

# 思路

学号：19373440 姓名：王雨飞

## 开始前的分析

- 1.发现这个 lab3 的难度非常非常高。
- 2.lab3 要求生成 LLVM IR 格式的文本。在我的 lab2 中我躲掉了这个的实现(因为觉得难度很高)，结果 lab3 就要求了，这也就意味着我需要完全重构 Exp 运算的代码。
- 3.如果继续使用 C 语言，面向过程的语言在处理这样非常复杂的数据结构时会非常力不从心。比如说标识符和寄存器之间的对应关系，毫无疑问如果有 Hashmap 的话会比较好实现。所以要考虑转为 C++或者完全重构一个 JAVA 代码。
- 4.在 lab2 中我使用递归来求出四则运算的值，现在我打算继续沿用这个递归，不过不直接求出值，而是输出 LLVM IR 文本。

## 写代码

- 1.一开始，我想继续用 C 语言来写(直接用代码来实现 Hashmap)，后来逐渐意识到生命的可贵，所以还是转成了 C++。

### 第一次重构

- 2.第一次重构是失败的，其实第一次也不能说是重构。第一次我首先尝试让 lab2 中的递归不直接算出一个值，而是在每个递归函数中都 fprintf 一些 LLVM IR 文本，但是在处理寄存器以及 LLVM IR 文法输出顺序时都出现了比较大的问题。
- 3.关于文法输出顺序，我想的是使用栈来解决。然后关于寄存器，我想使用 LLVM IR 来解决。
- 4.这里就出现了不少的问题。其一是我 lab2 的代码写的不太清晰，我对当前 token 的判断是根据 token 的内容，比如说 Ident(xxx)，表示这是个 Ident，其实这个是比较麻烦的，使用栈来解决的话，又要让栈中的某个 Ident 和 HashMap 中的寄存器对应，对于栈中的符号和数字，又不能对应。这样的话就导致越写越觉得乱，最后脑子已经不清晰了。
- 5.我希望的是能够更高效的将栈中可能存在的 Ident，包括 Exp 中的各个元素，用更好的方式来储存这些东西。所以，将数据结构较大幅度的重构，就是下面的第二次重构。

### 第二次重构

- 6.为变量和 Exp 中的元素分别设置了结构体，这样的话也就意味着可以为它们设置属性(以前从没用过 C++，但是这个感觉就像是面向对象的代码了)。为变量设置了属性：是否是常量（针对语义约束）；对应的寄存器的号码。为 Exp 中的元素设置了属性：种类、值(对应元素的不同种类，值也有着不一样的含义)。

- 7.建立用于对应<变量, 变量结构体>的 Map, 寄存器的号码在变量结构体里面。这样在后序输入变量(char[])时, 可以方便的找到对应的变量结构体。
- 8.对于输出 Exp(运算相关的)的 LLVM IR, 使用栈来实现, 建立<Exp 元素结构体>的栈。
- 9.对于栈中的 Exp 的入栈顺序, 建立两个函数分别处理运算和正负号, 并打印相应的 LLVM IR 语句。
- 10.重构程序返回非零的方法。以前使用函数递归返回值来报错, 当时认为的好处是可以用返回值来确定错误发生的位置, 后来发现 printf 也可以做到同样的事情(确定错误位置), 而且函数递归返回值会导致程序复杂, 某些函数的返回值甚至有其他的作用(不能作为是否出错的判断)。所以改成了 try, catch 来检验程序是否出错。

## 提交后 debug

- 1.在看上去能够通过所有样例的情况下进行提交。但实际上并没有把打印的 LLVM IR 文法和样例输出严格比对(看上去是对的), 结果第一次提交 5/15。后来发现在 LLVM IR 打印时有错误(指 fprintf 里面的东西打错了)。在修改后变成 7/15。
- 2.发现写的时候没有注意常量相关的语义约束。增加常量不能改变后, 变成 8/15, 增加常量初始化不能有变量参与后, 变成 9/15。
- 3.发现 fprintf 里面的 store 输出有问题, 少打了一个%。此外, 发现 token 数组太小, 而输入数据中有超级长的变量名, 所以加大数组。修改后 14/15。
- 4.最后是正负号判断逻辑有点问题, 修改完 AC。

## 感想

这个 lab 是目前为止我遇到的最难的 lab。光是 part5 就差不多花了十几个小时的时间来写完, 当然有不小的原因是因为做 lab2 的时候不想着要迭代开发, 所以 lab2 的代码本身就相当于白实现了(在 lab3 几乎完全重构了)。

提前看了 lab5 发现 lab5 需要常量折叠, 那我 lab2 写的也不能算是完全用不上(不过这也就证明我的 lab3 题意又理解错了)。

Part6 实现起来比 Part5 容易一些, 但是调试起来感觉就要麻烦不少, 最后调试的办法还是盯着打印的 LLVM IR 语句一行一行往下手算。

因为别的课程也有压力, 最终从 lab2 到 lab3 花了 20 天的时间。只算 lab3 写代码和 debug 的时间大概也有至少 20~30 小时了。让人感觉十分困难, 写的时候必须时刻保持头脑清晰, 一旦逻辑出错就全完了。

此外, 我发现理论课的讲课进度相比实验, 实在太慢了。我做 lab3 的时候已经是 lab3 实验发出后的 2 周后了, 这时候理论课才刚讲到文法的综合属性、继承属性。或许如果能早上理论课, 可以少走不少弯路。