编译原理实验一：C--语言的词法分析与语法分析

软件工程 范兆基 20331011 819402765@qq.com

一、词法分析(lexical.l)

1. 实现了终结符号、注释、空白字符、部分错误八/十六进制数、部分错误浮点数、部分错误注释的匹配
2. 匹配到注释或空白字符跳过不分析
3. 匹配到终结字符，将匹配结果返回值bison：包括错误八/十六进制数、错误浮点数

二、语法信息(syntax.y)

1. 声明标号：
   a. 各符号类型：type_node——指向语法树节点的指针
   b. 部分符号的结合性：解决大部分二义性
2. 根据附录A给出的产生式进行规则书写，且对if-else的移入-规约冲突进行处理

三、语法树(tree.h、tree.c)

1. 构建语法树节点类型Node，**限定了子节点的个数最大为10。**
2. 在进行词法分析时建立叶节点，在进行语法分析时建立内部节点，并连接父子节点
3. 实现语法树的递归遍历与栈遍历

四、错误处理

直至作业提交之时，我仍未能弄明白如何准确地添加包含error的产生式，其他产生式中的规约会使error匹配到意想之外的地方。所以本作业未能很好进行错误处理。

1. 对于error产生式所包含的错误：
   a. Bison自动调用yyerror()进行错误提示
   b. 我又调用自己写的报错函数my_yyerror()进一步提示是在进行哪个非终结符的规约时发生错误
2. 对于未被error产生式包含的错误：仅由Bison调用yyerror()进行错误提示，然后**整个分析程序就会停止。**
3. 对于某些特定的错误：不完整注释、错误八/十六进制数、错误浮点数，我在词法分析中将其匹配出来并进行提示。对于其中的错误八/十六进制数、错误浮点数，我为它们添加了对应的终结符与修改了相应产生式，不会导致分析停止。

五、代码编译与运行

1. 代码编译：进入Code目录下执行make命令，会产生相应可执行文件parser
2. 代码测试：进入Code目录下执行make test命令，会自动测试Test目录下的所有文件