

编译原理—实验报告

- 李润泽 211275011
- 211275011@smail.nju.edu.cn

程序功能

- 1、实现了对c--的词法，语法分析，并给出分析成功的语法分析树。
- 2、选做2.2：识别了指数形式的浮点数

使用flex进行词法分析；bison进行语法分析

语法树的数据结构如下：

```
typedef struct synaNode{
    char* name; //e.x.:ID,Program,INT,FLOAT...
    int _type; //is token or not[1 or 0]
    char* value; // yylval for token and "Value" for others
    unsigned lineno; // the position of object
    struct synaNode* children[9]; //children Node,and in this case,the most children nodes of one parent node is 7
}node;
```

对于叶子节点(也就是词法单元)以及内部节点分开创建新节点：

```
static inline node* CreateLeaveNode(unsigned lineno,char* name,int _type,char* value){
    node* newNode = (node*)malloc(sizeof(node));
    newNode->lineno = lineno;
    newNode->name = copyString(name);
    newNode->_type = _type;
    newNode->value = copyString(value);

    for (int i = 0; i < 9; i++) {
        newNode->children[i] = NULL;
    }

    return newNode;
}

static inline node* CreateInsideNode(unsigned lineno,char* name, int _type,char* value,int n_child,...){
    node* InitialNode = CreateLeaveNode(lineno,name,_type,value);
    va_list args;
    va_start(args, n_child);
    for (int i = 0; i < n_child; i++){
        node* tmp = va_arg(args, node*);
        InitialNode->children[i] = tmp;
    }
    va_end(args);
    return InitialNode;
}
```

copyString是自定义的函数。其中内部节点的创建只比叶子节点的创建多了添加子节点的步骤，由于子节点的个数不定，因此使用了变长参数，传递的是子节点的地址。如下所示：

```
ExtDef : Specifier ExtDeclList SEMI {$$ = CreateInsideNode(@$.first_line,"ExtDef",Not_a_Token,"__Value__",3,$1,$2,$3);}
```

lexical.l和syntax.y 的编写主要参照了实验手册的说明。

程序运行

使用提供的Makefile文件编译运行即可

