

# 编译PJ

COMP130014.01

# 简介

- Project分三部分，评分分别占20%、50%、30%，设置三个DDL
- 每组由四个人构成（成员名单请发至负责PJ的TA）
- Project介绍及demo：  
[https://github.com/linchuming/Compiler\\_Project](https://github.com/linchuming/Compiler_Project)
- TA联系方式：
  - 林楚铭 [17210240037@fudan.edu.cn](mailto:17210240037@fudan.edu.cn) （负责project）
  - 马晨曦 [17210240039@fudan.edu.cn](mailto:17210240039@fudan.edu.cn) （负责课程作业）

# 实验环境

- OS: linux, 推荐Ubuntu
- 需要gcc/g++编译器, 版本随意
- 另需要两个软件: flex 与 bison
  - `sudo apt-get install flex`
  - `sudo apt-get install bison`
- Mac 土豪自行安装flex与bison
- 总体来说, 就是C/C++/C++11配合flex与bison两个工具

# 处理语言简介

- 所有的测试样例为PCAT（简化版的Pascal语言）

```
(* test01:      *)  
(* test var decls. *)  
(*              *)  
PROGRAM IS  
  VAR i, j : INTEGER := 1;  
  VAR x : REAL := 2.0;  
  VAR y : REAL := 3.0;  
BEGIN  
  WRITE ("i = ", i, ", j = ", j);  
  WRITE ("x = ", x, ", y = ", y);  
END;
```

# Project 1 （20%）

- 使用flex，对于给定的PCAT语言的20个样例做词法分析，打印出所有的tokens
- 完成基本功能（10%）
- 输出每一个token的起始行号与列号（5%）
- 对于一些基本的错误，如整型溢出之类，能提供报错信息（5%）
- 提交项目代码与项目文档pdf（实现细节说明与成员分工）
- DDL：2017年10月23日00:00之前

# Project 1 (20%)

```
cmlin@cmlin-GL552JX: ~/PCAT-Compiler/lexer
→ lexer git:(master) ./lexer tests/test01.pcat
4, 1      PROGRAM
4, 9      IS
5, 5      VAR
5, 9      identifier(i)
5, 10     ,
5, 12     identifier(j)
5, 14     :
5, 16     identifier(INTEGER)
5, 24     :=
5, 27     integer(1)
5, 28     ;
6, 5      VAR
6, 9      identifier(x)
6, 11     :
6, 13     identifier(REAL)
6, 18     :=
6, 21     real(2.0)
6, 24     ;
7, 5      VAR
7, 9      identifier(y)
7, 11     :
7, 13     identifier(REAL)
7, 18     :=
```

输出格式可以自定义，无需按照图片样例

## Project 2 (50%)

- 结合project 1 完成的内容，使用flex & bison 完成对PCAT语言的语法树的建立，并将语法树打印出来
- 完成基本功能（40%）
- 具有语法报错功能并提示错误位置（10%）
- 该项目需展示给TA，先完成可先展示，时间与TA预约即可
- 提交项目代码与项目文档pdf（实现细节说明与成员分工）
- DDL: 2017年12月4日00:00之前

# Project 2 (50%)

```
cmlin@cmlin-GL552JX: ~/PCAT-Compiler/parser
→ PCAT-Compiler git:(master) ls
documents lexer LICENSE parser README.md
→ PCAT-Compiler git:(master) cd parser
→ parser git:(master) ls
ast.c ast.o main.c parser pcat.lex pcat.output pcat.y run_test.rb
ast.h gc-7.2 Makefile pcat.c pcat.o pcat.tab.h pcat.yy.c test
→ parser git:(master) ./parser test/test01.pcat

(program
  (body
    (declaration_block
      (var_decl
        (id_list_node i j)
        (type INTEGER) 1)
      (var_decl
        (id_list_node x)
        (type INTEGER) 2)
      (var_decl
        (id_list_node y)
        (type_empty) 3))
    (statement_block
      (write_state "i = " i ", j = " j)
      (write_state "x = " x ", y = " y))))%)
→ parser git:(master)
```

输出的树形结构可自定义，能看出是树形结构即可



# Project 3 (30%)

- 自由选题，根据选题的完成度与难度给分
- 选题1：调研其他类似于flex&bison的工具，分析比较其实现原理以及优点与缺点，提交一份pdf报告
- 选题2：结合project 2建立的语法树，在语法树的基础上实现进一步的功能，例如模拟代码运行，或翻译成机器代码等等（可只实现一部分语法的模拟或翻译），最后提交代码以及pdf文档
- 如果你有更好的选题想实现，可与TA联系
- DDL：考试前提交至TA邮箱

# 提交方式

- 如果文件太大，请先上传至百度云或复旦云或其他云，再将云存储地址发送至TA邮箱；文件小则直接打包发送给TA。
- TA邮箱： [17210240037@fudan.edu.cn](mailto:17210240037@fudan.edu.cn)
- 若对PJ有任何疑问，可与TA联系或提前预约TA线下交流

# 参考资料

- 项目说明Github: [https://github.com/linchuming/Compiler\\_Project](https://github.com/linchuming/Compiler_Project)
- 项目demo:  
[https://github.com/linchuming/Compiler\\_Project/tree/master/demo](https://github.com/linchuming/Compiler_Project/tree/master/demo)
- Flex and Bison: [http://aquamentus.com/flex\\_bison.html](http://aquamentus.com/flex_bison.html)
- Flex manual:  
[http://ranger.uta.edu/~fegaras/cse5317/flex/flex\\_toc.html](http://ranger.uta.edu/~fegaras/cse5317/flex/flex_toc.html)
- Bison manual:  
[http://ranger.uta.edu/~fegaras/cse5317/bison/bison\\_toc.html](http://ranger.uta.edu/~fegaras/cse5317/bison/bison_toc.html)

# Flex



- 通过用户定义的正则表达式，生成对应功能的C代码
- 使用例子： `flex -o lex.c test.lex`
- 规则：

定义段(definitions)

%%

规则段(rules)

%%

用户代码段(user code)

祝大家顺利毕业！

