# Lab 1: Write a 3-address-code to C translator

小组成员：张崴，刘梦馨

1. **实验完成目标：**

1.正确翻译算术运算

2.正确翻译分支指令

3.正确翻译数据转移指令

1）进行地址计算

2）正确翻译move指令

3）正确翻译load指令

4）正确翻译store指令

4．正确翻译I/O指令

5.正确翻译函数调用和返回

1）参数正确入栈

2）栈正确清空

3）正确进行栈指针管理

6．正确分配全局和局部变量

1. **程序使用方法**

程序执行需要原工具包里的csc可执行文件，将lab1目录放入cs380c\_lab1目录下，执行目录下的compile.sh和run.sh，即可在标准输出获得翻译后的c源码。

1. **程序设计思想**
2. 从stdin中读取三地址源码存储至code3add数组，并区分enter和ret对源程序按函数进行分段。全局存放一个funcnum记录当前函数个数，用来给函数进行命名，函数名为f+funcnum，参数部分为空，等待对应函数ret时进行处理回填。
3. 建立辅助函数进行字符串的解析和处理。

str2int()，int2str()分别进行字符串和数值变量进行相互转换。

parsepar()对运算参数进行处理

parseari(),parselog()对算术运算和逻辑运算进行处理

parseload(),parsestore()对全局变量和数组的赋值和读取进行处理

parseenter(),parseret()处理函数入口和出口相关操作

parsebr()对跳转语句进行处理

1. 针对每个语句计算当前语句表达式的值记入mediate数组。算术表达式记录当前表达式。

对每个跳转分支进行类型判断，选择插入if，else或while并设置go数组插入“}”。对每个变量进行地址计算，区分变量类型是数组还是long型变量。

根据变量地址的符号和大小，判断参数是属于全局变量、局部变量还是函数参数。全局变量记入paramg数组，局部变量记入paraml数组，函数参数记入param数组。函数调用参数记入paramf数组，在翻译完call后清空该数组。

对move，load和store进行地址和参数计算，填入等号和表达式。

遇到ret对该段函数的mediate数组进行整理，选择需要输出表达式输出到finalcode数组，调整finalcode数组，插入“}”和局部变量。根据ret 的值计算出函数的参数个数，利用param数组的值填写函数参数。完成一段函数的翻译。

依此类推完成隔断函数的翻译。

1. dfc()函数输出程序必须调用的头文件和宏定义。输出全局变量数组中对全局变量进行定义的语句，输出finalcode数组中的表达式。
2. **程序现存问题**
3. 一开始没有注意到翻译语言子集中有结构体的部分，缺少翻译结构体部分的程序。