|  |
| --- |
| 编译原理  实验报告  （第四周）  **指导教师：杨冬**  **年 级：2019级**  **班 级：5班**  **小组编号： 2**  **组长学号姓名：55190521 季圣鹏**  **组员学号姓名：55190516 孟令震**  **组员学号姓名：55190512 贾鹏程**  **2022年 6 月 2 日**  **软件学院** |

|  |  |  |  |
| --- | --- | --- | --- |
| **实验目的** | | | |
| 更好地理解编译原理。  直接实现核心程序词法分析器和语法分析器。 | | | |
| **实验内容** | | | |
| 1. 完成实验的基本任务，能够顺利运行  2. 生成语法树，并能够动态展示生成过程  3. 能够处理部分错误，能够完善补充部分SNL语言部分。 | | | |
| **小组成员任务占比及组内评分** | | | |
| 学号\_姓名 | 占比 | 评分 | 评分理由 |
| 55190521季圣鹏 | 85% | 10 | 认真完成 |
| 55190516孟令震 | 10% | 9 | 认真完成 |
| 55190512 贾鹏程 | 5% | 8 | 认真完成 |
| **程序设计思想** | | | |
| 字符串匹配。动态数组。双指针。递归。树型结构。深度优先遍历。  递归下降法。自顶向下文法。Predict集合。 | | | |
| **实验步骤** | | | |
| 1. 读入程序，转化为字符串存储 2. 利用字符串匹配和双指针形成token序列 3. 定义语法树节点，构思语法分析方法 4. 利用递归下降法和Pridect集进行动态生成语法树。 | | | |
| **数据结构及算法设计** | | | |
| 数据结构主要是语法分析树部分的树结点，然后token那边是动态数组  //定义token序列的结构  struct Token {  string value1;  string value2;  };  //节点的具体类型  struct Specificnode {  string dec="";  string stmt="";  string exp="";  };  //定义语法分析树的节点  struct Treenode {  Treenode\* child[10];  Treenode\* sibling;  int idchild=0;//儿子的个数  string nodekind;//节点类型  Specificnode specificnode;//详细类型  int idnum=0;//标识符个数  string value;//节点的值（有的节点只有一个标识符。也用value了）  string id[10];//部分标识符的值  };  算法主要是两个，双指针和递归 | | | |
| **实验结果截图** | | | |
|  | | | |
| **实验结论** | | | |
| 项目Github地址：https://github.com/jishengpeng/Design-compiler  完成编译过程中的词法分析器和语法分析器。 | | | |