# 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，并输出。

### 提交后debug

1.发现连续的加减是不行的(比如说 return 1+1-1)。Debug后，发现还是语法分析的问题，所以进行了语法分析的修改后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，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)。

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的时候以及是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。