

编译原理实验四报告

- 姓名：李润泽
- 211275011@smail.nju.edu.cn

一、程序功能

实现了c--语言从中间代码到目标代码的生成。

主要定义了以下结构体来实现对变量表和寄存器的维护

```
typedef struct _register {
    boolean isFree;
    char* name;
} Register;

typedef struct _variable {
    int regNo;
    pOperand op;
    pVariable next;
} Variable;

typedef struct _registers {
    pRegister regList[REG_NUM];
    int lastChangedNo;
} Registers;

typedef struct _varList {
    pVariable head;
    pVariable cur;
} VariableList;

typedef struct _varTable {
    pVarList varListReg; // 寄存器中的变量表
    pVarList varListMem; // 内存中的变量表
    boolean inFunc;
    char* curFuncName;
} VarTable;
```

首先输出目标代码的固定内容，包括对.data节.global节以及write和read函数的定义，另外因为默认无重名变量，所以选择直接把所有的数组声明DEC作为全局变量使用.word声明，这样可以避免在栈中维护数组带来的繁琐。

然后遍历上一步得到的中间代码，逐条翻译为目标代码即可。

对于寄存器的替换采用了比较简单的替换方式，当有空闲时候直接占用，无空闲时候先优先选择最近没用过的立即数占用的寄存器替换，如果无立即数占用的寄存器则找最近没用过的一个临时变量占用的寄存器替换。

二、执行程序

通过make编译程序，然后./parse (input_file) (output_file) 将.cmm文件输入得到目标代码文件,也可以选择直接make test, 会将Test/test1.cmm翻译

