# 思路

学号：19373440 姓名：王雨飞

### 开始前的分析

1.难。

2.语法规则改变很大，决定在写代码的时候，先去把语法规则写好，调试好，再去考虑输出。

3.LLVM IR 的数组初始化用到了几个之前从来没有用过的东西，先好好看一下实验指导，学习一下。

4.中午吃饭偶遇同一个软工小组的大佬(说不定是1921高工同学以外第一个写完lab8的同学)。提问lab7需要注意的东西，他说要注意lab8的参数传递。好巧不巧的是，lab7的实验指导中给出5个使用 GEP 指令的输出实例。所以现在我要选择自己的程序中，GEP指令的输出方法。于是我先去看了一下lab8的函数参数传递。

5.lab8的样例中，调用函数中以数组作为参数的例子不太多， 注意到***可以将二维数组的一部分传到形参数组中，如定义了 int a[4][3]，可以将 a[1] 作为一个包含三个元素的一维数组传递给类型为 int[] 的形参。***又结合一下自己数据结构的存储习惯，打算使用lab7中实验指导实例输出的第4个方法进行输出(转成一维数组，还是图方便)。

6.挑战实验中的第二个是多维数组，如果可以的话(虽然我写lab7的时候，这个实验的实验指导还没有出来)，最好在数据结构方面能提前为这个挑战实验准备。

### 写代码

1.先花了一个下午来完成语法分析部分。这个lab在语法分析部分需要修改的地方比较多，最大的问题是修改其中的一个语法后，这个语法之前的好几个相关文法的判断也需要再进行修改。有时候会忘记修改某个函数，有时候会改错。所以花了很长的时间debug，在正确的样例能够return 0的时候开始下一部分的编写。

**完成局部变量数组**

1.因为我自己代码里的分类讨论很多，拿数组定义与初始化举例，我应该需要写4份功能相似但又不同的代码。这让人感觉非常多，写起来也很费劲，如果同时写的话很可能会乱。所以我打算先写局部变量数组，等**到能够成功实现局部变量数组的所有功能(初始化、调用、赋值)以后，再去做剩下的3种讨论。**

2.扩展VarItem结构体，新增加3个属性：int dimesion(维数，如果不是数组则为0)

Int d[10](表示每个维度的大小，如a[3][4]的数组，则d[0]=3,d[1]=4)。

Int arraySize(表示数组的总大小)

3.在VarDef中的非全局变量讨论中，再增加一个判断是否为数组变量的讨论。对于数组变量，计算并赋值上面新增的3个属性，并输出相应的LLVM IR语句。

4.因为是转成了一维数组来做，所以在输出LLVM IR语句的时候感觉方便了一些。在VarDef中使用了memset语句(在我的编译器中，没有显性赋值的数组元素都初始化为0)。

5.在初始化方面，初始化相关的指令都在Initval()中，我通过循环来计算Offset(当然，初始化的时候这个值只能是纯数字), 然后使用getelementptr指令定位修改的位置，然后通过Exp()来计算出存的值 (这个值可能是数字，也可能是寄存器)，存值。

6.赋值方面需要着重修改LVal()函数。因为我开始前的分析有写：希望能为挑战实验中的“多维数组”准备，所以在这里写的还是比较绕的，基本都是通过各种循环来递增维数，然后计算出Offset。我的Lval()用来返回寄存器的数字，这里我改了很久，报告是在AC了以后写的，我也记不清刚开始的时候我是怎么写的了，感觉一直在改。

**完成数组的其它情况**

1.在完成了局部变量数组以后，还有3种分类讨论的情况需要去写。令人烦恼的是，他们每一种都和局部变量数组有一些区别，所以每一个都需要写的小心翼翼。

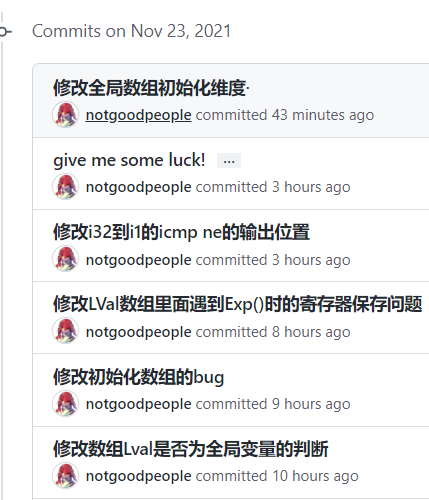
2.lab7写了两天多，github上提交了17次，我实在记不起来3种分类讨论的细节具体是哪里不同(写报告的时候刚AC，筋疲力尽了属于是)。我感觉是一边写，一边debug(用样例测试)，然后就到处都有点小问题，有问题就去改。所以这3种情况也不是同时完成的，而是像局部变量数组一样，一次做一个，等到一个测试得差不多了以后再写下一个。

3.简单提一下全局数组的初始化：我使用一个全为0的一维数组tempArr[1000]来初始化（当然，每次使用前都要memset）。这样其实是很方便的，我在tempArr[1000]这个数组中根据偏移量来修改对应的值，没修改的就默认为0。最后一并按照格式输出即可。

### Debug

1.bug出的非常多。篇幅有限，而且我也没有写代码时debug的记录(毕竟github有commit记录)，我这里只讨论提交后的debug(就是在通过了所有实验指导的样例以后)。

2.github上debug的记录：(数量众多，我这里随便挑几个感觉比较重要的写一写)

3.首先是，数组的偏移量我本来的命名是**index** (没文化了属于是)，在我的win10上使用devC++和VisualStudio2017都可以正常运行，但是在linux上使用g++不能编译，原因大概是g++里的**index** 是一个关键字。所以我就把所有的 index 改为了 offset (正确翻译)。

4. 测试样例里面并没有这样的例子 putint ( arr1 [ e[2][b] ] [ d ] ) ,其中b,d为int，e为二维数组。这也就意味着，对于 PrimaryExp()的调用，就可能需要LVal()返回数组的某个元素的寄存器的值，而这个我以前是没有考虑的，所以开始大改LVal。

5. 从上面的commit可以看出，我在LVal里面改了很久，写的时候确实脑子一直在转，一旦疲劳了就容易写错点什么(有时候并不是疲劳，就是情况想少了)。

6. i32到i1，我使用的是 icmp ne的方式(这个不是实验指导里给的zext，而是测试样例给出的)。但是很可惜，在lab4没出的错，在lab7出错了。

7. 全局变量初始化有错误，这个纯属是写麻了，全局变量的初始化和局部变量我用的是完全不同的两种方式。结果我不小心在某个分类讨论的全局变量初始化中，调用了局部变量的方法，所以错了。

### 感想

Lab7写起来真是太累了。代码量明显比之前的高出一个档次，而且前几次又有全局变量、又有常量，现在新增数组，自然也得按照以前的规则分类讨论，关键是每个分类讨论都有些不同，导致了写起来让人感到非常痛苦。

写完Lab7真是不容易。我想我在写Lab7的时候，即考虑了Lab8需要的参数传递，又考虑了挑战实验中的多维数组，希望以后写起来能轻松一些吧。