

# 编译原理第二次实验—语法分析

任课老师：李诚  
助教：邵新洋，白有辉

# 实验流程

1 Flex完成词法分析

3 生成LLVM IR



2 **Bison完成语法分析**

4 生成RISC-V代码

## 第二次实验要求



- 使用Bison工具辅助实现CMinus的语法分析器
- Bison是一款通用的语法分析器，其按照预定义的文法规则，对来自词法分析器的token流进行分析，并执行用户自定义的动作
- 关于Bison的使用入门:
  - ✓ [http://210.45.114.30/staff/compiler\\_cminus/blob/master/lab2\\_syntax\\_analyzer/Bison\\_Tutorial.md](http://210.45.114.30/staff/compiler_cminus/blob/master/lab2_syntax_analyzer/Bison_Tutorial.md)
- 有兴趣的同学也可以自行在网上查找额外的Bison资料进行学习

## 第二次实验步骤



- 同步第二次实验到本地仓库
  - ✓ [http://210.45.114.30/staff/compiler\\_cminus/tree/master/lab2\\_syntax\\_analyzer](http://210.45.114.30/staff/compiler_cminus/tree/master/lab2_syntax_analyzer)
- Cmake管理项目，按照教程进行编译
- CMinus产生式已给出，按照产生式编写bison规则到.y文件
- 按照给出的SyntaxTree结构填写bison动作，构造语法树
  - ✓  $E \rightarrow E + T$
  - ✓  $E : E \{node1\} \text{ ADD } \{node(+)\} T \{node2\}$
  - ✓ 将三个node添加到父节点
- 输出接口已实现，并调用输出到文件

## 第二次实验难点（加分项！）



- 参考入门手册，实现Flex和Bison的共享数据结构
- 过滤词法分析中无用的词法符号
  - ✓ 注释，换行，空白等
- 二义性文法
  - ✓ 移进归约冲突
  - ✓ 可开启Bison调试定位冲突
  - ✓ 设计解决方案
- 语法分析过程可视化

完整解决难点，有一定加分。可贡献解决思路，不能分享代码

## 第二次实验截止日期



- DDL: 2019年10月14日23点59分
- 补交:
  - ✓ 1. 必须在DDL前有提交 ( 特别简单的commit无效 )
  - ✓ 2. 不能超过一周 ( 2019年10月21日23点59分 )
- 补交会扣除一定分数

祝大家快乐实验，学有所获

