

# 编译原理实验一

姓名：殷天润 学号：171240565 邮箱：[171240565@smail.nju.edu.cn](mailto:171240565@smail.nju.edu.cn) 班级：基础班

## 项目环境：

1. GNU LINUX Release: Ubuntu 18.04.3 LTS,5.3.0-40-generic
2. GCC version:7.5.0
3. GNU Flex version:2.6.4
4. GNU Bison version: 3.0.4

## 实现功能

1.完成了必做内容+所有的选做内容

- 通过flex实现了词法结构的分析
- 在flex中通过添加正则表达式完成了词法错误的查找（包括浮点数，十六进制，八进制数，整数错误），主要是通过预测可能的错误实现然后书写相应的正则表达式实现的；在发现相应错误的时候仍然会返回相应的Token,比如0x9g会返回INT,并且我会把相应的值设置成0; 相应的错误会的正确显示；
- 通过flex实现了注释的忽略，主要是通过input()以及while语句完成的；在/\*模式中我也专门判断了eof防止/\*\*\*\*\*直到文件末尾产生的死循环；注释我选择直接忽略，不会返回Token;
- 通过bison完成了语法分析的部分；我参考了[bison的中文手册](#)，通过\$prec,实现了MINUS(负号)的右结合优先级；
- 我完成了语法部分的一部分错误恢复，修改了yyerror，使得能够按照要求报错；
- 通过实现一个node多叉树结构体，我在flex里面构建终结符节点，在bison相应的位置构建非终结符节点，并且将终结符节点连接到相应的非终结符节点上，实现了多叉树；在bison中因为获得的变量数目是不确定的，我查询了RUNOOB使用了va\_list类型；
- 为了能够让全都是注释的文件正确显示Program(eof行号)这个需求，我在flex文件中添加了全局flag:emptyflag,当处理到非注释的词法时会变成1，初始为0；我设置了一个全局的变量emptystart,在bison文件中的ExtDefList-->ε处更新为yylineno;当全是注释的时候用emptystart更新Program节点的行数值；
- 我完成了条件编译的debug辅助函数，通过DEBUGNOW,DEBUGBISONNOW两个宏定义控制显示一些信息；

## 编译方法

1. 使用make 编译，make test测试相应test文件夹的对应文件，或者./parser test.cmm 测试
2. 使用

```
flex lexical.l
bison -d syntax.y
gcc main.c syntax.tab.c -lfl -ly -o parser;编译，通过./parser test.cmm测试相应的测试集合；
```

## 反馈

1. 我觉得讲义讲bison 语法单元的位置的时候应该额外说明一下 %locations应该放在.y文件，否则会报错，相应的[stackoverflow回答](#)；我觉得错误恢复这部分内容在阅读讲义的时候有些疑惑(当然相应的手册以及flex and bison这本书也写的不是很具体)；如果相应的Appendix部分能够给一个去水印的版本方便复制粘贴(或者给一个文本文件之类的)，能够有效减少手打笔误的挫败感；

2. 我使用了计算机拔尖班共同维护的一个[测试库](#)进行了测试，感谢我的同学们；