编译实践课程

40分钟快速入门

邢其正 · maxxing@pku.edu.cn

要做哪些事呢?

一些具体要求

你要做哪些事

1. 使用**C/C++**开发一个编译器
2. 输入语言为**SysY**，输出为32位**RISC-V汇编**
3. **三个阶段**，Eeyore/Tigger**两种中间表示**
4. 你也可以**完全按照自己的思路**设计编译器

我们会做哪些事

1. 使用C/C++编译器编译你的编译器
2. 使用你的编译器编译功能/性能测试程序
3. 执行编译后的程序，检查是否符合预期
4. 在课程结束前进行最终的面测

大家看了这些要求也许会问很多问题

你要做哪些事

1. 使用**C/C++**开发一个编译器
2. 输入语言为**SysY**，输出为32位**RISC-V汇编**
3. **三个阶段**，Eeyore/Tigger**两种中间表示**
4. 你也可以**完全按照自己的思路**设计编译器

我们会做哪些事

1. 使用C/C++编译器**编译你的编译器**
2. 使用你的编译器**编译功能/性能测试程序**
3. 执行编译后的程序，**检查是否符合预期**
4. 在课程结束前**进行最终的面测**

常见问题

你想问的都在这里

Q: 我可以基于
何种**C/C++标准**编程?

A: 对于C语言程序：**最高C11**

对于C++程序：**最高C++17**

我们会使用**GCC/G++ 8.3.0**编译你的程序

Q: 可以使用Lex/Yacc
或其他ParserGen吗?

A: **Lex/Yacc：可以**

我们会使用Flex 2.6.4/Bison 3.3.2处理相关文件

其他ParserGen：暂时不行

Q: **SysY/Eeyore/Tigger的语法具体是如何定义的?**

A: 我们写了一份**在线文档**来说明这些内容

稍后会给大家介绍详细情况

Q: 按自己的思路实现编译器
最后怎么算成绩?

A: 你的编译器可以不生成Eeyore/Tigger
但最终**必须将SysY转换为RISC-V汇编**
此时**只考察RISC-V汇编的正确性**

Q: 功能/性能测试程序长什么样?

A:

我们提供了一些**公开的测试用例程序**

你可以在GitHub上查看

仓库名为**pku-minic/open-test-cases**

Q: 如何判断编译器输出的程序**功能正确**?

A: 测试程序可能调用库函数**向stdout输出**

主函数的返回值也有所不同

比较你输出程序的执行结果和GCC编译后的结果

Q: 你们会用什么方式编译
我们提交的编译器?

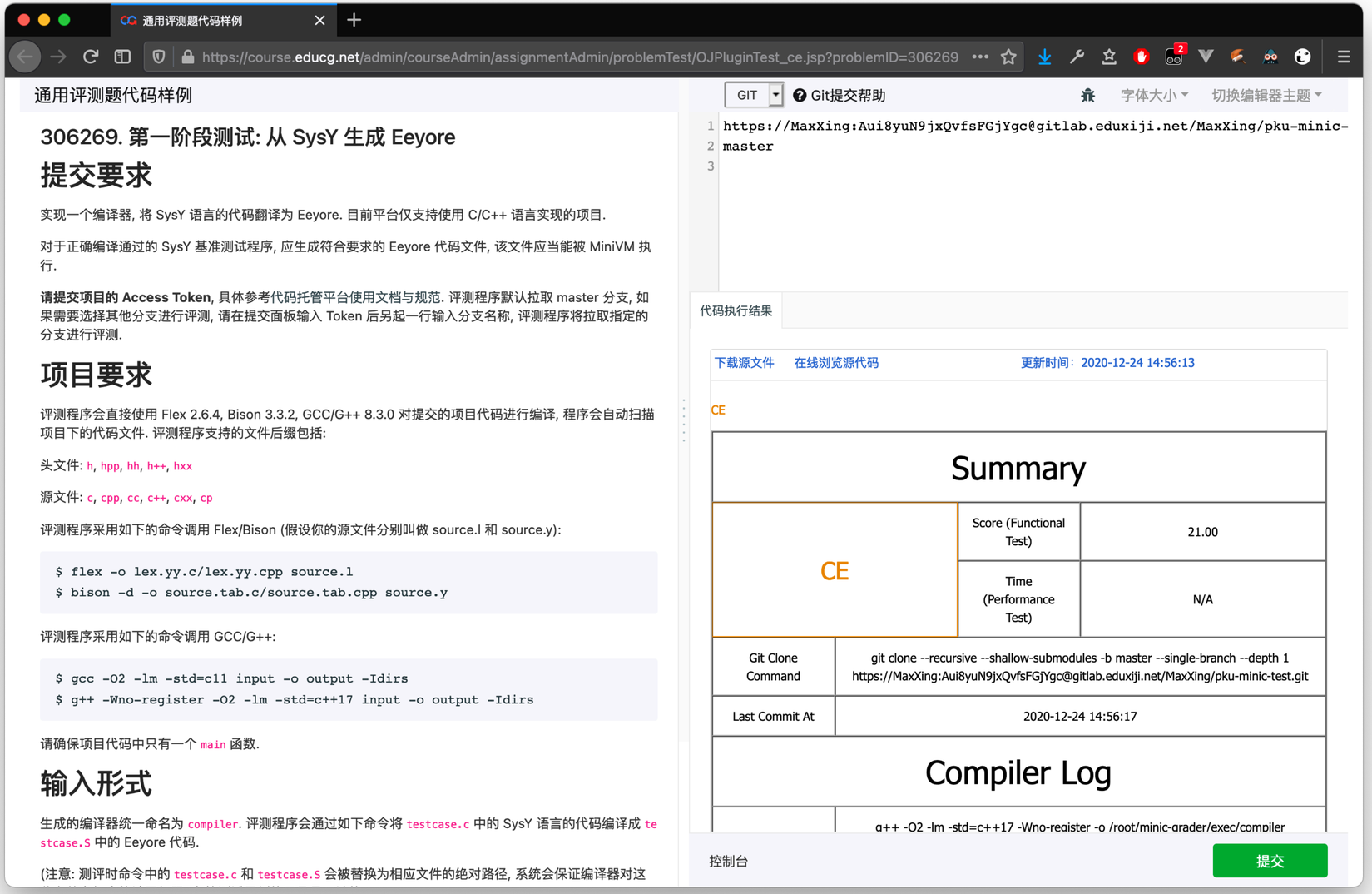
A: 提供**在线评测系统**（类似其他OJ平台）

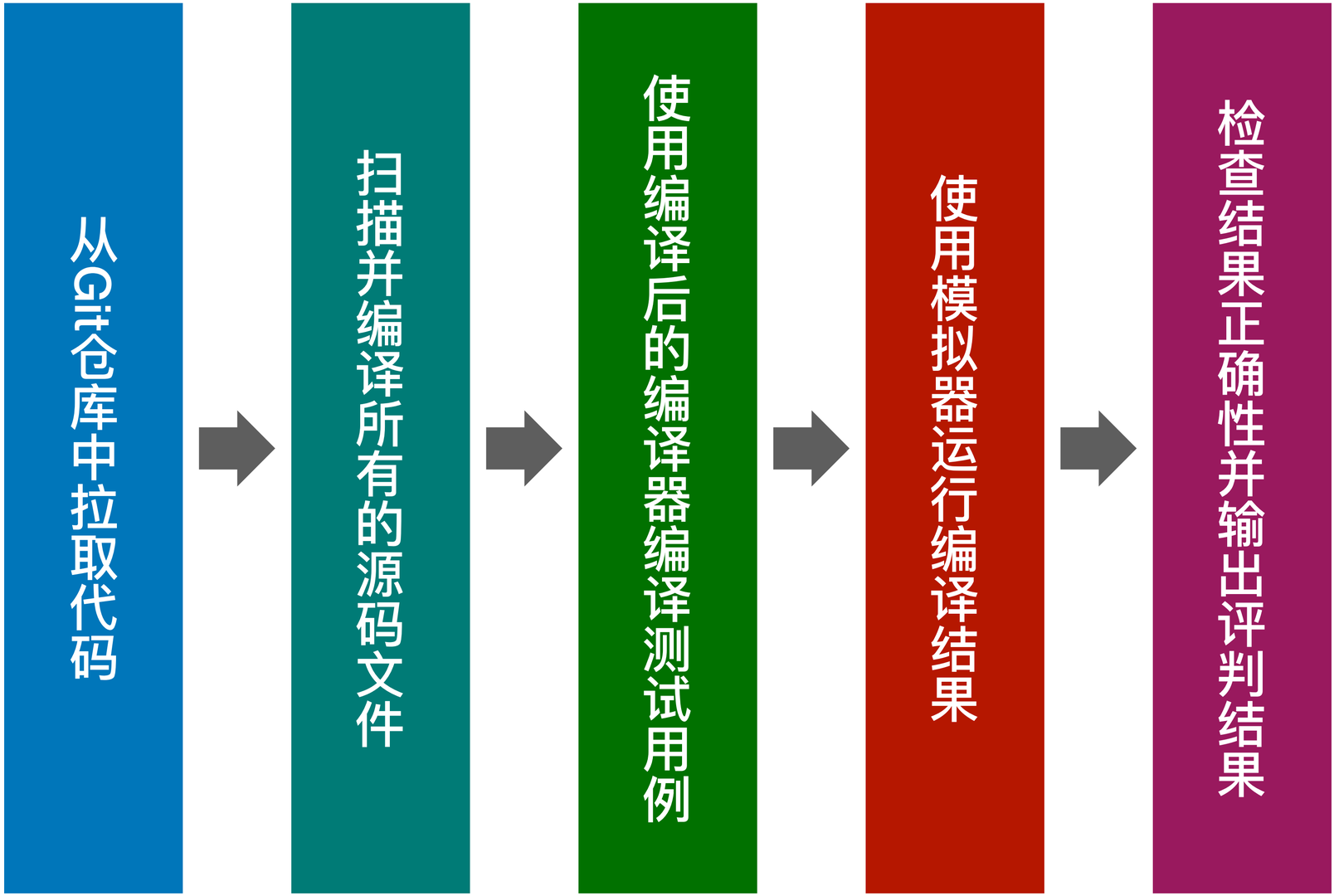
系统会扫描你提交的**Git仓库**

编译扫描到的Flex/Bison/C/C++源文件

在线评测系统

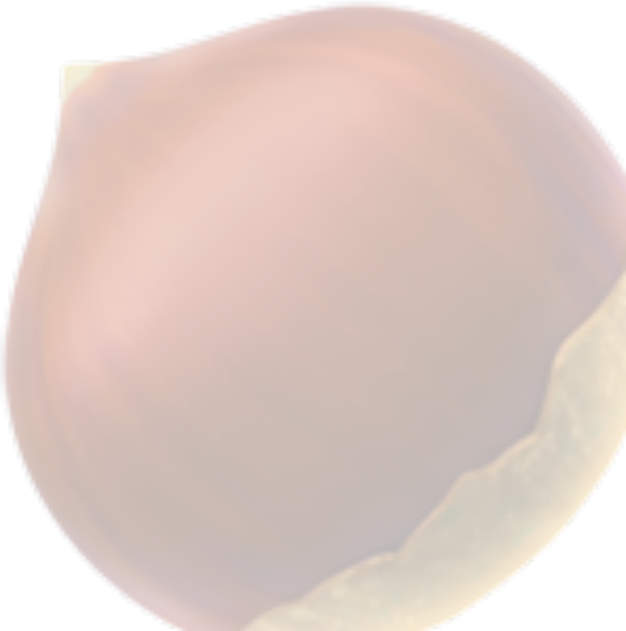
course.educg.net


通用评测题代码样例
https://course.educg.net/admin/courseAdmin/assignmentAdmin/problemTest/OJPluginTest_ce.jsp?problemID=306269
通用评测题代码样例
306269. 第一阶段测试: 从 SysY 生成 Eeyore
提交要求
实现一个编译器, 将 SysY 语言的代码翻译为 Eeyore. 目前平台仅支持使用 C/C++ 语言实现的项目.
对于正确编译通过的 SysY 基准测试程序, 应生成符合要求的 Eeyore 代码文件, 该文件应当能被 MiniVM 执行.
请提交项目的 Access Token, 具体参考代码托管平台使用文档与规范. 评测程序默认拉取 master 分支, 如果需要选择其他分支进行评测, 请在提交面板输入 Token 后另起一行输入分支名称, 评测程序将拉取指定的分支进行评测.
项目要求
评测程序会直接使用 Flex 2.6.4, Bison 3.3.2, GCC/G++ 8.3.0 对提交的项目代码进行编译, 程序会自动扫描项目下的代码文件. 评测程序支持的文件后缀包括:
头文件: h, hpp, hh, h++, hxx
源文件: c, cpp, cc, c++, cxx, cp
评测程序采用如下的命令调用 Flex/Bison (假设你的源文件分别叫做 source.l 和 source.y):
$ flex -o lex.yy.c/lex.yy.cpp source.l
$ bison -d -o source.tab.c/source.tab.cpp source.y
评测程序采用如下的命令调用 GCC/G++:
$ gcc -O2 -lm -std=c11 input -o output -Idirs
$ g++ -Wno-register -O2 -lm -std=c++17 input -o output -Idirs
请确保项目代码中只有一个 main 函数.
输入形式
生成的编译器统一命名为 compiler. 评测程序会通过如下命令将 testcase.c 中的 SysY 语言的代码编译成 testcase.S 中的 Eeyore 代码.
(注意: 测评时命令中的 testcase.c 和 testcase.S 会被替换为相应文件的绝对路径, 系统会保证编译器对这
GIT
Git提交帮助
字体大小
切换编辑器主题
1 https://MaxXing:Aui8yuN9jxQvfsFGjYgc@gitlab.eduxiji.net/MaxXing/pku-minic-
2 master
3
代码执行结果
下载源文件
在线浏览源代码
更新时间：2020-12-24 14:56:13
CE
Summary
CE
Score (Functional Test)
21.00
Time (Performance Test)
N/A
Git Clone Command
git clone --recursive --shallow-submodules -b master --single-branch --depth 1 https://MaxXing:Aui8yuN9jxQvfsFGjYgc@gitlab.eduxiji.net/MaxXing/pku-minic-test.git
Last Commit At
2020-12-24 14:56:17
Compiler Log
g++ -O2 -lm -std=c++17 -Wno-register -o /root/minic-grader/exec/compiler
控制台
提交


从Git仓库中拉取代码
扫描并编译所有的源码文件
使用编译后的编译器编译测试用例
使用模拟器运行编译结果
检查结果正确性并输出评判结果

举个栗子

自动评测系统如何评测你的编译器



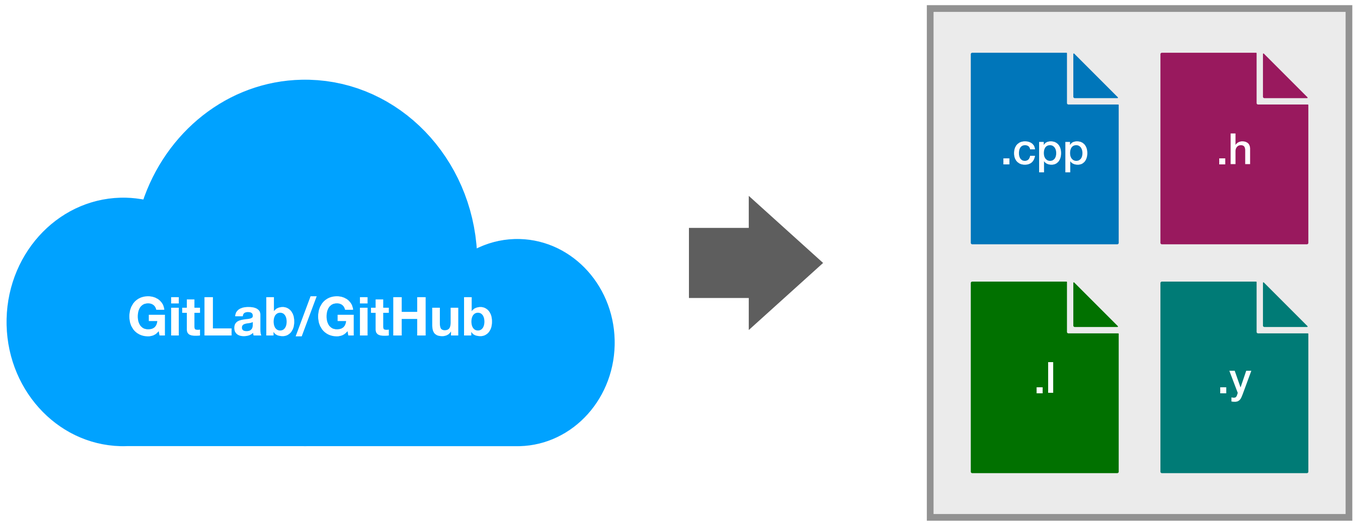
#1

你写好了编译器的代码实现

#2

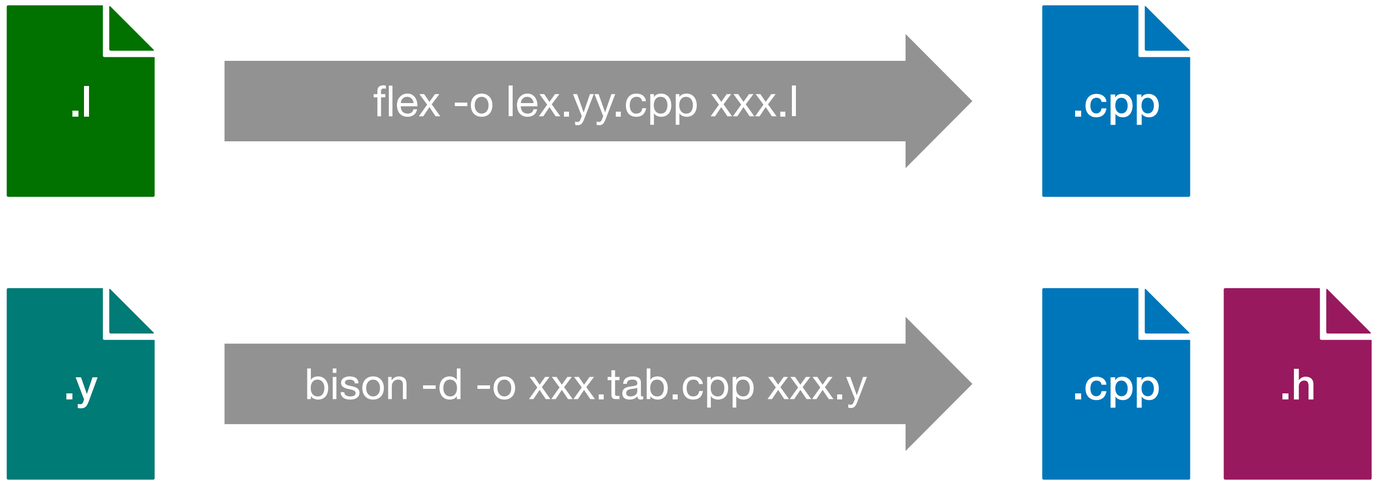
将代码实现上传到托管平台

#3



评测系统从托管平台拉取代码

#4

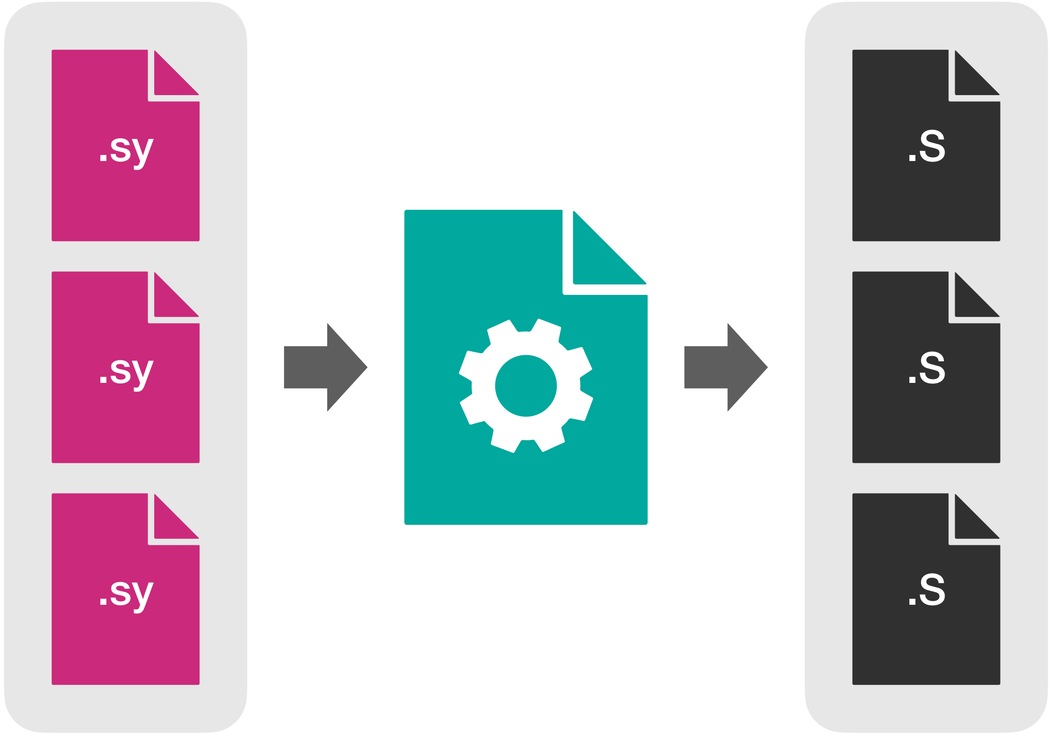


预先处理一部分文件

#5

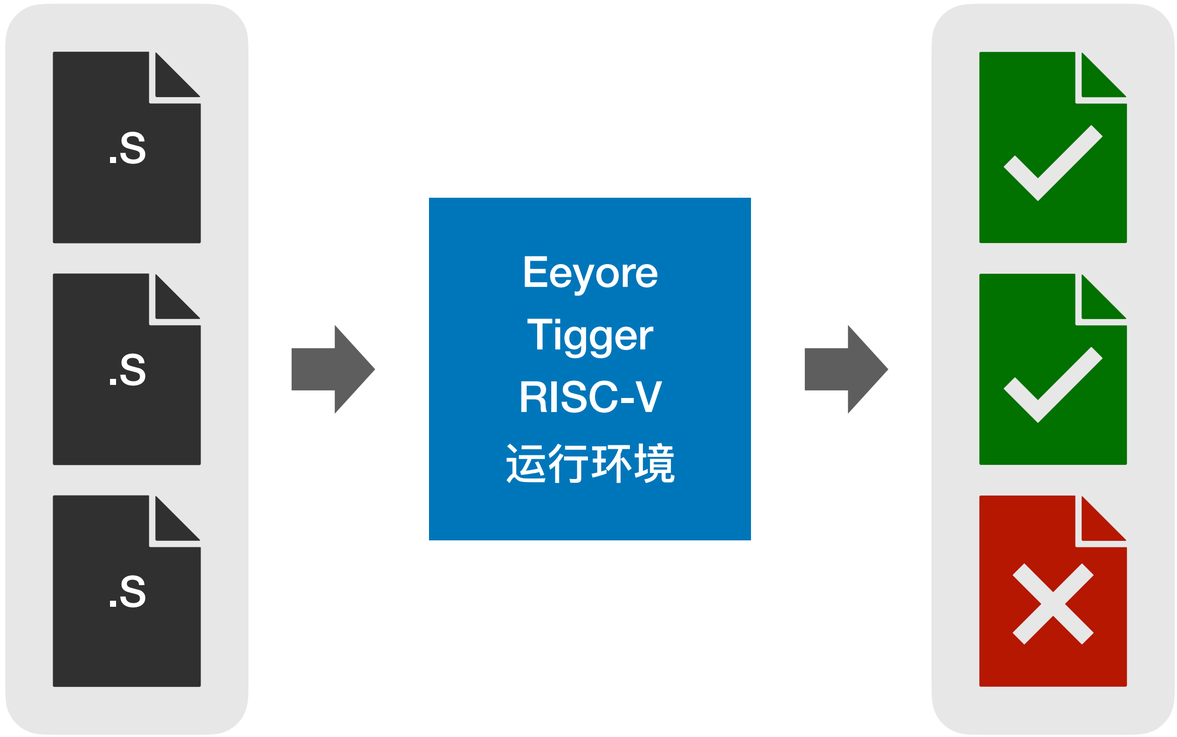
将所有C/C++源文件传给GCC编译

#6



用你的编译器编译所有测试用例

#7



运行编译后的测试用例，汇总最终结果

Demo

如何运行

Eeyore/Tigger

MiniVM

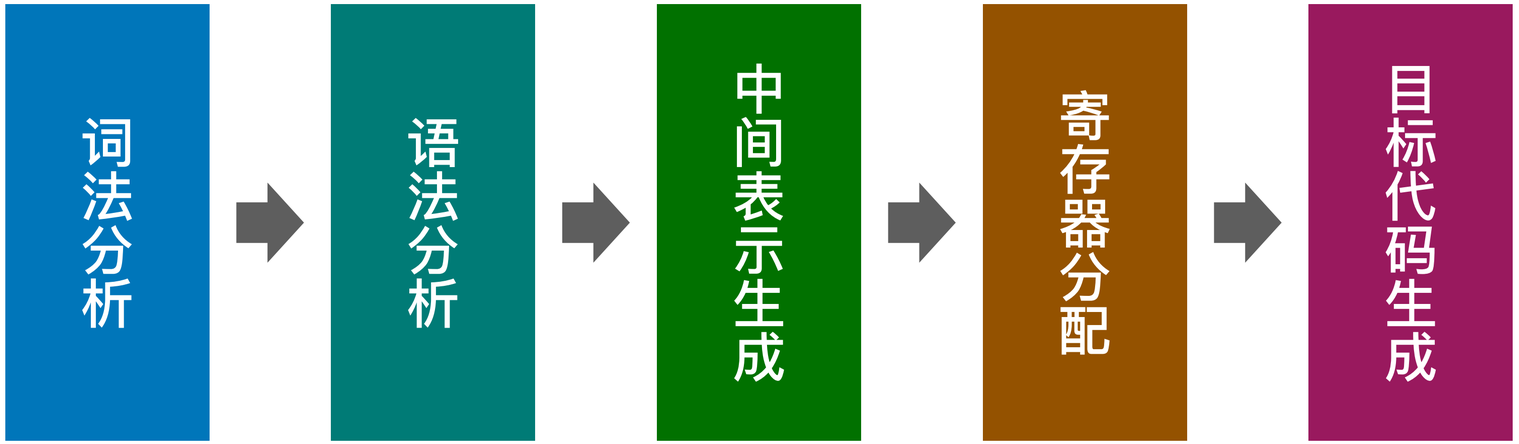
- **专为课程实践设计**的Eeyore/Tigger虚拟机
- **稳定/规范/高效/统一**，可扩展性强
- 具备**功能强大的调试器**，易于调试
- **已于GitHub开源**，接受大家的反馈

Demo

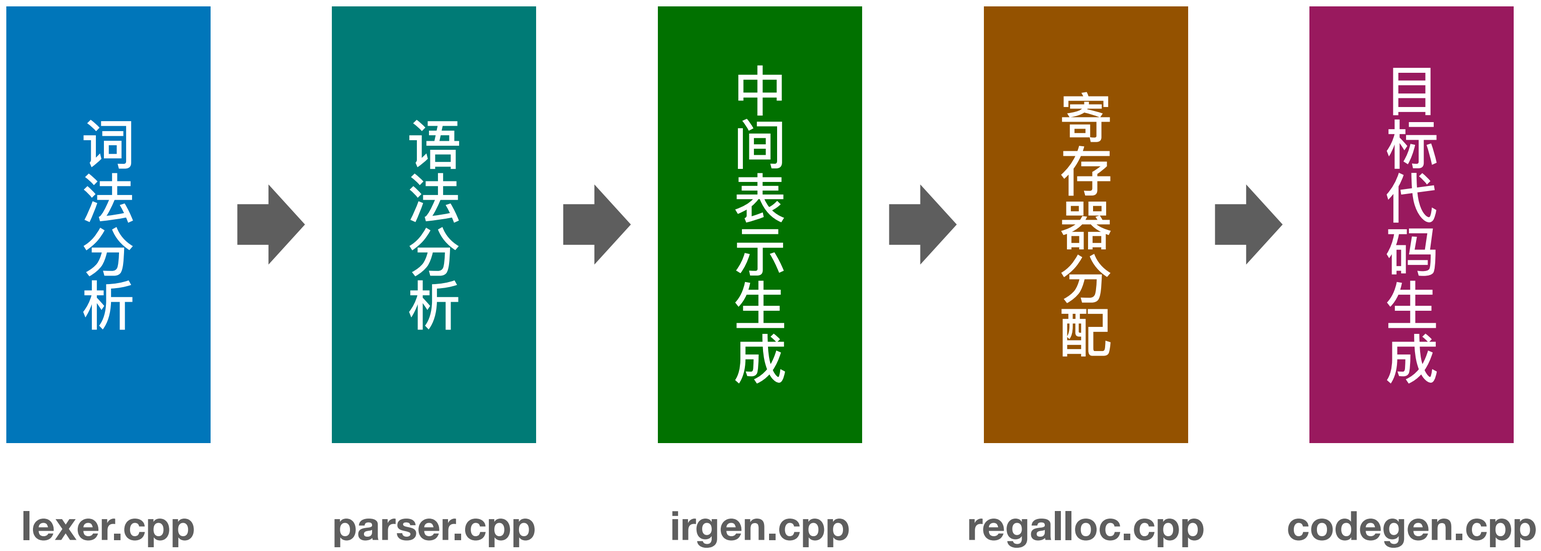
到底应该

怎么写编译器?

怎么写编译器?



怎么写编译器?


词法分析
语法分析
中间表示生成
寄存器分配
目标代码生成
lexer.cpp
parser.cpp
irgen.cpp
regalloc.cpp
codegen.cpp

如何区分词法错误/语法错误/语义错误?

怎么使用递归下降法做语法分析?

怎么理解token流的概念?

我必须实现NFA/DFA/各类算法才能做词法分析吗?

怎么划分token?

令人头大

如何设计AST才能[illegible]语法结构 同时方便处理?

怎么做类型检查?

怎么设计中间表示的形式才能方便进行优化?

MAKE PEOPLE HEAD BIG

寄存器分配怎样最简单? 怎样效果好?

我能做哪些平台无关/相关优化?

如何优雅地遍历各类数据结构?

两个例子

first-step和MimiC

first-step

实现简单，**只有1300行**

不使用除标准库外的**任何依赖**

具备**解释器和编译器**的实现

将first-step语言**编译到RISC-V汇编**

MimiC

完整的**带优化的编译器**实现

不使用除标准库外的**任何依赖**

第一届编译大赛**参赛作品**，排行第七

将SysY语言**编译到ARM或RISC-V**

Demo

你说的东西太多了

我记不住

编译课程实践

在线文档

pku-minic.github.io/online-doc

托管于GitHub Pages，**便于更新和维护**

开放评论区，接受大家的讨论和反馈

希望每一个人都能

感受编译原理的乐趣

謝謝