

编译原理概述

魏恒峰

hfwei@nju.edu.cn

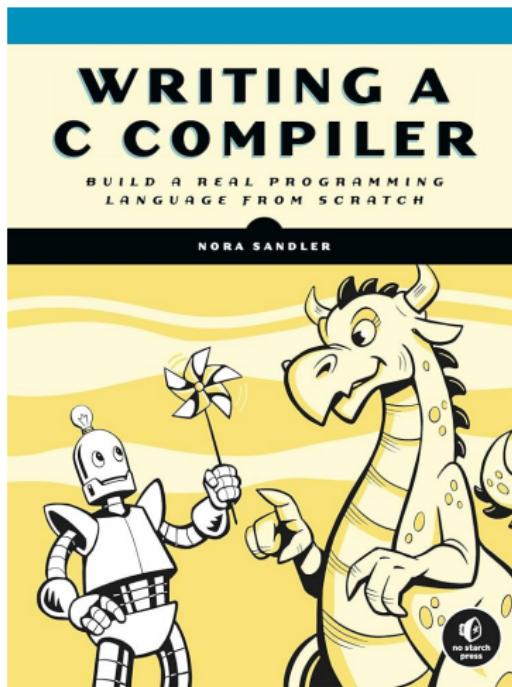
2024 年 03 月 01 日 (周五)



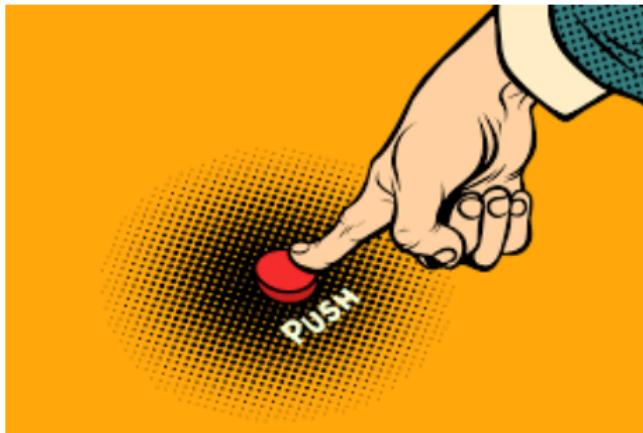
哦？编译原理？



只学原理不行，更重要的是…

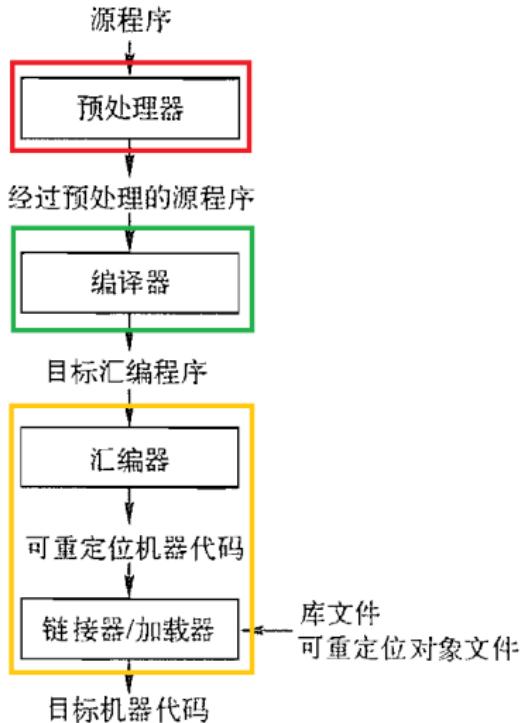


编译器：最熟悉的陌生人



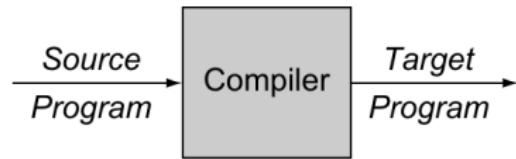
```
gcc factorial.c -o factorial
```

“蚂老师，发生甚么事了？”



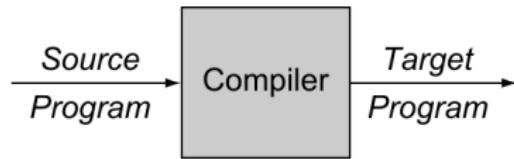
“高级”语言 \Rightarrow (通常) “低级”语言 (如, 汇编语言)

汇编语言经过**汇编器**生成机器语言



“高级”语言 \Rightarrow (通常) “低级”语言 (如, 汇编语言)

汇编语言经过汇编器生成机器语言



GopherJS - A compiler from Go to JavaScript

[godoc](#) [reference](#)

GopherJS compiles Go code ([golang.org](#)) to pure JavaScript code. Its main purpose is to give you the opportunity to write front-end code in Go which will still run in all browsers.

“还要学习汇编语言啊？不都已经还给你们了吗？”



(Reduced Instruction Set
Computer)

“还要学习汇编语言啊？不都已经还给你们了吗？”

隆重推荐 <https://godbolt.org/>



(Reduced Instruction Set
Computer)



<https://godbolt.org/z/6f3o51oeh>
(factorial.c)

Join Us at RISC-V Summit Europe | June 24-28

X in 📺 📧 📩 🔍

Languages Tech Meetings Community Meetings Working Groups Portal Join

RISC-V®

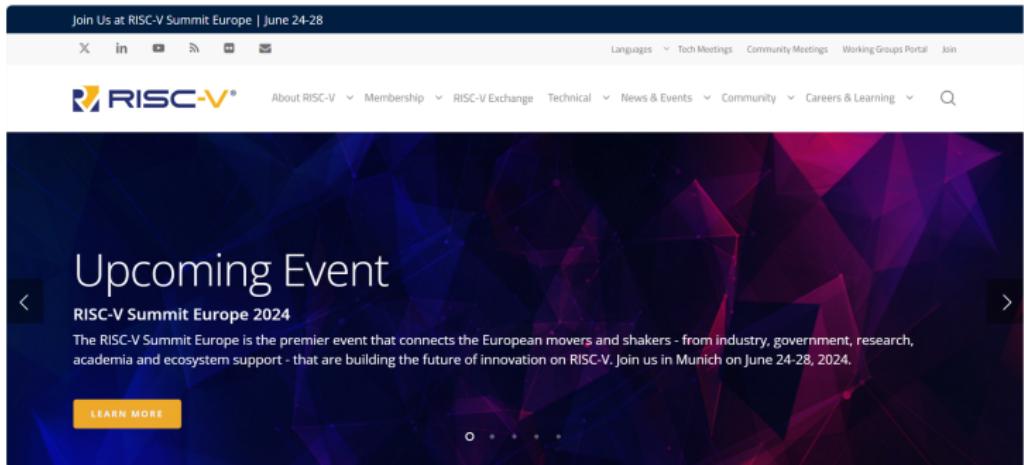
About RISC-V Membership RISC-V Exchange Technical News & Events Community Careers & Learning

Upcoming Event

RISC-V Summit Europe 2024

The RISC-V Summit Europe is the premier event that connects the European movers and shakers - from industry, government, research, academia and ecosystem support - that are building the future of innovation on RISC-V. Join us in Munich on June 24-28, 2024.

LEARN MORE



Join RISC-V International



RISC-V International is a forum for members to collaborate building the first open architecture community of software and hardware innovators in the world.

<https://riscv.org/>

```
book-covers$ cd .. > book-covers
1 .file "book-cover.s"
2 .option nopic
3 .attribute arch, "rv32i|lp64"
4 .attribute unaligned.access, 0
5 .attribute stack_align, 16
6 .text
7 .align 2
8 compute_the_answer_to_the_ultimate_question_of_life_the_universe_and_everything:
9    .size 0x42
10   .ret
11 .align 2
12   .globl do_something_1000_times
13 .type do_something_1000_times, @function
14 do_something_1000_times:
15   addi $a0,-16
16   sw $a0,16($sp)
17   sw $a0,32($sp)
18   li $a0,1000
19 .Lhi
20   addi $a0,-1
21   call do_something
22   lw $a0,16($sp)
23   lw $a0,32($sp)
24   addi $sp,16
25   jr $ra
26 .section .rodata.str1.4,"a@P",@progbits,1
27 .align 2
28 .LC0:
29 .ascii "There are 10 types of people in this world "
30 .ascii "those who understand binary and those who don't"
31 .align 2
32 .LC1:
33 .string "Assembly language you must learn!"
34 .align 2
35 .LC2:
36 .ascii "The Unicamp CS course was created in 2009 - "
37 .ascii "The first one in Brazil!"
38 .LC3:
39 .byte 78, 385, 90, 181, 33, 32, 99, 151, 157, 33, 187
40 .byte 159, 113, 119, 32, 65, 85, 97, 73, 73, 33, 0
```

<https://riscv-programming.org/book.html>

```

book-cover.c
1 书名: book-cover
2   .file "book-cover.c"
3   .option nopic
4   .attribute arch, "rv32iipf"
5   .attribute unaligned.access, 0
6   .attribute stack_align, 16
7   .text
8   .align 2
9   compute_the_answer_to_the_ultimate_question_of_life_the_universe_and_everything:
10    .text
11    .align 2
12    .global do_something_1000_times
13    .type do_something_1000_times, @function
14    do_something_1000_times:
15    .addi $a0,$a0,-16
16    sw $a0,0($sp)
17    sr $a0,12($sp)
18    li $a0,1000
19    .lbi
20    addi $a0,$a0,-1
21    call do_something
22    .align 2
23    lw $a0,12($sp)
24    sw $a0,0($sp)
25    addi $sp,$sp,16
26    jr $ra
27    .section .rodata.str1.4,"a#0",@progbits,1
28    .align 2
29    .LC0:
30    .ascii "There are 10 types of people in this world "
31    .ascii "Those who understand binary and those who don't"
32    .align 2
33    .LC1:
34    .string "Assembly language you must learn!"
35    .align 2
36    .LC2:
37    .ascii "The Unicamp CS course was created in 2009 - "
38    .ascii "The first one in Brazil!"
39    .LC3:
40    .byte 78, 385, 99, 181, 33, 32, 99, 151, 157, 33, 187
41    .byte 159, 113, 119, 32, 65, 89, 97, 73, 73, 33, 0

```

**An Introduction to
Assembly Programming
with RISC-V**



<https://riscv-programming.org/book.html>

<http://www.riscvbook.com/>

你的笔记本很可能不是 RISC-V 架构, 需要使用模拟器

We are having a problem billing the courses-at-nju-by-hfwei organization. Please [update your payment method](#) or call your payment provider for details on why the transaction failed. If the features included in GitHub Free meet your needs, you can downgrade by going to your [Billing settings](#). You can always [contact support](#) with any questions.

rars Public

Code Issues Pull requests Wiki Security Insights

Type to search

Watch 25 Fork 201 Stared 1.1k

master 3 Branches 12 Tags

Go to file Add file Code

privat and TheThirdOne MessagesPane: Force black text color on yellow selection b0c5cd1 · 6 months ago 365 Commits

examples fix some typos 6 months ago

src MessagesPane: Force black text color on yellow selection 6 months ago

test Update pseudo-op test 8 months ago

.gitignore Ignore build files 3 years ago

.gitmodules Start using JSoftFloat to solve floating point issues 5 years ago

.travis.yml Update travis to only build on master 4 years ago

License.txt Refactor the mars package to rars 7 years ago

README.md fix some typos 6 months ago

build-jar.sh Fixed build-jar.sh to work with MacOS 3 years ago

About

RARS -- RISC-V Assembler and Runtime Simulator

education simulator ide assembler
riscv mars risc-v

Readme View license Activity 1.1k stars 25 watching 201 forks Report repository

Releases 12 Lots of small improvements (Latest)

<https://github.com/TheThirdOne/rars>

Q : 机器语言是如何跑起来的?

Q : 机器语言是如何跑起来的?

作业 (P1 ~ P9): <https://www.bilibili.com/video/BV1EW411u7th>
(计算机科学速成课 40 集全 Crash Course Computer Science)

Q : 机器语言是如何跑起来的?

作业 (P1 ~ P9): <https://www.bilibili.com/video/BV1EW411u7th>
(计算机科学速成课 40 集全 Crash Course Computer Science)



Q : 机器语言是如何跑起来的?

作业 (P1 ~ P9): <https://www.bilibili.com/video/BV1EW411u7th>
(计算机科学速成课 40 集全 Crash Course Computer Science)



Q : 机器语言是如何跑起来的?

作业 (P1 ~ P9): <https://www.bilibili.com/video/BV1EW411u7th>
(计算机科学速成课 40 集全 Crash Course Computer Science)



“我只想安静地做个码农，你为什么要来打扰我？”



.fun



install tutorial cheat sheet docs community

Alda is a text-based programming language for music composition. It allows you to write and play back music using only a text editor and the command line.

```
piano:  
o3  
g8 a b > c d e f+ g | a b > c d e f+ g4  
g8 f+ e d c < b a g | f+ e d c < b a g4  
<< g1/>g/>g/b/>d/g
```

The language's design equally favors aesthetics, flexibility and ease of use.

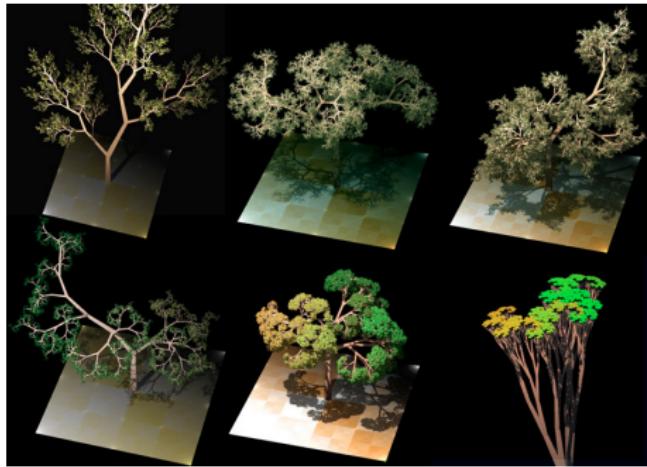
alda repl

alda play -f xxx.alda

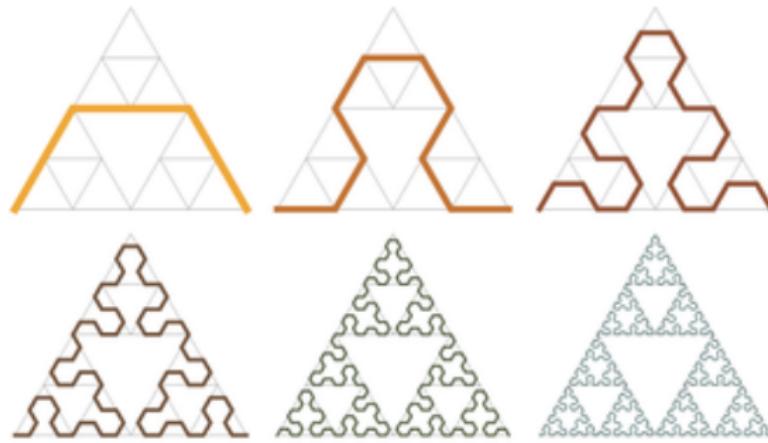
alda @ 知乎



10 Demo: Alda @ youtube



Fractal Grower (Play with It!)



Sierpinski Triangle @ wiki



语言类应用程序

- ▶ 配置文件解析 (.properties)
- ▶ CSV 文件 (Comma-Separated Values)
- ▶ JSON 文件 (JavaScript Object Notation)

语言类应用程序

- ▶ 配置文件解析 (.properties)
- ▶ CSV 文件 (Comma-Separated Values)
- ▶ JSON 文件 (JavaScript Object Notation)
- ▶ SQL 引擎 (Structured Query Language)
- ▶ TLA⁺/TLAPS (TPaxos.tla)
- ▶ (Java) 字节码解释器
- ▶ C/C++ 语言编译器

语言类应用程序

- ▶ 配置文件解析 (.properties)
- ▶ CSV 文件 (Comma-Separated Values)
- ▶ JSON 文件 (JavaScript Object Notation)
- ▶ SQL 引擎 (Structured Query Language)
- ▶ TLA⁺/TLAPS (TPaxos.tla)
- ▶ (Java) 字节码解释器
- ▶ C/C++ 语言编译器
- ▶ 排版工具 (LATEX)
- ▶ 绘图工具 (TikZ, Dot/Graphviz)
- ▶ L-System (Cantor Set)

“好吧，我学，我学，可是到底难不难啊？”



学习编译原理最大的难点：“只见树木，不见森林”



絕紙知上
此得事來
要終覺
躬行淺



考勤 (0%): 非必要不点名, 不需要请假

考勤 (0%): 非必要不点名, 不需要请假

平时作业 (0%): ≈ 10 次作业, 每次 ≤ 3 题

考勤 (0%): 非必要不点名, 不需要请假

平时作业 (0%): ≈ 10 次作业, 每次 ≤ 3 题

课程实验 (60%): 8 ~ 10 次实验

考勤 (0%): 非必要不点名, 不需要请假

平时作业 (0%): ≈ 10 次作业, 每次 ≤ 3 题

课程实验 (60%): 8 ~ 10 次实验

期末测试 (40%): 考试周统一安排; 3 小时; 上机考试

考勤 (0%): 非必要不点名, 不需要请假

平时作业 (0%): ≈ 10 次作业, 每次 ≤ 3 题

课程实验 (60%): 8 ~ 10 次实验

期末测试 (40%): 考试周统一安排; 2 小时; 开卷

每周五发布作业 下周五 23:55 前自愿提交作业



邀请码: 8G928EBJ

课程实验：开发 SysY 语言编译器



L0: 环境配置 预计本周五 16:00 发布

鼓励讨论，但需独立编码完成课程实验



课程实验：抄袭者当次实验计 0 分

QQ 群号: 869910463



助教: 顾龙、李和煦、钱品亦

<http://docs.compilers.cpl.icu/>



Q 搜索

- > 课程简介
- > 课程通知
- > 课程教材
- > 课程作业
- > 课程资源
- > 抄袭与惩罚
- > 课程实验

编译原理课程网站，请**收藏**并及时关注网站更新

这学期, 你将与它日夜相伴 ··

<https://oj.compilers.cpl.icu/>

Online Judge

The screenshot shows the Online Judge system interface. On the left, there is a vertical sidebar with a light gray header "系统导航" and several menu items: "系统主页" (highlighted with a blue bar), "作业列表", "用户设置", and "系统管理" (which contains "用户管理", "作业管理", "提交管理", "备份查询", and "成绩查询"). On the right, the main content area has a "LOGO" placeholder and a title "2024 南京大学春季学期编译原理校外OJ账号申请". Below the title are four input fields with validation messages: "姓名" (Name) with "请输入您的姓名", "学号" (Student ID) with "请输入您的学号", "申请时间" (Application Time) with "请选择您申请OJ的日期时间", and "邮箱" (Email) with "请输入您的邮箱，我们会将您的账号密码等信息通过邮箱的形式转发给您". At the bottom right, there are navigation icons for back, forward, search, and refresh.

系统导航

系统主页

作业列表

用户设置

系统管理

用户管理

作业管理

提交管理

备份查询

成绩查询

LOGO

2024 南京大学春季学期编译原理校外OJ账号申请

姓名 *

请输入您的姓名

学号 *

请输入您的学号

申请时间 *

请选择您申请OJ的日期时间

邮箱 *

请输入您的邮箱，我们会将您的账号密码等信息通过邮箱的形式转发给您

Zulip 交流平台

The screenshot shows the Zulip web interface. On the left is a sidebar with navigation links: USERS, DIRECT MESSAGES, STREAMS, and more. The main area displays a stream event titled "lecture-0-overview" from "stream events". The event was created by "Hengfeng Wei" on "FEB 28 00:47". The description is "Discussion about Lecture 0-overview". A list of users is shown on the right, including Hengfeng Wei, 聂芳, 吴洪波, Heze Li, jehowza, mich, qianpingyi, Qirekai SHI, shrey0010, and The Galaxy. There is a link to "Invite more users". At the bottom, there are buttons for "Start new conversation" and "New direct message". The status bar at the bottom shows the URL "Message #lecture-0-overview > stream events" and the date "2024-03-01 11:21".



[https://2024-compilers-at-software-nju.zulipchat.com/join/
wxwq3fib56ltlff2mk6qyrz5/](https://2024-compilers-at-software-nju.zulipchat.com/join/wxwq3fib56ltlff2mk6qyrz5/)

overview.pdf

overview-handout.pdf

compilers-lectures / 2024 /

Add file

...

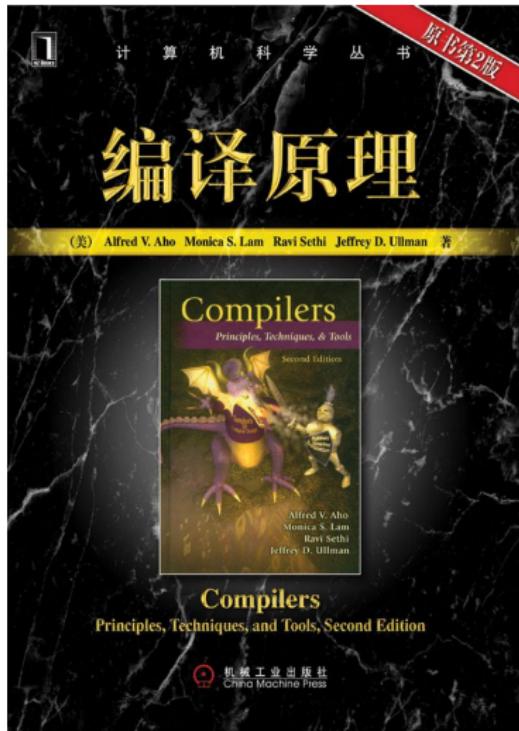
 hengxin +material/

778270f · 4 minutes ago

 History

Name	Last commit message	Last commit date
 ..		
 0-overview	+material/	4 minutes ago
 material	+material/	4 minutes ago
 README.md	+material/	4 minutes ago
 preamble.tex	+0-overview	7 minutes ago

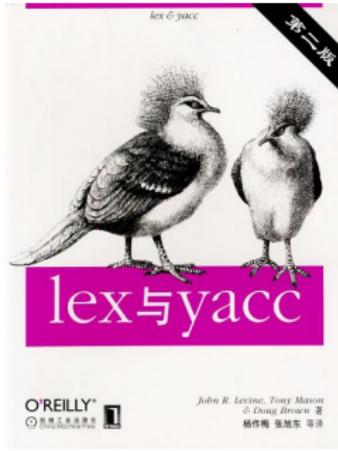
<https://github.com/courses-at-nju-by-hfwei/compilers-lectures/tree/master/2024>



也可使用“本科教学版”



Flex: 词法分析器生成器



Bison: 语法分析器生成器

不够现代, 本学期课程实验**不再支持**这些工具



(Since 1988)



Terence Parr (University of San Francisco)

<https://www.antlr.org/index.html>

<https://www.antlr.org/tools.html> (IntelliJ Plugin)

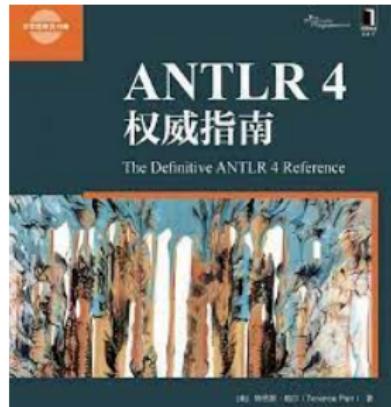
<http://lab.antlr.org/> (Online lab)

The Definitive
ANTLR 4
Reference



Terence Parr

The
Pragmatic
Programmers



基于 ANTLR 4, 是课程实验指导的**重要**参考资料

Language Implementation Patterns

Create Your Own Domain-Specific and General Programming Languages

Terence Parr



Language
Implementation Patterns

编程语言 实现模式

Create Your Own Domain-Specific
and General Programming Languages

[译] Terence Parr 著
李直康 译
高鹏翔 审校



基于 ANTLR 3, 与 ANTLR 4 相比有些过时,
但可以看作理解 ANTLR 4 的基础

自制编译器

How to Develop a Compiler

[日] 青木峰郎 / 著 严基连 焦云 / 译

TING
图灵
图灵原书
设计丛书

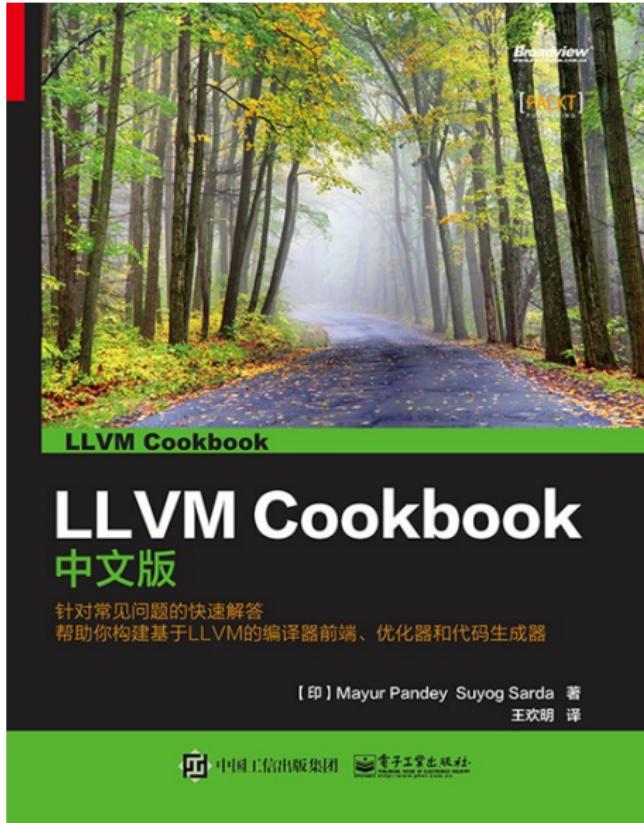


中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

“从零开始制作真正的编译器”，对课程实验很有帮助，**强烈推荐**



从某次实验开始，你就会开始接触 LLVM (<https://llvm.org/>)



LLVM 简介 @ Bilibili

更多参考书，随课程进展陆续发布



更多参考书, 随课程进展陆续发布



<http://docs.compilers.cpl.icu/#/2024/resources>

资源汇总

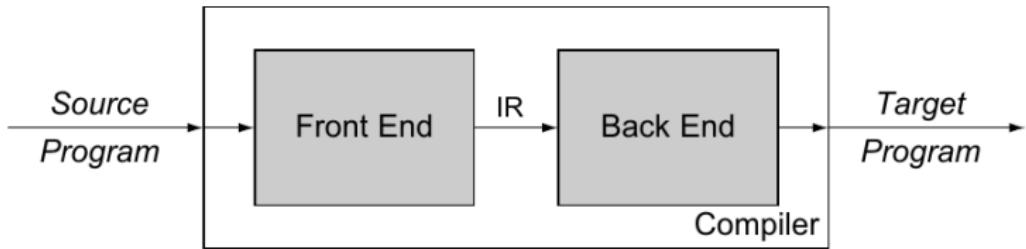
- [compilers-resources @ github](#): 编译原理相关资源
 - [books: 电子书籍](#)

LET'S GET
STARTED





IR: Intermediate Representation (中间表示)

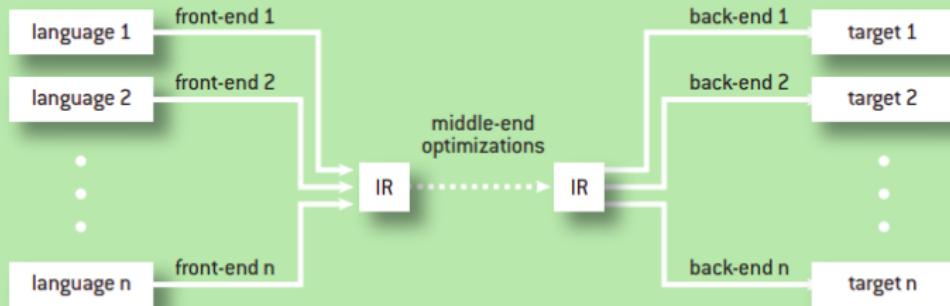


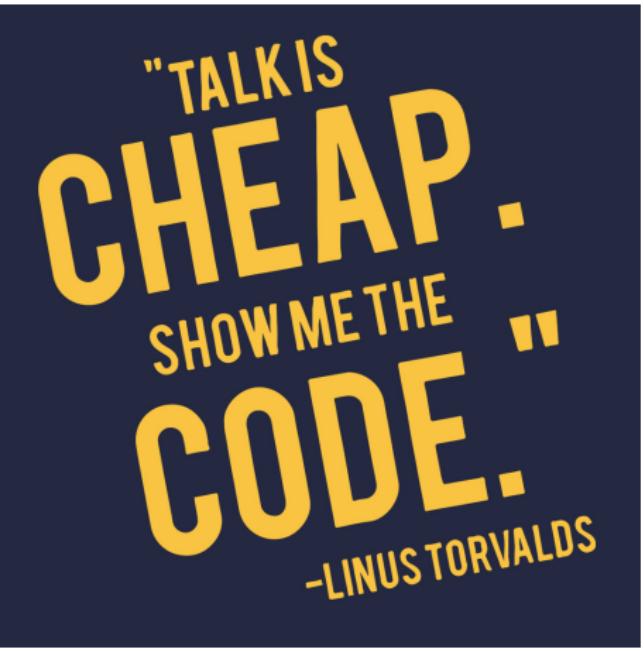
前端（分析阶段）: 分析源语言程序, 收集所有必要的信息

后端（综合阶段）: 利用收集到的信息, 生成目标语言程序

FIGURE
2

A Compiler System Supporting Multiple Languages and Multiple Targets





<https://shorturl.at/bGUV> (Clang 的四种读法)

Clang: a C language family frontend for LLVM

The Clang project provides a language front-end and tooling infrastructure for languages in the C language family (C, C++, Objective C/C++, OpenCL, CUDA, and RenderScript) for the [LLVM](#) project. Both a GCC-compatible compiler driver (`clang`) and an MSVC-compatible compiler driver (`clang-cl.exe`) are provided. You can [get and build](#) the source today.

<https://clang.llvm.org/>

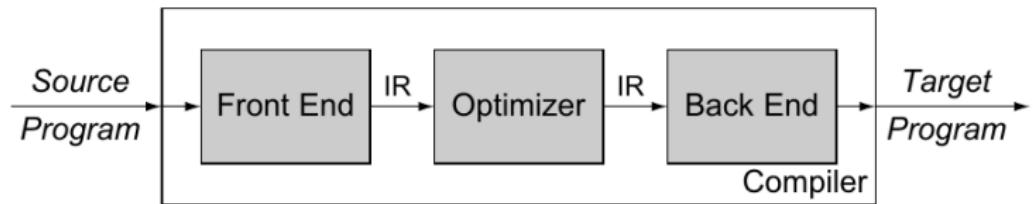
<https://shorturl.at/bGUV> (Clang 的四种读法)

Clang: a C language family frontend for LLVM

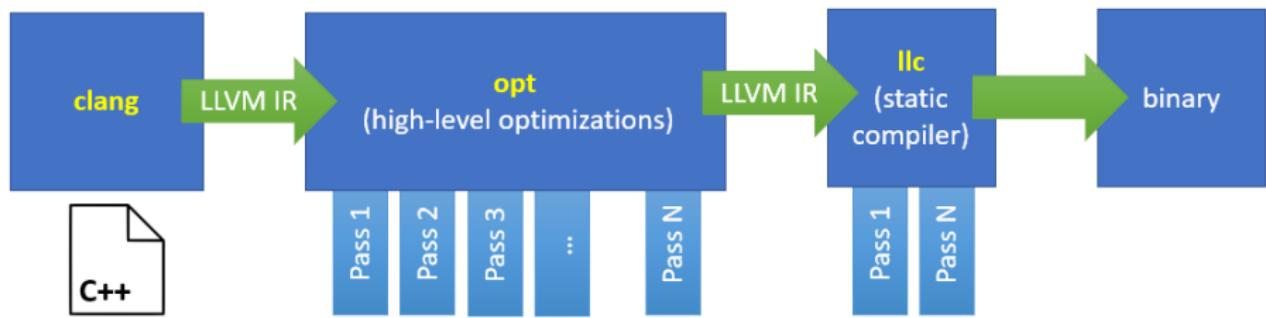
The Clang project provides a language front-end and tooling infrastructure for languages in the C language family (C, C++, Objective C/C++, OpenCL, CUDA, and RenderScript) for the [LLVM](#) project. Both a GCC-compatible compiler driver (`clang`) and an MSVC-compatible compiler driver (`clang-cl.exe`) are provided. You can [get and build](#) the source today.

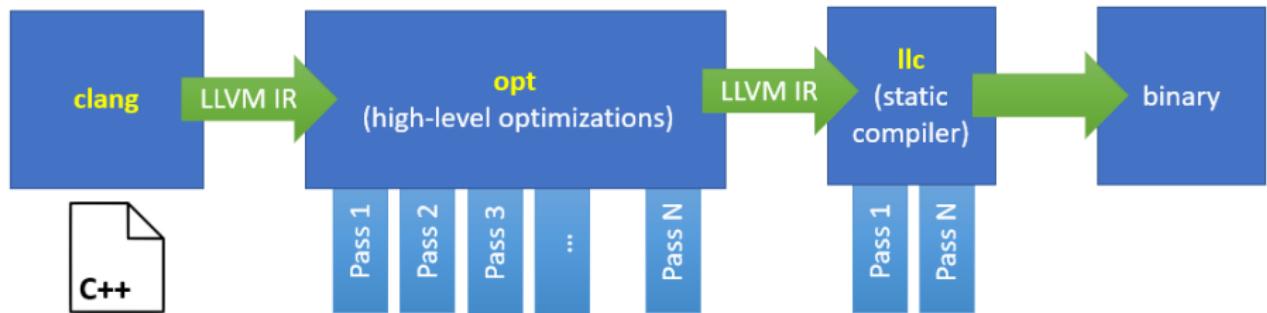
<https://clang.llvm.org/>

(`factorial.c`)



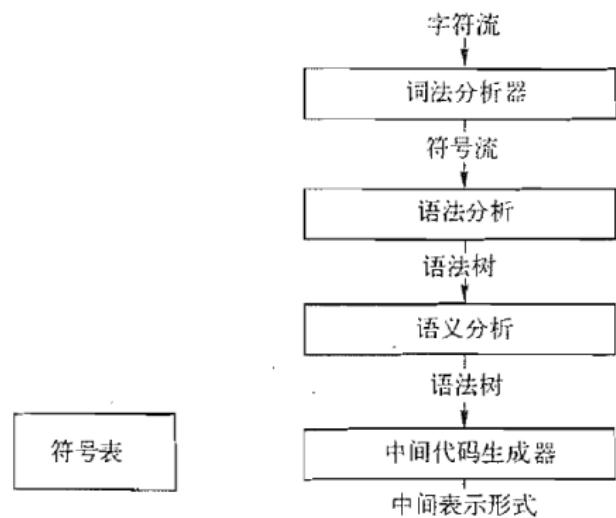
机器无关的中间表示优化



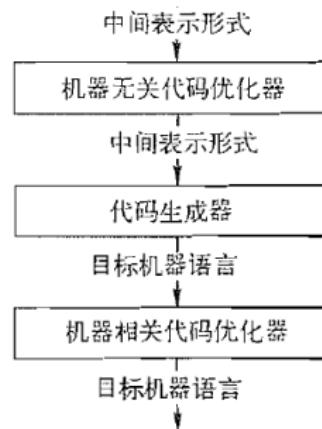


(opt.c)

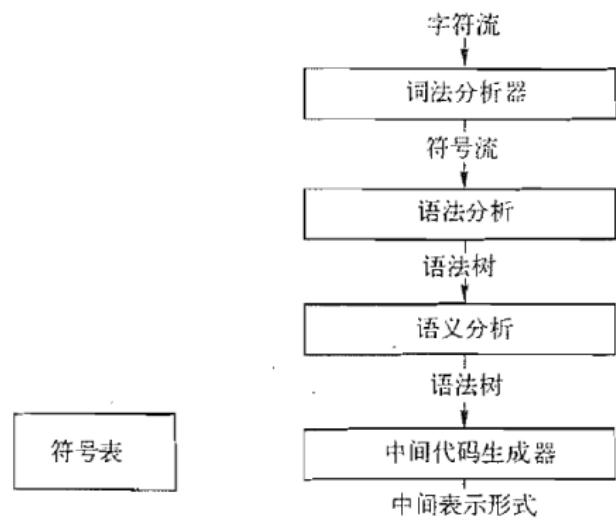
编译器前端



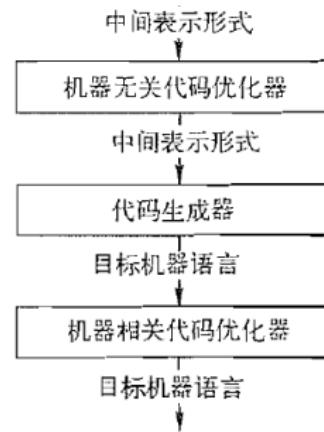
编译器后端



编译器前端

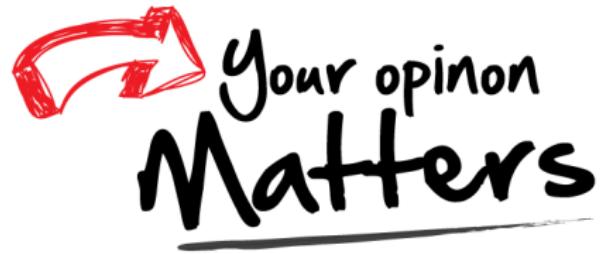


编译器后端



(naming.c)

Thank You!



Office 926

hfwei@nju.edu.cn