

# ISE lab1

221900073 孙佳琪

## 程序所实现的功能

通过标准输出打印程序的运行结果。对于那些包含词法或者语法错误的输入文件，只要输出相关的词法或语法有误的信息即可。在这种情况下，注意不要输出任何与语法树有关的内容。要求输出的信息包括错误类型、出错的行号以及说明文字，其格式为：Error type [错误类型] at Line [行号]: [说明文字]。

因为我需要做的选做是1.1，所以该程序能识别八进制和十六进制数为int类id的值。

## 如何被编译

1. `bison -d syntax.y` 编译.y文件
2. `flex lexical.y` 编译.l文件
3. `gcc main.c syntax.tab.c -lfl -ly -o parser` 生成parser分析程序

## 个性化的内容

在词法分析部分，新建一个node结构体来表示语法树中的节点

```
typedef struct Node{
    char* name;
    MyType type;
    int line;
    union {
        unsigned type_int;
        float type_float;
        char type_str[40];
    } val;
    struct Node* child;
    struct Node* sibling;
} Node;
```

child表示子节点，sibling表示邻居节点。

并根据每个语法节点是否是终结符或者null而采用不同的函数进行建立节点的处理（TerNode，NterNode，NullNode方法建立不同类型的节点）。

最后在print\_node函数中对整个语法树进行前序遍历的输出。

而在报错方面，在产生式中加入error来辅助判断错误并自动调用yyerrok函数，并新建一个line\_error数组，设置其每个元素的初始值都为0，只要在某一遇到错误就将数组中对应行的元素设为1，在打印每次的报错信息之前先判断line\_error[yylineno]是否为0，为0则打印报错信息后再将line\_error[yylineno]设为1。