实验一 词法分析和语法分析

姓名：张子谦　　学号：151220166

院系：计算机科学与技术系　联系方式：zhangziqian1011@126.com

一、完成的功能

1. 识别词法错误(A 型)，打印行号与错误信息；

2. 识别语法错误(B 型)，打印行号与错误信息；

3. 按要求在没有错误时打印语法树；

4. 完成所有额外要求：

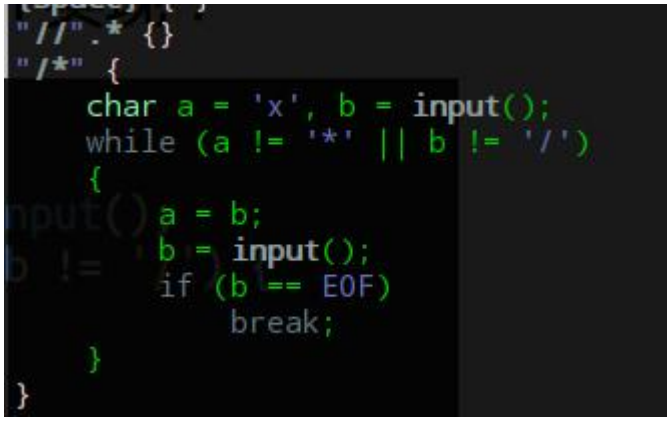
   （1）识别 8 进制 16 进制整型树并在其错误时报错；

   （2）识别指数形式的浮点数并在其错误时报错；

   （3）识别两种形式的注释；

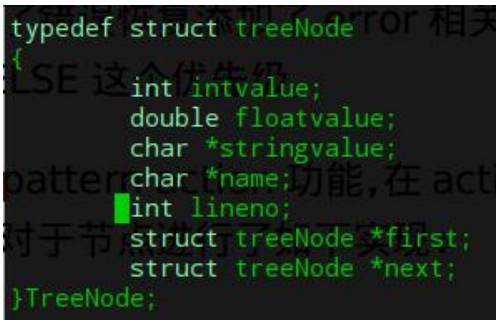
二、实现方法

1. 使用 flex 并为每一类词法单元书写了正则表达式来识别它们。其中三种额外要求都在这里就完成了，尽管注释是属于语法分析的内容。8 进制、16 进制、指数型浮点数包含了检错报错的功能，而“//”通过以下 pattern 和 action 实现：



2. 使用 bison，按照附录上的语法规则与相应的优先级与结合性书写了语法公式。其中按照讲义上的要求为了错误恢复添加了 error 相关的语法规则，为减少冲突增加了 LOWER_THAN_ELSE 这个优先级。

3. 在 flex 和 bison 提供的 pattern action 功能，在 action 中进行语法树的构建，考虑到语法树是多叉树，对于节点进行了如下实现：



这是经典的多叉树实现方式通过记录第一个子节点和下一个兄弟节点将树的每一层变成了链表，方便了操作。本来准备将 YYSTYPE 用 union 来取值的，但是考虑到语法树构建的便捷性，直接将 YYSTYPE 设置为了 TreeNode*型。在.y 文件里的 action 模块中考虑到产生式的右部有多个语法单元用了 va_list 来简化操作。

三、运行方式

1. 在 Code 文件夹运行 make 可以在项目目录中产生可执行文件 parser，./parser testfile 可以对 testfile 进行词法语法分析并在屏幕上输出结果。

2. make test 可以测试 testcase/文件夹下面的所有.cmm 文件并输出到对应的.out 文件中。

3. make clean 清除所有 make 生成的文件。

四、实验总结

1. bison 和 flex 极大地减少了词法分析和语法分析的工作量，lex.yy.c 和 syntax.tab.c 中都有 2000 行的代码，而.l 和.y 文件中只有 100 行左右，因此这种工具软件为词法语法分析工作带来了极大的方便。

2. 本次实验只用到了 bison 和 flex 的基本功能，一些高级功能还没用上，如果能对这些功能加以熟悉，词法语法分析工作将变得更简单。