**电子科技大学 信息与软件工程 学院**

**标 准 实 验 报 告**

**（实验）课程名称 编译技术**

**电子科技大学教务处制表**

**电 子 科 技 大 学**

**实 验 报 告**

**学生姓名：李艳超 学 号：2022090902007**

**指导教师：肖堃**

**实验地点： 信软303 实验时间：2023.10.26**

**一、实验室名称：信软 303**

**二、实验项目名称：递归下降语法分析**

**三、实验学时：4学时**

**四、实验目的、原理、内容及步骤：**

**目的：**通过本实验加深对编译技术中重点算法和编译技术的理解，提高学生的编程能力培养好的程序设计风格。了解和掌握递归下降分析法的基本原理，根据给出的文法能够完成递归下降程序的实现。

**原理：**递归下降分析器编译思想是简单的，从识别符号开始，在语法规则支配下进行语法分析，它逐个扫视源程序中的所有字符，根据文法和当前输入字符预测到下一个语法成份U时，便确定U为目标，并调用分析和识别U的子程序，在分析U的过程中，又有可能确立其它(或自身)子目标并调用相应子程序，如此继续下去。

**内容：**

1、学习所提供的“表达式文法”的递归下降处理

理解 lex.l、rdparser.c 的内容

在 vscode/Clion 中建立工程并调试运行

2、学习所提供的文法

与词法分析所提供的文法作比较

3、编写 rdgram 所提供文法的递归下降程序

(1).编写不生成“语法树”的递归下降程序 rdcheck.c

(2).将 rdcheck.c 改造为生成语法树的递归下降程序 rdparser.c

(3).改进 词法分析程序、showAst 函数、main 函数等，使递归下降程

序 rdparser 最终从命令行读取要分析的程序 test.c,分析后调用 showAst 打印该程序的结构。

**实验步骤：**

依据PPT中“实验任务”部分，我的实验流程大致如下：

1.rdlex.l、rdparser.c的代码分析，前者其实上次实验很熟悉，主要是后者代码难懂，据此文件明确了icoding实验的大致方向和要求，并尝试进行一定的修改

2.学习了sysy.y中的文法，这部分其实理解并不是很透彻，只是明白大概我要根据其中的内容来写

3.在网上搜索相关的资料，主要的参考是

[【Compile】递归下降语法分析(C++)\_语法分析程序c++-CSDN博客](https://blog.csdn.net/comscience/article/details/124600724)

[编译原理学习笔记（四）：语法分析 - 知乎 (zhihu.com)](https://zhuanlan.zhihu.com/p/331794391)

[【编译原理】【C语言】实验三：递归下降分析法-CSDN博客](https://blog.csdn.net/qq_41315788/article/details/122455432)、

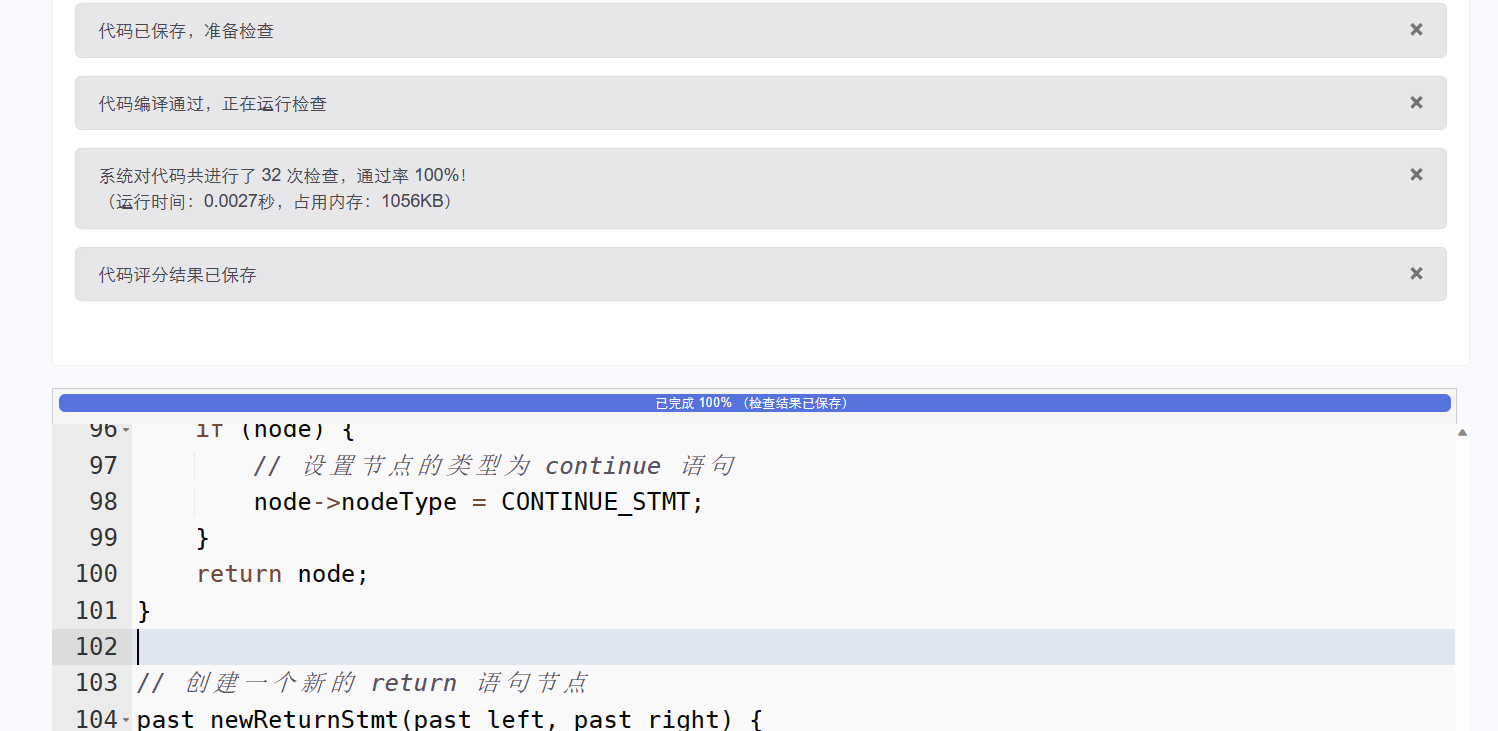
大致明确了思路

4.本地调试代码

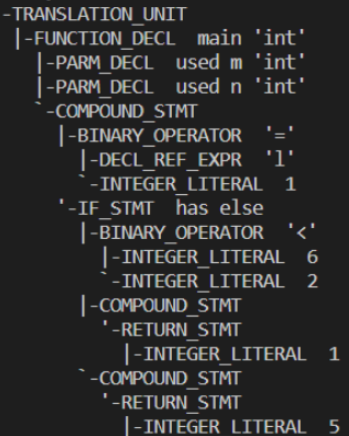
5.尝试icoding实验

**五、实验运行结果：**

个人Icoding通过



终端输入int main(int m, int n){l = 1; if(6 < 2){return 1;}else {return 5;} }显示结果如下



**六、实验结论与总结：**

代码的主体部分是创建各种结点和对应的解析各类表达式

具体来说，我们的创建结点包括：二元操作，声明引用，while、if、break、continue、return，而解析表达式是：加减法、乘除取模、一元、基本、与、或、相等、调参、关系、语句，后者执行递归下降语法分析，根据语法规则来构造AST

\*\*我们先介绍第一大部分：\*\*

二元操作：创建结点，设置结点类型、存储操作符，存储左右操作数

声明引用：创建结点，设置结点类型，存储名称，存储左右子结点

while：创建结点，设置类型，左是条件，右是循环体

if：创建结点，注意这里是if\_cond里面是条件，左是if部分，右是else部分

break：创建结点，设置类型是break结束

continue：创建结点，设置类型是continue结束

return：创建结点，设置类型，左右仍然设置，但一般情况下返回一个就是左

\*\*下面是解析表达式部分：\*\*

基本表达式：初始结点设置空，判断当前token，然后分别进入相应处理，这是根据文法，相对较难的是遇到标识符（处理左值）和括号表达式，标识符则获取标识符的字符串值，调用现成的函数解析数组下标，然后构造结点（接收刚才的标识符和数组下标），移进结束；括号表达式则移进，创建结点，判断是否有右括号，右就继续移进

一元表达式：初始结点为基本表达式，基本表达式空就循环，如果是标识符获取字符串值存储到s中，移进，如果是左括号，那这就是函数调用，移进，解析函数调用参数，创建结点返回；如果是加减反就记录符号，移进，获取一元表达式，构造二元操作表达式

加减表达式：当token是加减符时，记录运算符，移进，获取乘法表达式（得到结点表示数字），构造二元表达式

乘除表达式 相等表达式极为类似

关系表达式：一直循环，如果时关系运算符就构建结点并break，不是就返回左

或表达式：获取逻辑与表达式，如果是或就一直循环，记录运算符，移进，获取与运算符，构造二元操作

函数调用参数：使用链表来处理，获取加法表达式，定义头结点，循环：当前符号非逗号时break，移进获取加法表达式，当前的下一个新结点，指针移动到下一个继续循环

各种语句：直接根据当前的token选择进入哪一个（把上面的那一部分给包括进来）：赋值语句、空语句、代码块、while语句、if语句、break、continue、return

实现的递归下降分析程序能够完成对sysy文法中多个部分完成分析并生成AST，满足了icoding要求

个人通过这次实验温习了递归下降文法分析法，亲自实现过程需与书本感觉完全不同，考虑到了许多细节，做完感觉需要明显指出的是：递归下降分析法确实对含有公共左因子和左递归的文法词法分析效果很差。

此外，因为实验报告可能稍显单薄，为了补充和完善本次实验，我将此次相关的个人代码、笔记、总结内容上传GitHub，网站：[LiZaiyuan0619/Exps-of-Compilation (github.com)](https://github.com/LiZaiyuan0619/Exps-of-Compilation)

**报告评分：**

**指导教师签字：**