

2019级软件学院编译原理实验规划

语言介绍

SysY语言是本实验的源语言。是一个C语言的子集，sysy语言是单文件的，去除了C语言中的include/define/pointer/struct等较复杂特性，留下的具有实现挑战性的内容有短路求值，不限上限的多维数组。sysy语言本身不具有IO功能，通过链接运行时库的方式进行IO。

LLVM 是一个模块化的、可重用的编译器和工具链的集合，目的是提供一个现代的、基于SSA的、能够支持任意静态和动态编译的编程语言的编译策略。在最近几年已经成为表现上能够和gcc对标的项目。

LLVM IR 是LLVM项目中通用的中间代码格式，用于连接其他语言和目标体系架构 $n \times 1 \times m$ 中的1，（在目前的计划中）是学生需要从源语言中编译并翻译到的目标语言。

实验目的

本实验希望通过将复杂且庞大的SysY语言的特性拆分为多个小的模块与实验，引导同学们在思考-设计-实现-重新设计的过程中了解编译技术的整个流程，并且掌握一定的优化能力。

若完成所有实验，同学应该：

- 学会并理解了编译器的翻译流程
- 掌握了词法分析与语法分析的原理以及操作
- 掌握了语法制导翻译的原理以及操作
- 了解了编译优化的思想
- 掌握了部分简单的编译优化
- 熟悉了gcc,llvm等常见编译套件的使用以及用法

实验中不会涉及的东西：

- 手动内存管理
- 寄存器分配
- 体系结构相关的优化

为什么选择 llvm ir 而不是 gcc ir , mips 汇编以及其他体系架构或者是栈式虚拟机：

gcc ir过于复杂，学习成本太高，不适宜用作教学实验。

软件学院并没有系统性地学习过CISC或者是RISC指令集，而且下个学期软件学院其他课程过于繁重，增加指令集的学习会导致学习内容大幅增加。此外，根据了解的情况，由于时间关系，计算机学院的大多数人最终版本的编译器也没有实现对寄存器的分配。llvm ir本身不需要太多对体系架构的知识，并且完全能够胜任一门目标语言的工作。

栈式虚拟机本身结构较为简单，并且生成对应代码也比较简单，但是脱离了现在实际应用中的环境，学习栈式虚拟机和学习llvm ir的成本相当，但llvm ir具有实际且广泛的应用场景。

具体实验设计

总共分为6-7次实验，实验之间具有先后关系，评测方式暂定为使用上届助教的评测机，大致实验内容为

助教全程工作

- 维护评测系统
- 实验答疑
- 编写实验指导
- 编写示例编译器(会尽快)

0 pre (预计需要 3h)

学生工作

LLVM工具链的熟悉与使用

ANTLR工具链的熟悉与使用

词法分析与语法分析的小实验

助教工作

编写LLVM工具链的教程，编写ANTLR工具链的教程

指导学生从现有LLVM IR中选取需要的指令

介绍LLVM IR中最关键的几个概念(User Value Operand等)

设计词法与语法分析的小实验

1 仅有main函数的编译器 (5-7h)

- part 1 设计实现从AST翻译到LLVM IR的架构，并实现最简单的从main函数的返回的 codegen Middle

2 常量表达式 (3h)

- part 2 实现正号，负号，逻辑非 Easy
- part 3 实现加减乘除以及模运算 Easy
- part 4 实现关系表达式 Easy

3 局部变量与语句 (3h)

- part 5 局部变量和赋值
- part 6 if语句与条件表达式

4 加入作用域与循环 (5h)

- part 7 作用域与块
- part 8 循环语句
- part 9 continue 与 break 与代码回填

5 数组 (2h)

- part 10 多维数组

6 函数与全局变量 (2h)

- part 11 函数
- part 12 全局变量

7 挑战实验

- 13 mem2reg
- 14 死代码删除 (做出mem2reg以后这个基本是送分)
- 15 函数内联 (因为没有函数声明，所以只有调用自身的递归，还是比较简单的)
- 16 短路求值 (需要提前放出，不然到后面代码逻辑得大改)

