

编译原理实验一：词法分析与语法分析

——杨嘉兴+21311054+2807593076@qq.com

一、词法分析(`lexical.l`)

1. 实现了 C 语法中未定义的字符以及任何不符合 C 词法单元定义的字符。实现了终结字符、注释、空白字符、部分的错误八/十六进制数、部分的错误浮点数、部分的错误注释的匹配。实现了识别八进制数、十六进制和浮点数的错误识别判断。
2. 实现了注释或空白字符跳过不进行分析。
3. 当匹配到终结字符时，将匹配结果返回给 `bison`，即返回到语法分析 (`syntax.y`)

二、语法信息(`syntax.y`)

1. 声明终结符与非终结符：
2. 声明了部分符号的优先级与结合性，解决了大部分的二义性。
3. 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` 与 `Tree.c`)

1. 构建语法树节点类型 `Node`。
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. 实现语法树的遍历。

四、代码编译与运行

1. 代码编译：进入 `Code` 目录下执行 `make` 命令，会在该目录下产生相应可执行文件 `parser`。
2. 代码测试：编译完成得到 `parser` 文件后，通过 `./parser ../test` 的形式进行相对应的测试文件进行测试。