**实验报告**

-- 编译原理 实验一

姓名: 梁宇方 学号: 171860695

邮箱: leungjyufong2019@outlook.com

1. 我的程序实现了所有的必做功能和选做功能
2. 必做内容
3. 检查C词法中没有定义过的字符: 在lexical.l文件中使用正则表达式和响应函数来检查出”定义过”的TOKEN; 剩余的字符就是没有定义过的字符, 使用通配符来匹配, 相应的响应函数调用yyerror报错.
4. 检查语法错误: 在syntax.y文件中使用Bison的语法来重写C语法; 最后能规约到起始符号Program的情况为没有语法错误, 反之则有语法错误, 程序会自动调用yyerror进行报错.
5. 打印语法树: 对于没有语法错误的程序, 自底而上构建一颗抽象语法树, 然后从根结点开始深度优先遍历. 在词法分析器中构造叶子结点, 在语法分析器中构造中间结点.
6. 选做内容
7. 识别八进制和十六进制数: 修改lexical.l文件中对INT的正则定义, 使之可以包含八进制数和十六进制数的情况, 即

dec [1-9][0-9]\*|0

oct 0[0-7]\*

hex 0[xX][0-9a-fA-F]\*

INT {dec}|{oct}|{hex}

1. 浮点数的科学计数法表示: 同上, 前导可以是浮点数或整数

flt [0-9]+\.[0-9]\*|\.[0-9]+

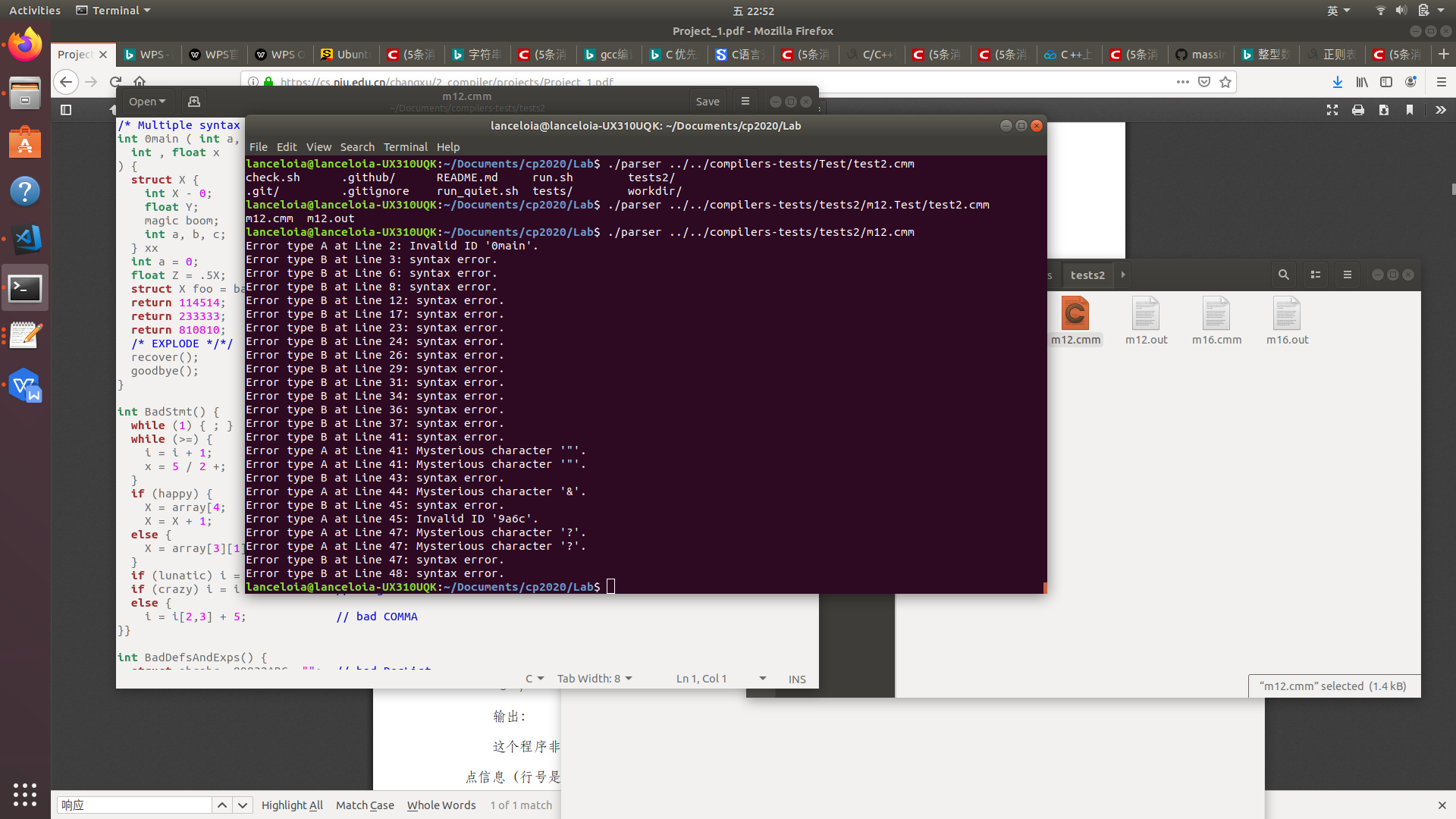
FLOAT {flt}|{flt}[eE][-\+]?[0-9]+|{dec}[eE][-\+]?[0-9]+

1. 行注释和块注释: 在词法分析器处使用正则表达式识别这些注释, 将其响应函数设置为空函数, 这部分内容不会反馈给语法分析器, 因此不会影响原有结构

LINECOMMENT \/\/.\*

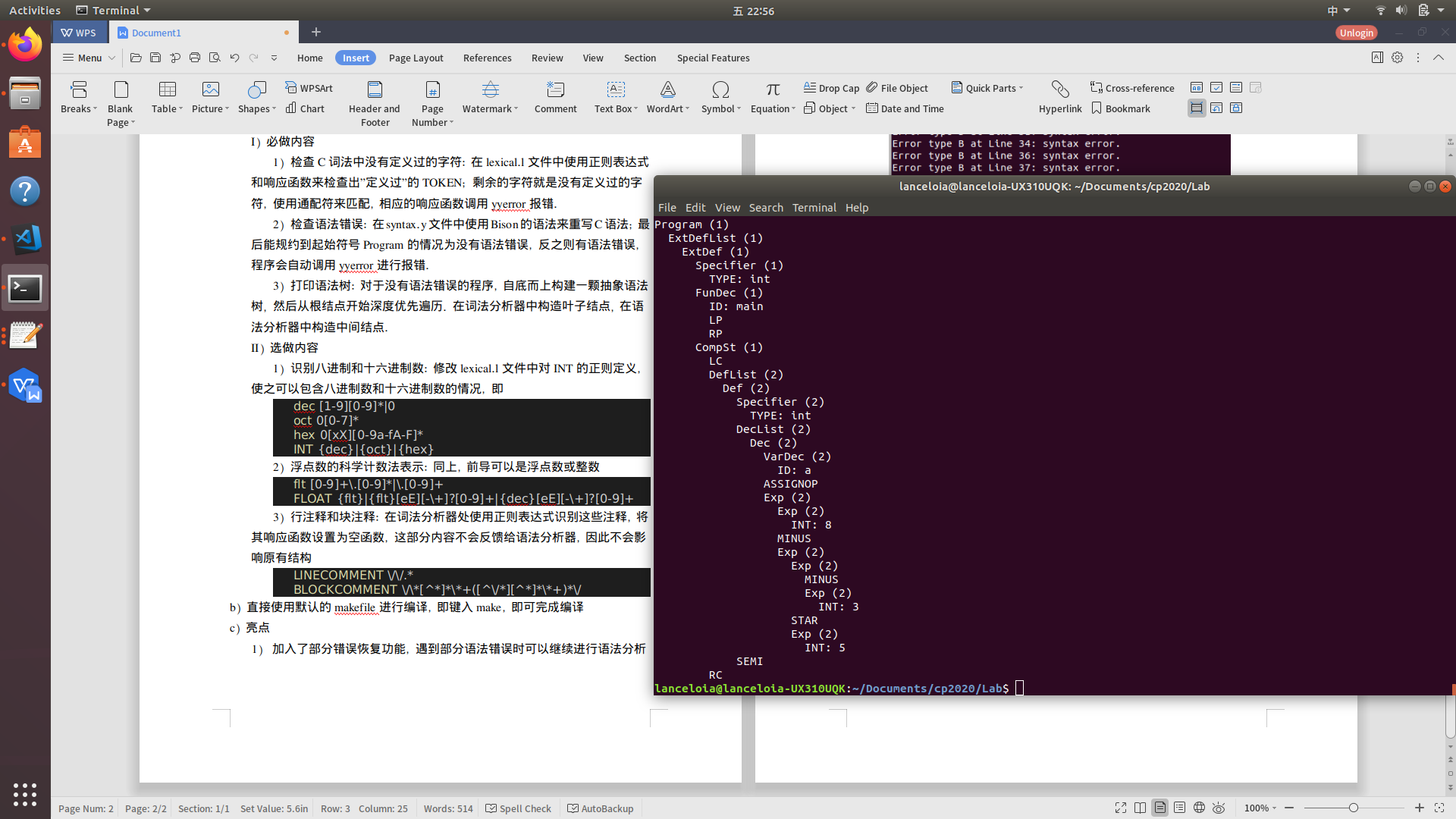
BLOCKCOMMENT \/\\*[^\*]\*\\*+([^\/\*][^\*]\*\\*+)\*\/

1. 直接使用默认的makefile进行编译, 即键入make, 即可完成编译
2. 亮点
3. 加入了部分错误恢复功能, 遇到部分语法错误时可以继续进行语法分析



1. 区分了减号和负号的优先级和结合性, 并能构建出正确的语法树

对于表达式8--3\*5, 给出了语法树(片段)为



对于表达式----2-3, 给出了语法树(片段)为

