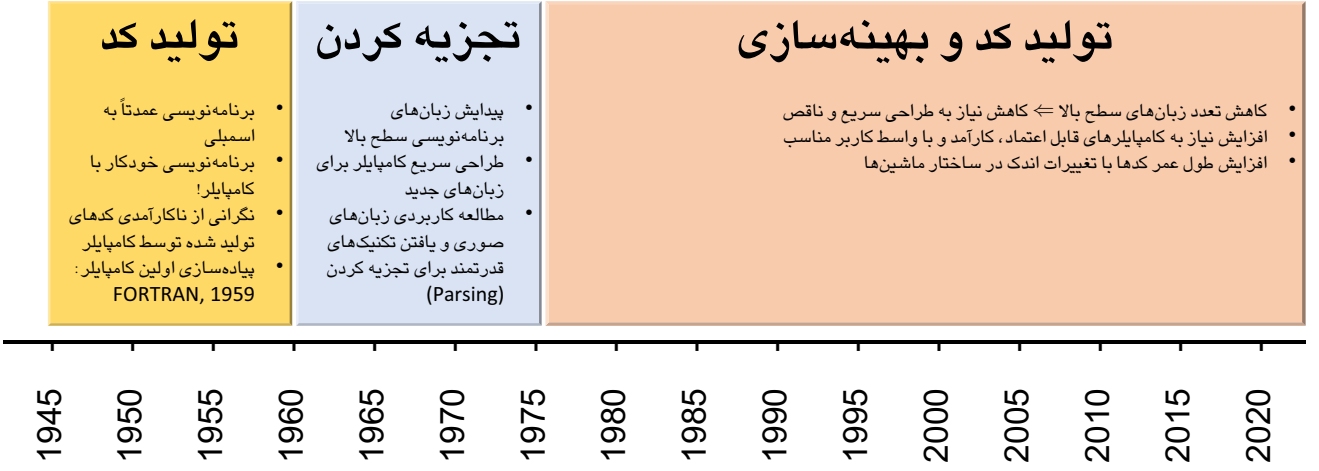
4 : to run (as a program) through a compiler

from the Merriam-Webster Online Dictionary

چرا طراحی و ساخت کامپایلرها را مطالعه می‌کنیم؟

- **کامپایلر** شاخه‌ی موفقی از علم کامپیوتر است.
- **کامپایلر** کاربردهای فراوانی دارد.
- **کامپایلر** الگوریتم‌های مفیدی دارد.
- **کامپایلر** و ساخت آن مهارت‌های برنامه‌نویسی را بالا می‌برد

تاریخچه کامپایلر



تغییر مسئله در مورد کامپایلر

چگونه کد باید کامپایل شود؟

↓

چه‌چیزی از کد باید کامپایل شود؟

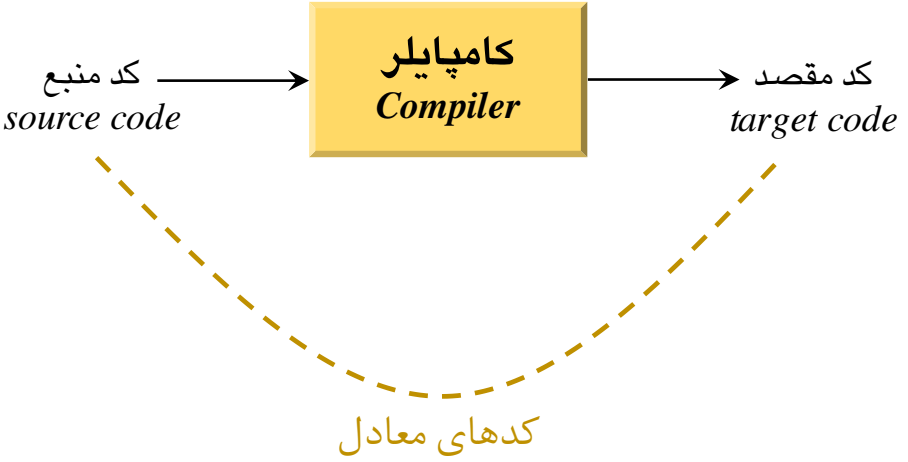
آیا کامپایلر یک مسئله‌ی حل شده است؟

تغییر در معماری کامپیوتر ⇐ **تغییر در کامپایلرها**

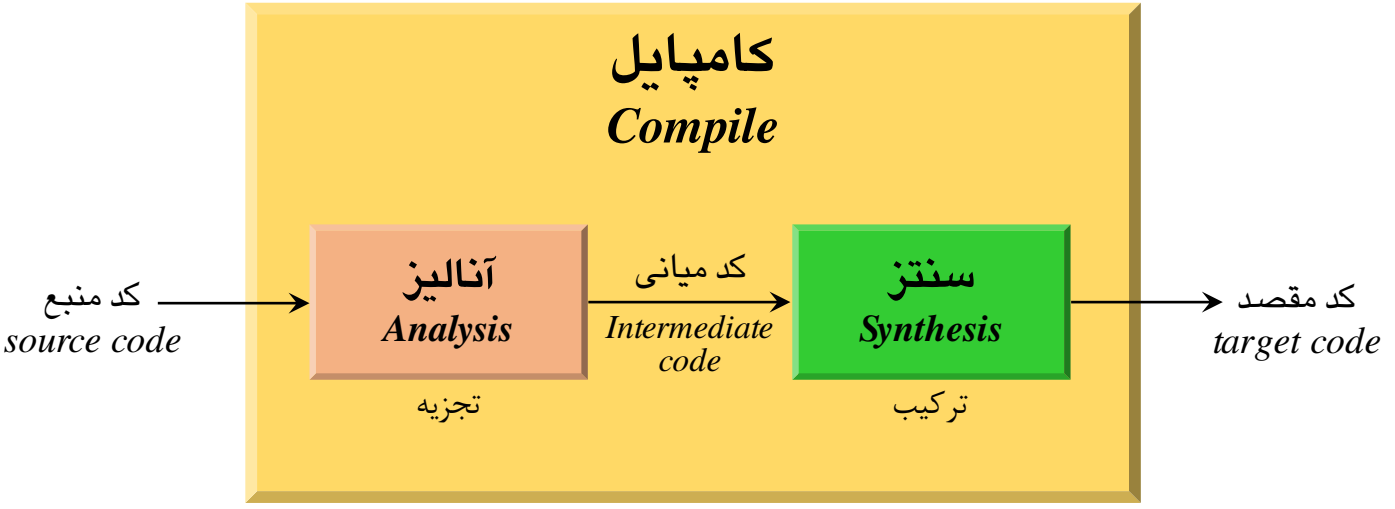
ویژگی‌های جدید معماری ⇐ مسائل جدید کامپایلر
تغییر هزینه‌ها در معماری ⇐ تغییر مسائل پراهمیت در کامپایلر
⇐ ضرورت مهندسی مجدد راه‌حل‌های معروف

کامپایلر

کامپایلر


کد منبع
source code
کامپایلر
Compiler
کد مقصد
target code
کدهای معادل

آنالیز و سنتز در فرآیند کامپایل



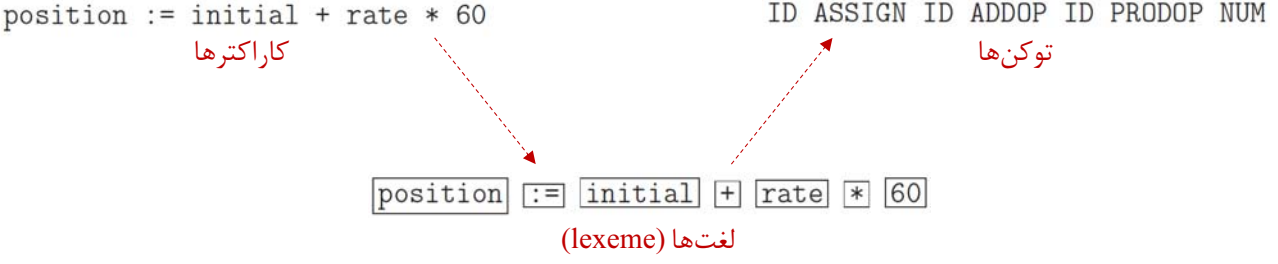
مراحل فرآیند کامپایل در یک نگاه

تحلیل لغوی


دنباله‌ی کاراکترها
sequence of charactrs
تحلیل‌گر لغوی
Lexical Analyzer
دنباله‌ی توکن‌ها
sequence of tokens

تحلیل لغوی

مثال



تحلیل نحوی


دنباله‌ی توکن‌ها
sequence of tokens
تحلیل‌گر نحوی
Syntax Analyzer
درخت ساختارها
tree of structures

تحلیل نحوی

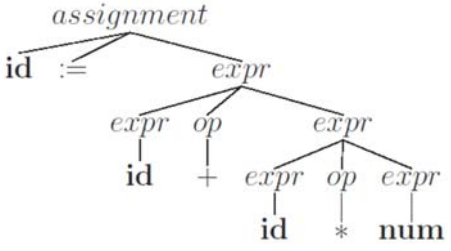
مثال

دنباله‌ی توکن‌ها
sequence of tokens

تحلیل‌گر نحوی
Syntax Analyzer

درخت ساختارها
tree of structures

$\mathrm{id} := \mathrm{id} + \mathrm{id} * \mathrm{num}$



گرامر

$$
\begin{aligned}
assignment &\rightarrow \mathrm{id} := expr \\
expr &\rightarrow \mathrm{id} \mid \mathrm{num} \mid expr\ op\ expr \mid (expr) \\
op &\rightarrow + \mid - \mid * \mid /
\end{aligned}
$$

تحلیل معنایی

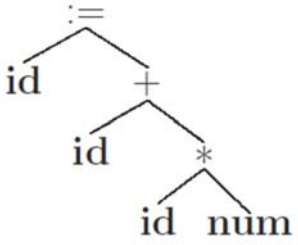
تحلیل معنایی

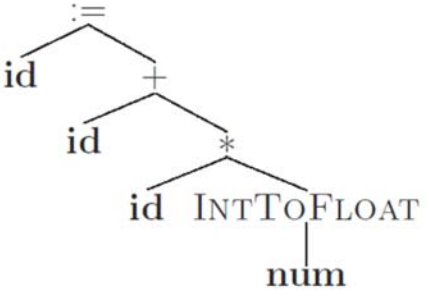
مثال

درخت ساختارها
tree of structures

تحلیل‌گر معنایی
Semantic Analyzer

حاشیه‌نویسی‌شده
درخت ساختارها
annotated tree of structures





تولید کد میانی

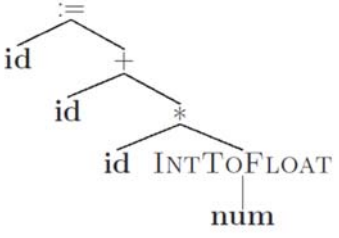
تولید کد میانی

مثال

حاشیه‌نویسی‌شده
درخت ساختارها
annotated tree of structures

مولد کد میانی
Intermediate Code Generator

دنباله‌ی اجزای کد میانی
sequence of interm. code parts



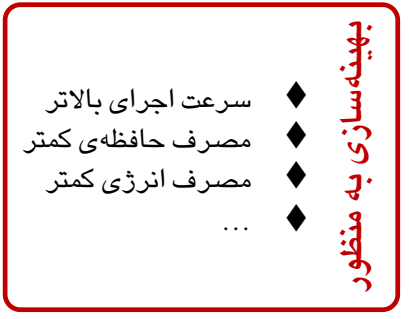

```
temp1 := inttofloat(60)
temp2 := id3 * temp1
temp3 := id2 + temp2
id1 := temp3
```

کد سه‌آدرسی

بهینه‌سازی کد میانی

بهینه‌سازی کد میانی

اهداف



بهینه‌سازی کد میانی

مثال


```
temp1 := inttofloat(60)
temp2 := id3 * temp1
temp3 := id2 + temp2
id1 := temp3
```

```
temp1 := id3 * 60.0
id1 := id2 + temp1
```

تولید کد


دنباله‌ی اجزای کد میانی
sequence of
interm. code parts
مولد کد
Code Generator
دنباله‌ی اجزای کد نهایی
optimal sequence
of final code parts

تولید کد

وظایف مولد کد

- انتساب حافظه به متغیرها
- انتساب متغیرها به ثبات‌ها
- ترجمه به دستورالعمل‌های کد اسمبلی
- استفاده از مشخصات معماری سخت‌افزار
- ...

تولید کد

مثال

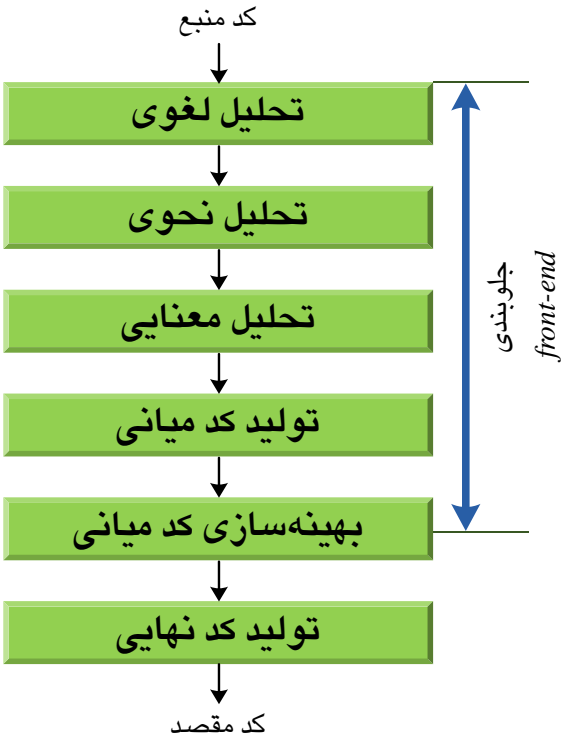

```
temp1 := id3 * 60.0
id1 := id2 + temp1
```

```
MOVF id3, R2
MULF #60.0, R2
MOVF id2, R1
ADDF R2, R1
MOVF R1, id1
```

مراحل فرآیند کامپایل در یک نگاه

مراحل فرآیند کامپایل در یک نگاه

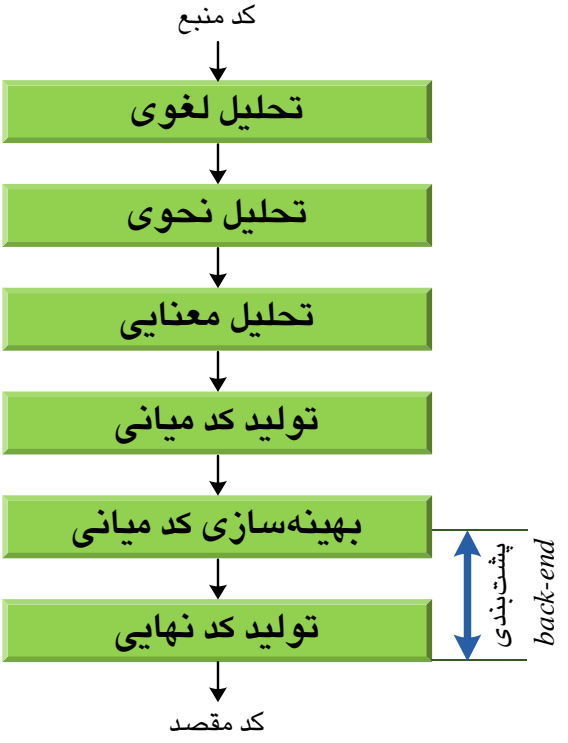
جلوبندی



وابسته به زبان منبع
نابسته به ماشین مقصد

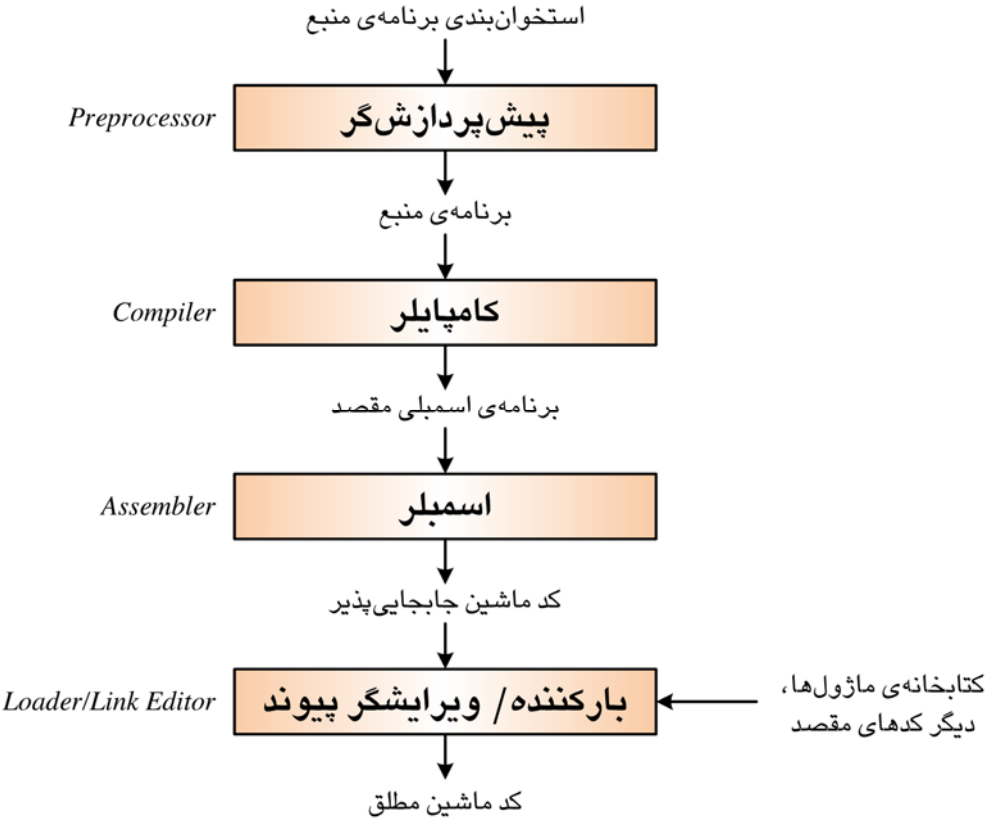
مراحل فرآیند کامپایل در یک نگاه

پشت‌بندی



نابسته به زبان منبع
وابسته به ماشین مقصد

تولید کد قابل اجرا از پیش‌پردازش تا پیونددهی



جدول نمادها

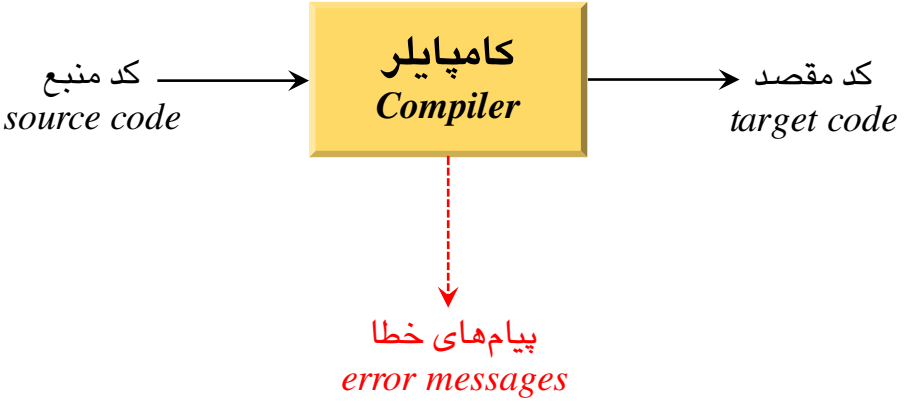
یک ساختمان داده برای نگهداری اطلاعات در مورد نمادهای برنامه

مانند: نوع توکن، حوزه‌ی دید، آدرس حافظه و ...

SYMBOL TABLE

1	position	...
2	initial	...
3	rate	...
4		

اداره کردن خطاها توسط کامپایلر



انواع خطا در فرآیند کامپایل

محل رخداد خطا	نام خطا	مثال
تحلیل لغوی	**خطای لغوی (lexical error)**	شروع کردن یک شناسه با عدد
تحلیل نحوی	**خطای نحوی (syntax error)**	عدم تطابق پرانترها در یک عبارت ریاضی
تحلیل معنایی	**خطای معنایی (semantic error)**	انتساب یک مقدار اعشاری به متغیر صحیح
تولید کد میانی		کمبود حافظه بر روی دیسک
بهینه‌سازی کد میانی		کمبود حافظه بر روی دیسک
تولید کد نهایی		کمبود حافظه بر روی دیسک

گذر در کامپایلر

PASS

تعداد گذر: تعداد دفعات خواندن کد منبع توسط کامپایلر

* مطلوب: کمتر بودن تعداد گذرهای کامپایل
** عدم امکان انجام برخی مراحل کامپایل در یک گذر (مثل بهینه‌سازی و بخشی از تحلیل معنایی)

انواع کامپایلرها	
تک‌گذره *single-pass*	چندگذره *multi-pass*

حمل‌پذیری و تغییر پذیر مقصد

حمل‌پذیری یک کامپایلر:
عدم نیاز به تغییرات متعدد برای اجرا در ماشین‌های مختلف

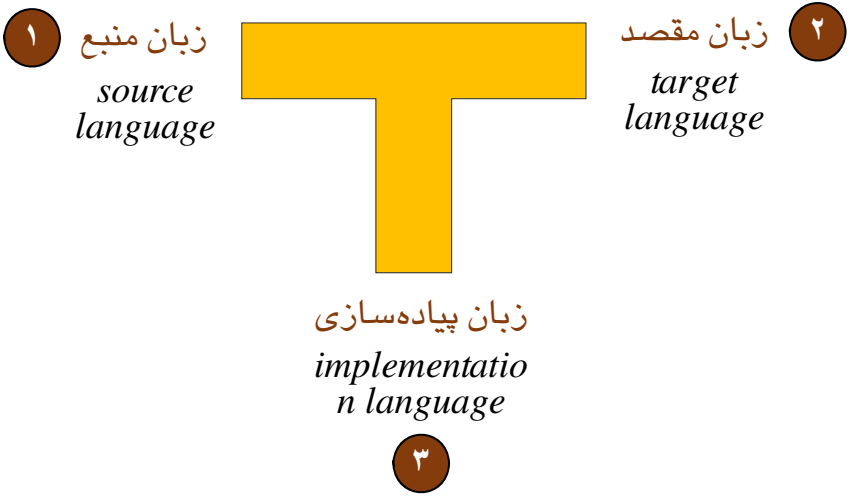
حمل‌پذیری *portability*	تغییرپذیری مقصد *retargetability*
قابلیت اجرای کامپایلر در ماشین‌های مختلف	تولید کد برای ماشین‌های دیگر (تغییر مقصد)

کامپایلر تقاطعی

CROSS COMPILER

کامپایلر تقاطعی:
کامپایلری که بر روی یک ماشین اجرا می‌شود و کد مقصد را برای ماشین دیگری تولید می‌کند.

* کاربرد: پیاده‌سازی کامپایلر



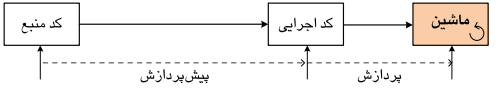
انواع مترجم برای زبان‌ها

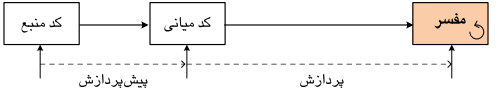
انواع مترجم‌ها	
کامپایلر ***compiler***	**مفسر** ***interpreter***
ترجمه برنامه به کد زبان ماشین قابل اجرا انجام پیش‌پردازش‌های سنگین سرعت اجرای بالاتر	اجرای برنامه «همان‌گونه که هست» بدون ترجمه‌ی اولیه سرعت اجرای پایین‌تر

تمایز فرآیند کامپایل و تفسیر

انواع مترجم‌ها

کامپایلر *compiler*	مفسر *interpreter*
ترجمه برنامه به کد زبان ماشین قابل اجرا انجام پیش‌پردازش‌های سنگین سرعت اجرای بالاتر	اجرای برنامه «همان‌گونه که هست» بدون ترجمه‌ی اولیه سرعت اجرای پایین‌تر





مزیت مفسر نسبت به کامپایلر

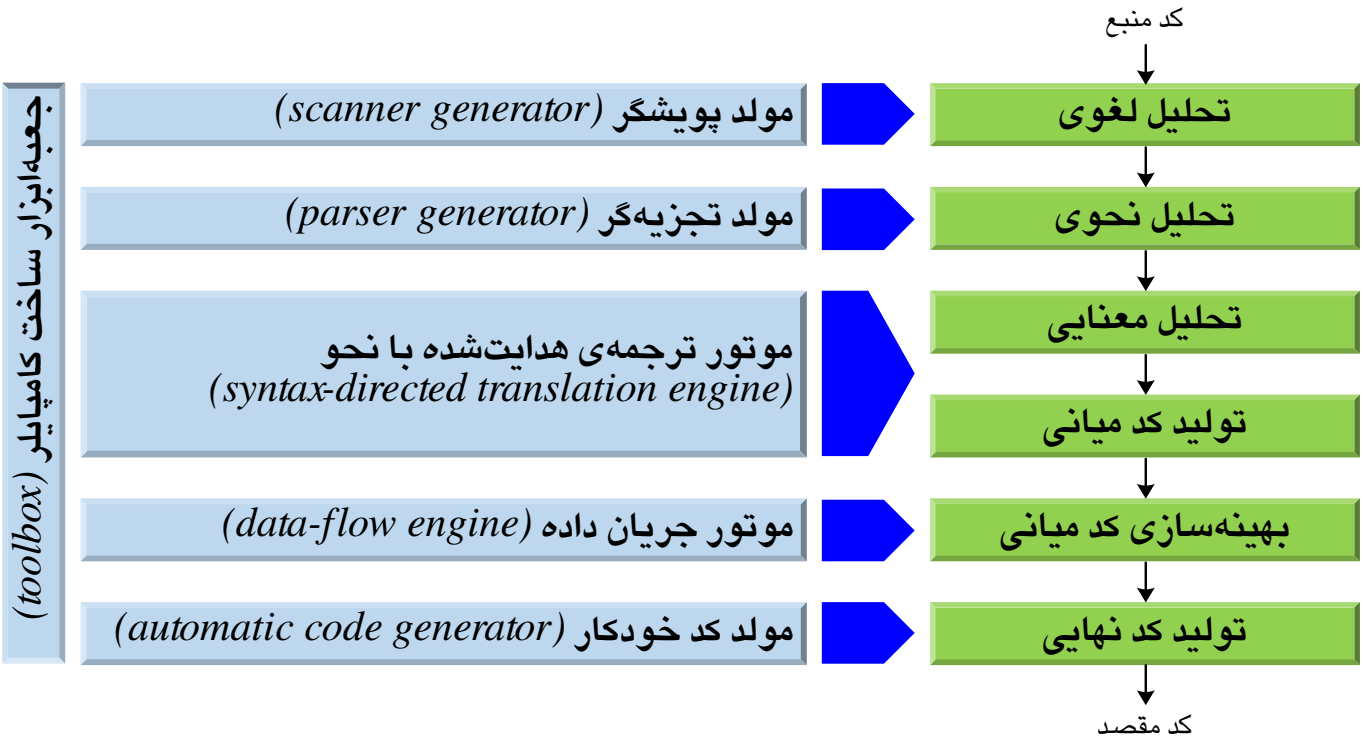
انواع مترجم‌ها	
کامپایلر *compiler*	**مفسر** *interpreter*

قابلیت اجرا بر روی بیشتر ماشین‌ها
نوشتن مفسر ساده‌تر از کامپایلر است
عیب‌یابی و گزارش خطای ساده‌تر: با اجرای مستقیم برنامه‌ها
افزایش امنیت

خانواده‌ی مترجم‌ها

انواع مترجم‌ها				
مبدل‌های فایل *file converters*	کامپایلرهای سیلیکون *silicon compilers*	قالب‌بندی‌کننده متن *text formatters*	مفسرهای پرس‌وجو *query interpreters*	کامپایلرهای زبان‌های بر.سا *PL compilers*
				

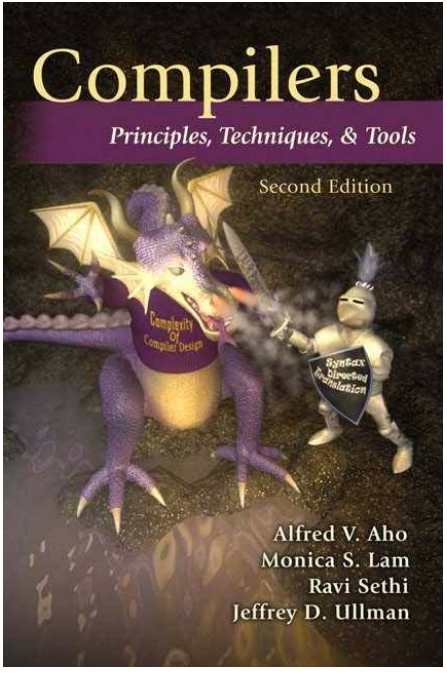
ابزارهای ساخت کامپایلرها



translator-writing /compiler-generator /compiler-compiler

منابع

منبع اصلی



A. V. Aho, M. S. Lam, R. Sethi, J. D. Ullman,
Compilers: Principles, Techniques and Tools,
Second Edition, Addison-Wesley, 2007.

Chapter 1