

『编译技术』 SysY-Mips编译器设计——实验总结

章节目录

- [『编译技术』 SysY-Mips编译器设计——总体设计概述](#)
- [『编译技术』 SysY-Mips编译器设计——词法分析](#)
- [『编译技术』 SysY-Mips编译器设计——语法分析](#)
- [『编译技术』 SysY-Mips编译器设计——语义分析\(符号表管理与错误处理\)](#)
- [『编译技术』 SysY-Mips编译器设计——中间代码LLVM生成](#)
- [『编译技术』 SysY-Mips编译器设计——目标代码Mips生成](#)
- [『编译技术』 SysY-Mips编译器设计——中端代码优化](#)
- [『编译技术』 SysY-Mips编译器设计——后端代码优化](#)
- [『编译技术』 SysY-Mips编译器设计——实验总结](#)

实验得分

最终八个点得分如下所示，排名除第六个点外均位于20名附近，而第六个点来到50+。第六个点较为反常，出现了分配给其的寄存器越多，cycle反而越高的情况，mem执行较多，猜测是使用了较深的递归而产生了大量寄存器传参冲突。

#	提交备注	testfile1	testfile2	testfile3	testfile4	testfile5	testfile6	testfile7	testfile8
121	1	6533.0	1509.0	110072.0	61333.0	13022587.0	7034155.0	61100.0	15401617.0
122	1	6533.0	1509.0	110072.0	61333.0	13022587.0	6961373.0	61100.0	15401617.0
123	1	6533.0	1509.0	110072.0	61333.0	13022587.0	6733105.0	61100.0	15431625.0

实验总结

总而言之，从一个月前正式着手进行优化，再到如今优化后代码来到1w+行，即使循环外提等关键优化未来得及细细琢磨，排名似乎也不是很理想，这场优化之路至此也将走向终点。

这是一次充满艰辛的苦痛旅程，也是一次锻炼自己的绝佳机会，我收获到了不只是编译优化方面还有图论算法等等的知识，也练就了自己去看懂伪代码，看不懂就自己造算法的见山开山，见水渡水的执着心态，最后再次感谢坚持一路走来的自己，感谢这学期和我一同讨论编译的朋友们，也感谢课程组和老师们，感谢他们为我们带来了不逊色于coos的编译实验。纵使心中万般不舍，也不得不为这学期的努力画上句号，编译再见噜，下一次接触编译就是作为助教帮助大家了！

最后我想说编译在我心中不愧是oo的promax升级版，最后的最后献上我的巨巨巨型类图，从左至右依次为前端架构，中端架构，后端架构。



