编译原理实验文档

|  |  |
| --- | --- |
| 姓名 | 寇瑞 |
| 学号 | 2015302555 |
| 班级 | 10011501 |

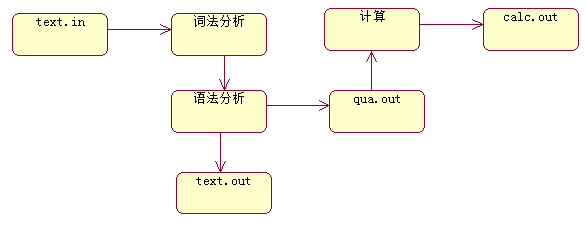
**1.需求分析**

**1.1需求概述**

软件的目的是设计一个简单编程语言的小型编译器，这个简单的编程语言支持的语法格式简单，主要有变量的声明、定义、初始化，变量的赋值和四则运算，布尔表达式的计算，支持if else语句和while语句，支持print语句，支持逗号分隔符，每个语句的结束以分号分割。本软件首先通过词法分析识别用户写下的程序代码的单词，转化为程序定义的一系列token，然后通过语法分析部分利用LR(1)自底向上的语法分析结合语义动作将token转换为四元式，同时将程序中用到的临时变量和定义的变量存放到变量表中，为四元式的生成提供变量表中的索引位置，四元式又分为了算术运算、布尔运算、跳转语句的类型，最后通过计算执行模块以产生的四元式和变量表为基础计算源程序的执行结果，并返回给用户。

**1.2功能需求**

此软件的IO接口全部通过读写文件的形式实现，用户将源程序代码写到text.in文件中，然后启动程序开始执行，程序将分析得到的四元式写到qua.out文件中，将具体的详细分析过程（包含规约中用到的产生式，变量表的状态，真链和假链等）写到text.out文件中，然后程序将qua.out输出的四元式执行计算得到源代码的执行结果写到calc.out文件中。



**1.3非功能需求**

运行平台：ubuntu 1604-64b

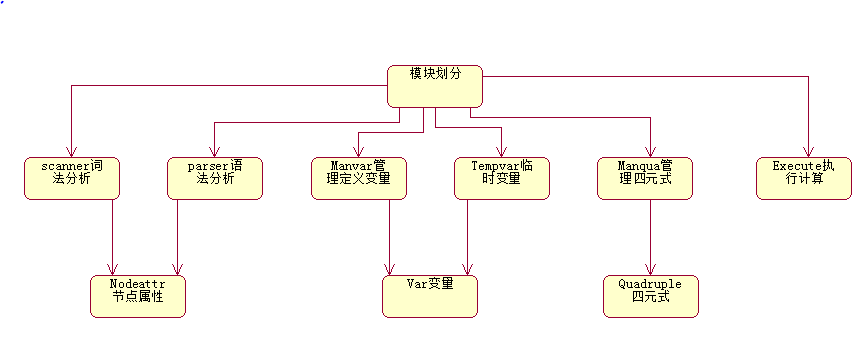
内存大小：1G及以上

编译器：g++ 4.9.2

flex:2.6.4 bison:2.7

**2.详细设计**

**2.1各模块/类及功能**



|  |  |  |
| --- | --- | --- |
| 模块/类名称 | 功能 | 主要数据/函数 |
| scanner.l文件 | 识别源程序单词，将文本转换为token,将字面常量保存到Nodeattr的litetext数据域中 | void unrecognized\_char(char c) |
| parser.y文件 | 定义文法用到的token，定义文法及相应的语义动作，指挥节点属性的设置，变量的存储和四元式的生成 | %token T\_Int T\_Float V\_Int V\_Float Identifier PRINT IF THEN ELSE WHILE DO BEG END Rop AND OR OPS |
| Nodeattr | 语法分析的节点属性和对属性的操作，包括Place,litetext,真假链，回填和合并真假链 | string litetext;  int loopstartplace;  int place;  string type;  int truechain;  int falsechain;  int chain;  void setlitetext(Nodeattr node,char \*text);  int mergechain(int p1,int p2);  void backpatch(int pc,int num); |
| Var | 变量的名称，类型，值以及变量的构造函数 | Var();  Var(string n,string t,int i,float f);  string name;  string type;  int iv;  float fv; |
| Quadruple | 四元式的操作名称，操作数的index值，操作结果的index值，以及四元式的构造函数 | Quadruple();  Quadruple(string on,int o1,int o2,int r);  //opname分为int和float和print  //int : +i -i \*i /i unmi =i  //float : +r -r \*r /r unmr =r  //跳转：< > j  //类型转换 : itr rti  //输出语句 : print  string opname;  //op1/2等于0表示为空  int op1;  int op2;  int res; |
| Manvar | 管理变量的增加，查找，索引到Var的映射，标识符到Var的映射 | int addvar(Var &v);  int name2index(string name);  Var& index2var(int index);  vector<Var> vartab;  map<string, int> varindex;  int curnum; |
| Manqua | 管理四元式的增加，查找，索引到Quadruple的映射，输出四元式到text.out和qua.out文件中 | int addqua(Quadruple q);  Quadruple & getqua(int index);  void outqua();  vector<Quadruple> quatab;  int curnum;  ofstream quafile; |
| Tempvar | 管理临时变量的生成，查找，索引到Var的映射 | int newtemp();  int addvar(Var v);  Var& index2var(int index);  vector<Var> vartab;  int curnum; |
| Execute | 以生成的四元式文件qua.out和变量表vartab为基础对源程序执行计算并输出结果到calc.out文件中 | Quadruple& idx2qua(int idx);  void calculate();  Var& idx2var(int idx);  ofstream exeout;  int curidx;  int quasize; |

**2.2模块之间详细工作流程**

（1）临时变量的添加：

在parser.y和scaner.l文件的声明部分都有一个#define YYSTYPE Nodeattr语句，用以指定语法分析对应的节点的数据类型为Nodeattr类型，Nodeattr类型定义了一些数据域string litetext; int loopstartplace; int place; string type; int truechain; falsechain,chain，这样在语法分析的语义动作中可以通过$$.place,$1.truechain,$n.litetext的方式使用、设置节点的属性。Nodeattr中定义的setlitext函数可以在词法分析阶段将有用的字面值以string的形式保存下来，例如：

在词法分析阶段的{INTGER}对应的操作：

{INTEGER} { Nodeattr::setlitetext(&yylval,yytext);return V\_Int; }

应用到语法分析阶段，可以利用litetext已知的值来定义一个临时变量，

| V\_Int

{

...

Var temp\_ivar("","int",atoi($1.litetext.c\_str()),atoi($1.litetext.c\_str()));

$$.place = Tempvar::addvar(temp\_ivar);

$$.type = "int";

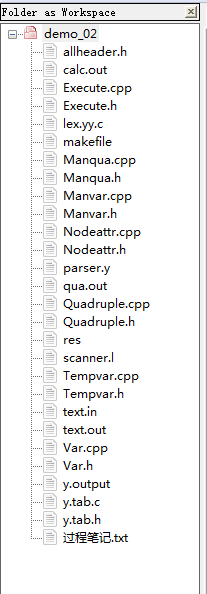
...

}

（2）变量的管理：

变量分为定义的变量和临时变量，为了在四元式中的op1,op2,res标号上进行区分，规定定义的变量的index为正数，临时变量的index为负数，如果op1,op2,res中有一个为0，表示此处的操作数为空。

**2.3工程目录结构：**



**2.4编译/运行方式：**

编译：提供一个makefile文件，最终生成的可执行程序为res

在demo\_02目录下，终端输入make命令即可编译

result : parser.y scanner.l Manqua.cpp Manqua.h \

Manvar.cpp Manvar.h Nodeattr.cpp Nodeattr.h \

Quadruple.h Quadruple.cpp \

Tempvar.cpp Tempvar.h Var.cpp Var.h allheader.h Execute.h

bison -vdty parser.y

flex scanner.l

g++ -o res y.tab.c lex.yy.c Manqua.cpp Manvar.cpp Nodeattr.cpp Quadruple.cpp Tempvar.cpp Var.cpp Execute.cpp

.PHONY : clean

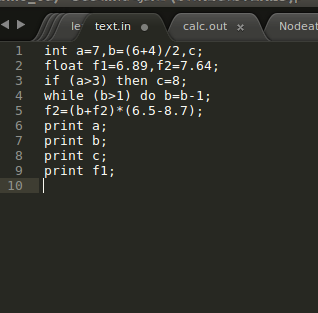
clean :

rm -f res lex.yy.c y.tab.c y.output

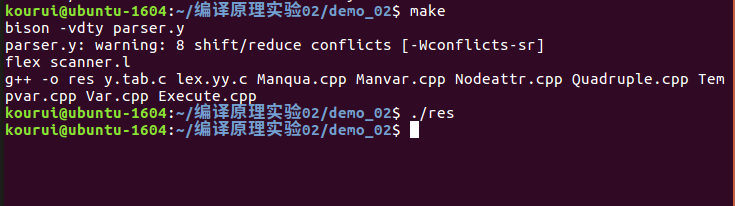
运行：首先在demo\_02目录下，新建一个text.in文件，写下源程序代码，然后在终端输入./res命令即可运行

**3.测试及维护**

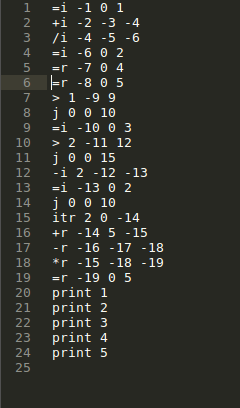
（1）新建一个text.in文件，输入一些程序语句保存。



（2）在终端进行编译，运行res程序



（3）查看生成的qua.out文件



（4）查看生成的calc.out文件，里面执行了源程序的print语句，输出了a,b,c,f1的值，经过计算验证是正确。

