

# 编译实验课大作业框架总体情况

本学期的编译实验有3个框架可以选择，其中C++语言有两个框架，分别是RISC-V后端和armv7后端。rust语言有一个框架，后端为RISC-V。

同时同学们可以不选择已有的框架，自行从零开始编写你的编译器，但是测试用例依旧要使用目前框架中的，并且你需要自行编写测试脚本并在线下检查时向助教讲解。

## C++语言编写的RISC-V框架简介

该框架改编自2024编译系统实现赛RISC-V赛道一等奖作品，源代码仓库链接：

<https://github.com/yuhuifishash/SysY>

框架仓库的链接：<https://github.com/yuhuifishash/NKU-Compilers2024-RV64GC>

环境配置请参考环境配置实验指导书，指导书中提到的工具均已在共享文件夹中上传

[https://nankai.feishu.cn/drive/folder/PLp0fytJLlz4kCdsVu0cVJ5qnWb?from=from\\_copylink](https://nankai.feishu.cn/drive/folder/PLp0fytJLlz4kCdsVu0cVJ5qnWb?from=from_copylink)

框架采用C++语言编写，没有过多地使用现代C++特性(其实是助教用的也不熟练)，同学们在大一学习的C++语法已经足够应对本学期的编译实验。

代码量估计(以实现所有进阶要求，拿到编译实验满分为标准，只是一个粗略的估计，具体情况需要根据你的代码质量来看)：

1. 词法分析：200-300行
2. 语法分析：300-400行
3. 类型检查和中间代码生成：2000-3000行（如果你代码复用性强，可能会少于2000行）
4. 代码优化：基本要求约200行（支配树+完整mem2reg约400行，其余进阶要求根据你的选择会有所不同）
5. 目标代码生成：1000-1500行（大部分代码量在指令选择的苦力活，寄存器分配主要在于阅读代码）

arm后端框架由于只提供了基础要求的内容，所以代码量会稍多于RISC-V后端框架版本。

rust语言框架代码量和难度要远大于C++语言

关于C++语言编写的RISC-V后端框架的其他问题请与助教华志远联系。

## 选择C++语言编写的RISC-V框架的优势：

1. 框架包括进阶要求在内的所有类的定义，输入输出均已全部写好，包括**词法分析每个token的输出，所有语法树节点类的定义和输出，所有LLVM-IR类的定义和输出，所有RISC-V汇编指令类的定义和输出**。同学们在实验时只需要注重算法的实现细节即可。
2. 框架改编自竞赛作品，并且该比赛的项目已开源，所以有公开的项目可以参考。（参考时记得在注释中标明，少量参考不会扣分，但你必须保证你读懂了你参考的代码）
3. 本学期新增了代码优化内容，同学们只需要在框架的基础上填写函数即可，如果选择arm后端的框架可能需要自己探索一部分内容。
4. RISC-V后端为64位指令集，并且RISC-V指令集非常精简，很容易就能上手（操作系统实验应该也是RISC-V）。同时你不需要处理armv7的一些非常麻烦的特性(例如armv7的文字池，事实上，armv7对于全局变量的处理比较麻烦)
5. 之后会在群中发一个共享文档，同学们对框架有什么问题可以直接写在共享文档中，助教会定期回答。（例如某个变量是什么意思，某个函数的功能是什么等，同时考虑到今年是第一次RISC-V架构实验，关于RISC-V预备工作的问题也可以直接提出）

请注意不是特别隐私的问题不要私聊助教，直接写在文档中即可。

所以同学们不用担心选择新框架会像CMU-15445一样完全不知道自己要干什么

## 选择C++语言编写的RISC-V框架的劣势：

1. 框架的后端(目标代码生成实验)需要阅读的代码较多，尤其是寄存器分配环节，总体大概有2000行左右需要阅读的代码。（读懂后完型填空就比较简单了，当然如果你不想阅读代码，或者想自由发挥的话也可以自行选择寄存器分配算法重写）
2. 框架设计时只考虑了SysY的文法，如果你想扩展到完整的C语言，可能不是很方便，需要额外做很多工作，甚至对框架进行重构。
3. 可以参考的代码只有一个竞赛作品，可能不像arm后端那样有学长丰富的参考资源
4. 由于大部分类的定义均已写好，可能不太适合想自由发挥的同学。

## 一些注意事项

1. 参考一定要加注释，具体见上机大作业要求文档
2. 尽早开始你的工作，各实验代码量差距较大，推荐提前进行下一步的实验。
3. 注意保存你的gitlog并上传到git server，每完成一点代码就commit一次，一是可以保存历史版本代码，防止代码丢失或新版本代码出错。二是如果你的代码被我们怀疑抄袭，gitlog是能证明你代码是自己写的一个证据。

4. 检查时除了展示运行结果外，助教会随机选几处你写的代码，你需要现场快速回答出该代码的意思。
5. 目前框架代码的语法测试用例，注释均已完成，后续可能根据同学们的反馈情况增加一些注释。同时代码优化的测试样例和测试脚本会在后续给出。（请同学们注意框架仓库的变化并及时merge，每次实验开始前如果有变动会在群里通知）
6. 如果你的提问可以轻易的通过搜索引擎找到答案或者你的问题描述地模糊不清，助教不会进行回复。