

编译实践在线文档

50分钟快速上手

邢其正 (MaxXing)

maxxing@pku.edu.cn



编译实践在线文档

手把手教你从零开始写一个编译器

绝赞开源中！

为了完成实践

你需要做的事情

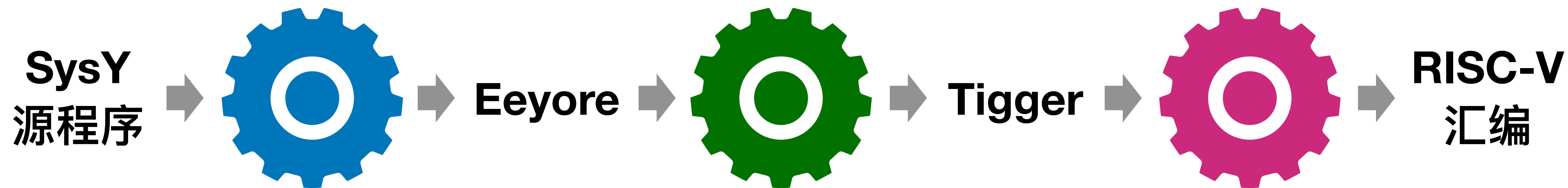
为了完成实践 你需要做的事情

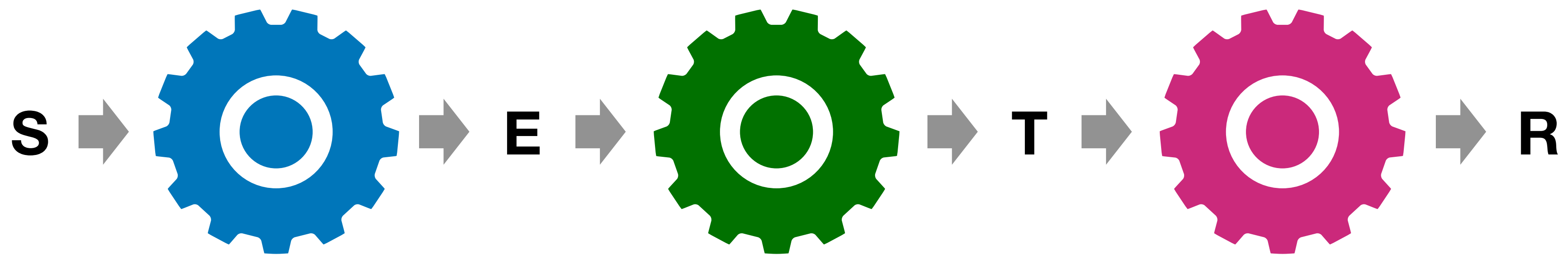
- 阅读文档，并跟随文档的指示配置环境
- 新建文件夹，初始化Git仓库
- 跟随文档，用C/C++/Rust实现编译器
- 在本地进行功能测试
- 在评测系统中进行测试，拿到成绩
- 在DDL前完成面测和实验报告

总结： 看文档就完事了

为什么要设置
在线文档？

以前的实践安排





S2E阶段

做一个SysY的前端
解析变量、控制流、函数、数组
设计并构建AST
也许需要做语义分析
生成Eeyore代码并输出
一次写**几千行**代码！

E2T阶段

前端，可能有AST
寄存器分配，也可以不做
生成Tigger代码并输出
一次写**近千行**代码！

T2R阶段

查表翻译，几乎不需要思考
生成RISC-V汇编并输出
几百行代码搞定！

S2E阶段

开幕雷击

E2T阶段

勉勉强强强

T2R阶段

哇! 超简单

S2E阶段
开幕雷击

E2T阶段
勉勉强强

T2R阶段
哇! 超简单

各阶段难度居然是
逐步下降的?!

阶段划分不合理

且基本靠自己摸索

新版编译实践在线文档

“我们希望给大家尽可能详细的指导”

现在的实践安排



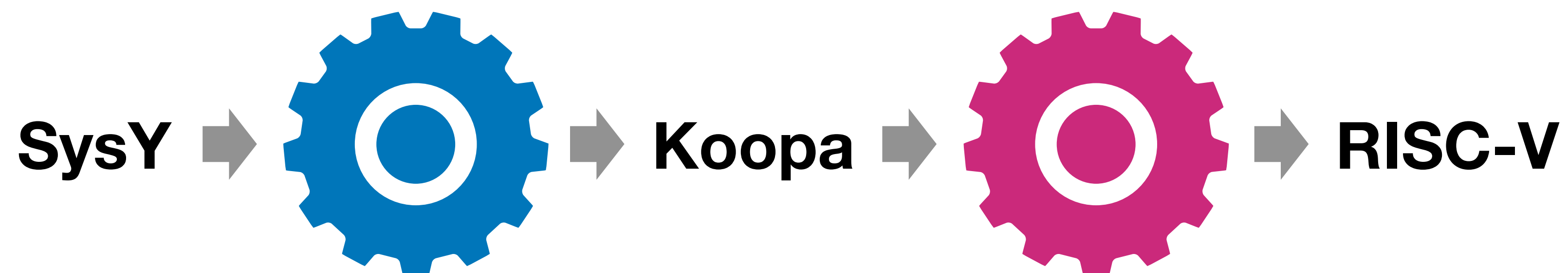
Eeyore/Tigger

v.s.

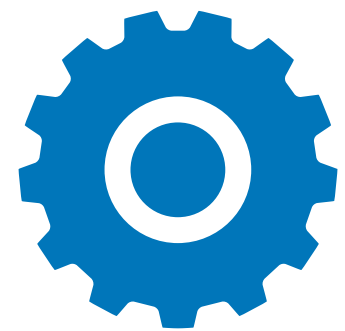
Koopa IR

- 两种**分离**的IR
- 设计一言难尽，**不是很好用**
- 不易在IR层面做优化
- **难以阅读**，不易调试
- 不附带相关处理工具

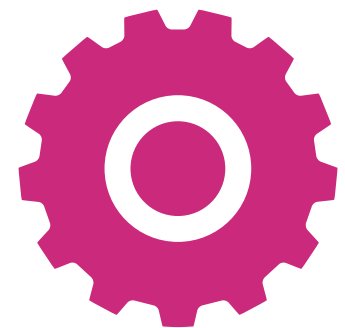
- **格式统一**的IR
- 抽象合理，强类型，**很好用**
- 支持SSA形式，**易于优化**
- **易于人类阅读**和调试
- 附带C/C++/Rust的库



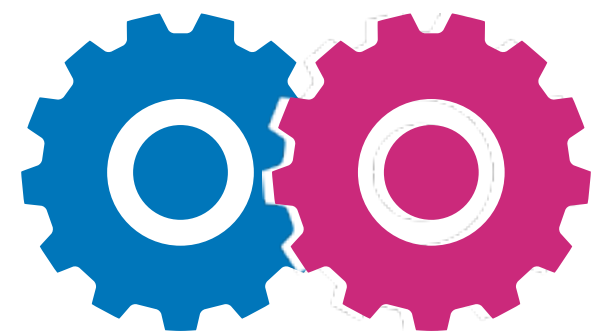
如何划分阶段？



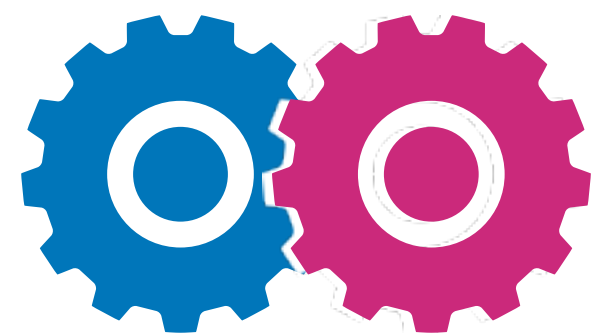
处理main函数（IR生成）



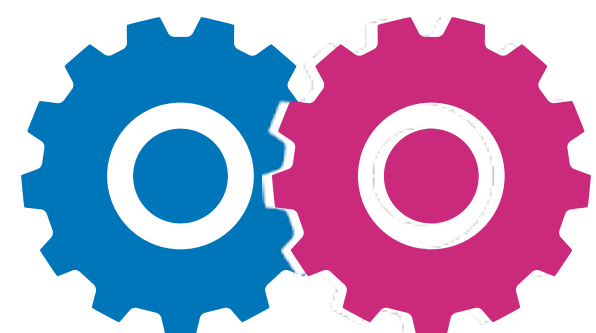
处理main函数（目标代码生成）



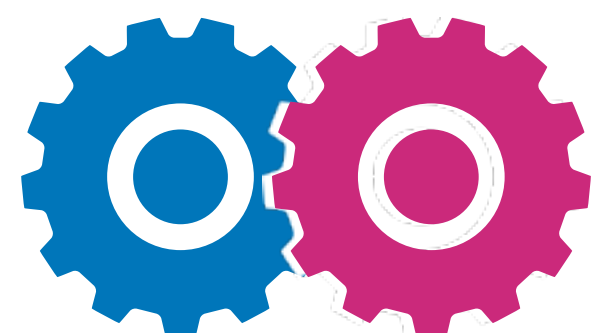
处理表达式



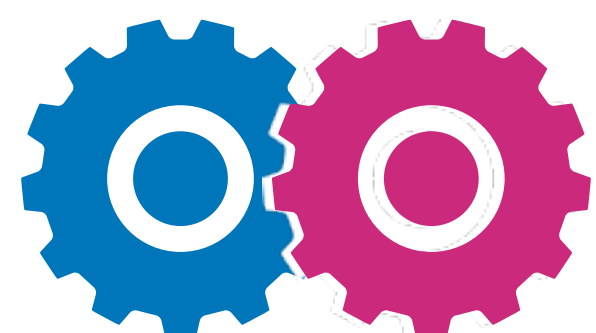
处理变量和常量



处理语句块和作用域



处理各类控制流.....

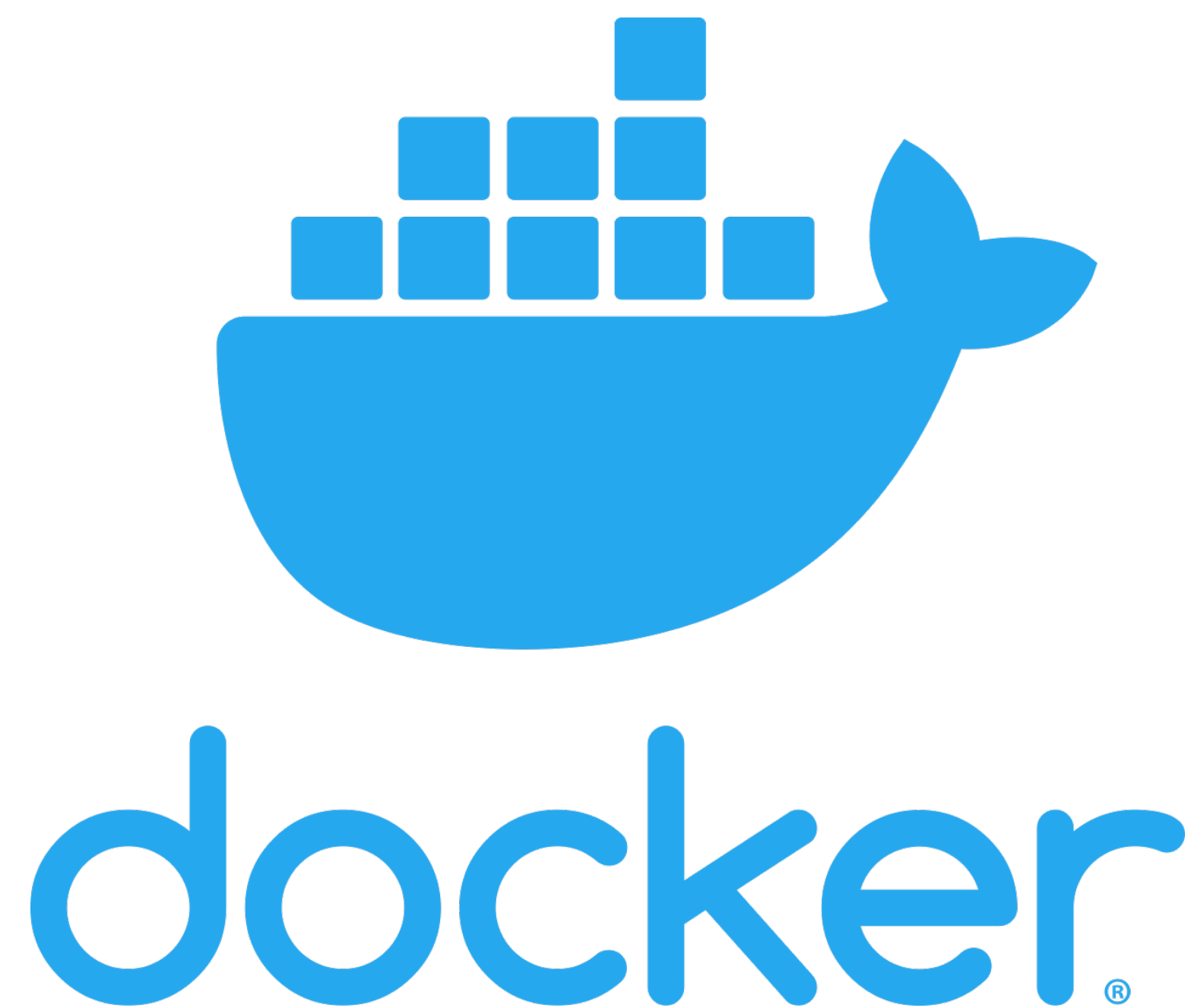


处理函数和数组.....

垂直划分实践阶段

逐步实现编译器

配置实验环境



实验环境基于**Docker**构建

一键配置：`docker pull maxxng/compiler-dev`

全平台适用，开箱即用，用完即删，没有任何残留

Demo

选择项目模板

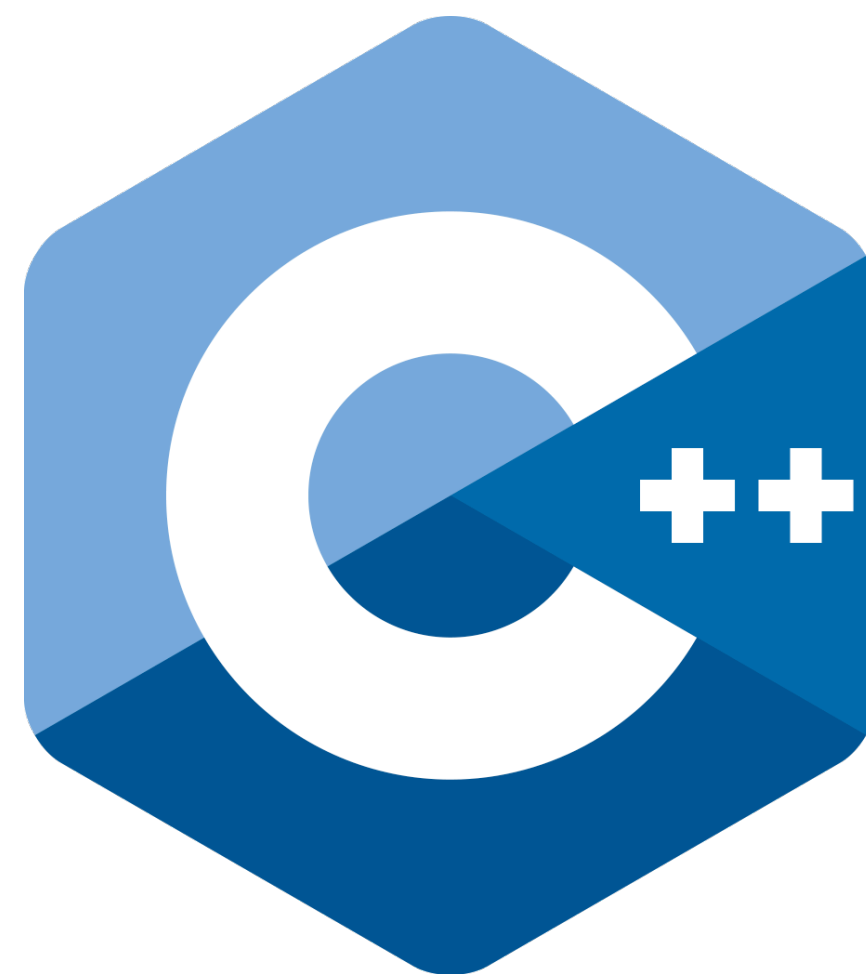
C语言



基于Makefile的模板

基于CMake的模板

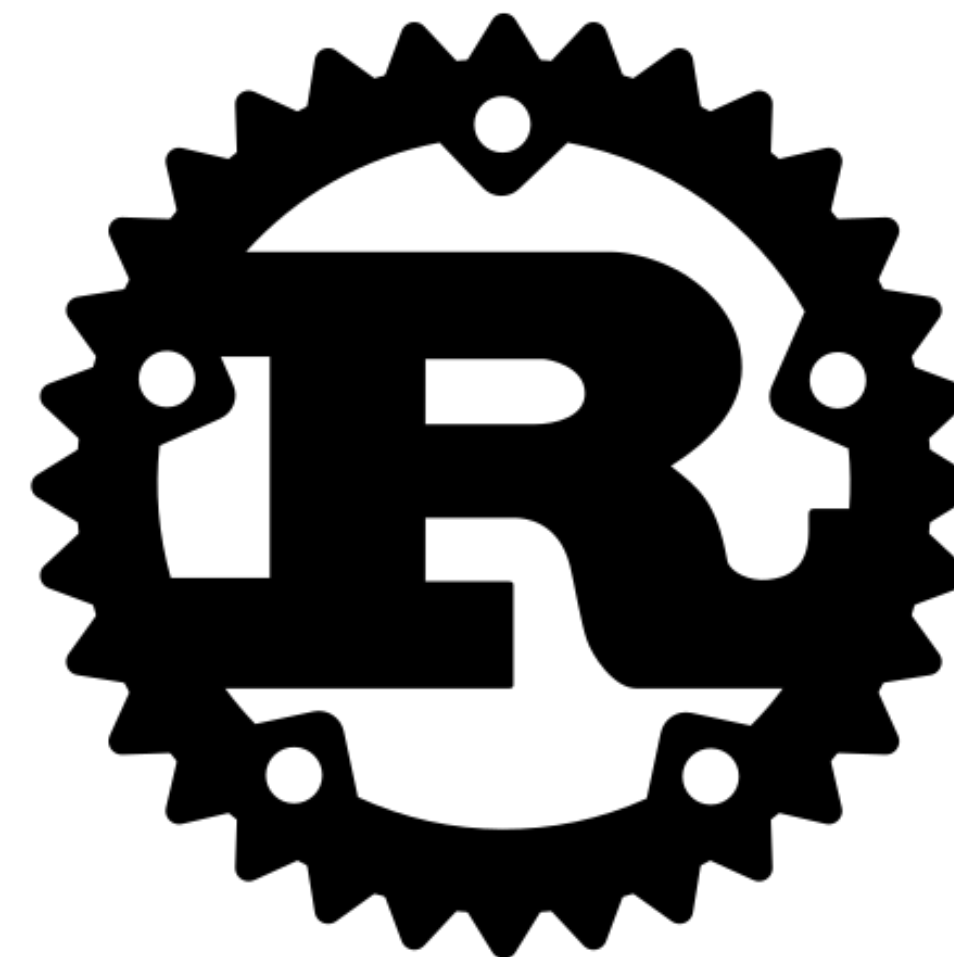
C++



基于Makefile的模板

基于CMake的模板

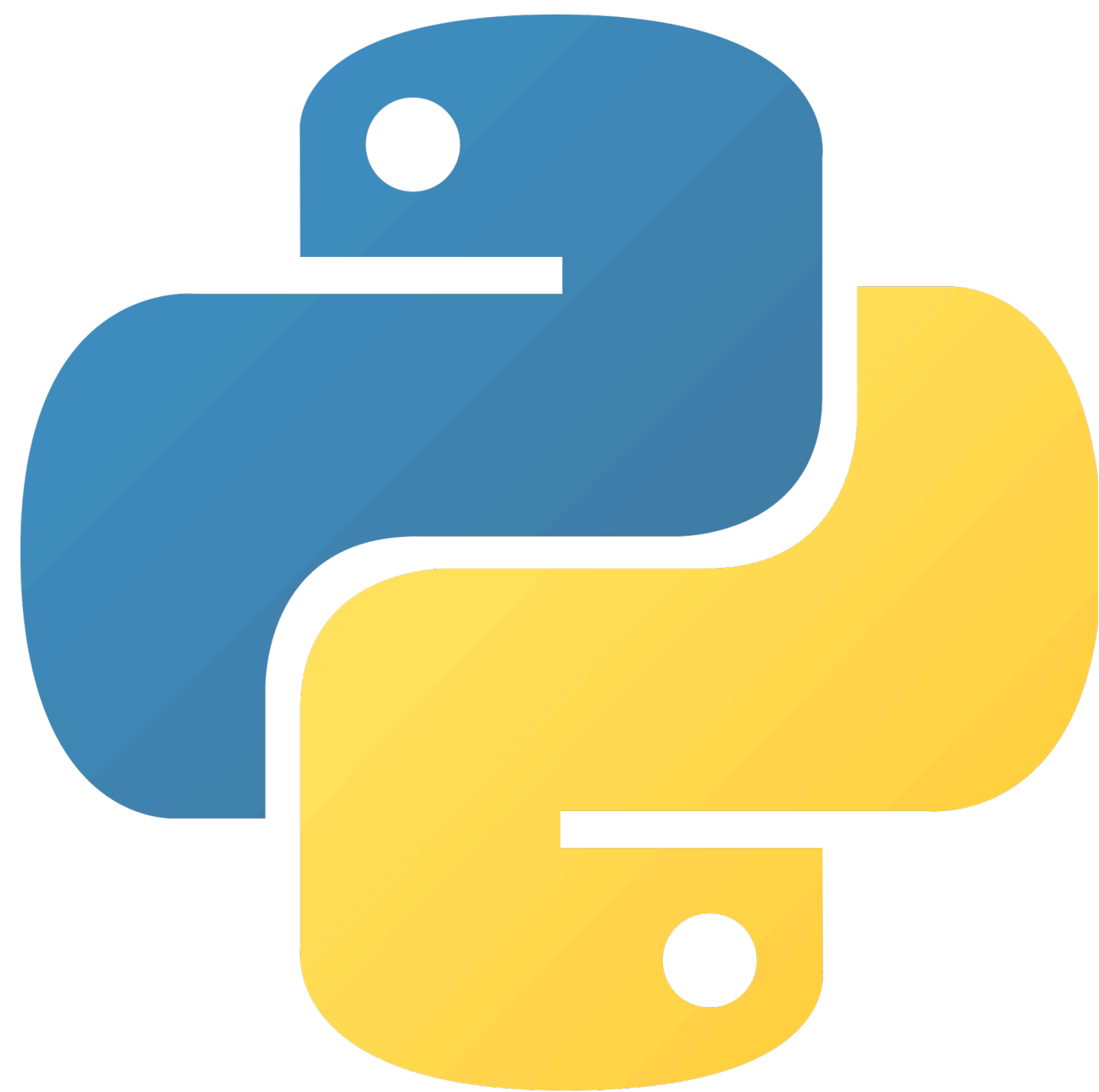
Rust



基于Cargo的模板

(其实根本不需要模板)

进行本地测试



环境内置自动测试脚本

一键测试: `autotest path/to/compiler/repo`

自带130个功能/性能测试用例，支持分阶段测试

Demo

开始写代码



开始写代码

- 把原有的代码和Git目录删掉
- 重新初始化Git仓库
- 逐步添加自己写的代码
- 执行对应的命令来编译/运行项目

Demo

示例编译器



Kira (Rust实现)

Rust版的示例SysY编译器

LALRPOP实现的前端

使用Koopa框架的相关接口构建IR



可以参考示例编译器
但**请勿照搬代码**

Demo

完成实践的

正确姿势

完成实践的正确姿势

- 遇到问题随便改改代码，**应付就完事了**
- 随便改改没用，**不假思索地找身边的大佬求助**
- 求助时问一些**相当没价值**的问题
- 在百度/CSDN上**随便抄抄**，然后遇到一堆问题
- **DDL前三天才开始新建文件夹**，在做了在做了

不要这么做

- 不放过任何一个问题，**包括编译器警告**
- 遇到问题时**思考背后的原因**
- **独立解决问题**，不要怕debug
- 三原则：**STFW、RTFM、RTFSC**
- 远离百度，在Google/Bing上**用英文搜索**

应该这么做

学会如何提问

低质量提问

- 我的编译器segmentation fault了，怎么办？
- RISC-V汇编里能写“add t0, t1, 233”吗？
- 我是Windows，怎么安装Docker？
- C语言里为什么不能#include <iostream>？
- （问各种在文档里强调过的问题）

低质量提问

- 我的编译器segmentation fault了，怎么办？
- RISC-V汇编里能写“add t0, t1, 233”吗？
- 我是Windows，怎么安装Docker？
- C语言里为什么不能#include <iostream>？
- （问各种在文档里强调过的问题）

高质量提问

- 提问前**先STFW等**，尝试找出解决方法
- 提问时附上**报错信息和截图**
- 说明自己在提问前**做了何种尝试，结果如何**
- 说明自己的**思考过程**，以及对问题的分析



编译实践课程问答墙

如果遇到问题，优先在问答墙中搜索

如果找不到相关提问，请在问答墙中提问

我们会及时回复和整理问答墙中的提问

希望每个人都能感受到
编译原理的乐趣

谢谢