Lab1 实验报告

191870147 屈力

实现功能

完成了所有内容（必做+选做）

1）通过GNU Flex和GNU Bison分别实现词法分析和语法分析的基本功能，可以识别C--词法和语法定义的词和语法。

2）在lexical.l文件中添加了对于八进制、十六进制、指数形式浮点数的识别。除此之外，我还用正则表达式表达了这些词的可能的错误形式，当对应的表达出现错误时，会报词法错误。

3）识别"//"和"/.../"形式的注释。当词法分析器遇到"//"或"/*"时，会不断过滤后面的字符，直到遇到行尾或"*/"。

4）不完全的错误恢复，当遇到一些语法错误时，语法分析器在报错后不会终止，而是从错误中恢复，继续语法分析。该功能通过在一些产生式中加入特殊字符error可能出现的位置来实现。

5）实现了语法树的构造。词法层面，在识别出一个合法的词后，词法分析器对应的行为是创建一个树节点（位于叶子上），供上层（语法分析器）直接使用。语法层面，每识别出一个语法，会为产生式左侧的非终结符创建树节点，然后建立其与右侧终结/非终结符的联系（父节点指向第一个子节点，子节点之间通过链表连接），逐渐构造一棵语法树。在这里我把树的构建放到了新的SyntaxTree.h和SyntaxTree.c文件中，将建树和语法分析的功能分开，增强了代码的可读性和debug的方便性。

6）实现了报错信息输出。我重写了yyerror，当发现一个语法错误时，会调用新的yyerror函数，该函数会打印错误所在的详细位置（行数通过Flex自带的功能在程序内部自动更新）。

编译方法

在当前目录下直接make（我删除了Makefile的-ly选项，因为这个选项一般没有影响，但是加上后我这里(Ubuntu 20.04)无法编译）。
在终端输入./parser ../Test/filename可以测试某一文件，或修改Makefile中的test目标然后直接输入make test以测试某一文件。