

# 编译Project 1

COMP130014.01

2019.09

# 简介

- Project分成两部分，评分分别占40%与60%
- 每组不超过4个人
- 小组成员名单发送至负责PJ的TA
- Project介绍及demo，demo基础上补充完成相应功能即可：  
<https://github.com/WangJY06/CompilePJ>
- TA联系方式：
  - 王峻逸 [18110240005@fudan.edu.cn](mailto:18110240005@fudan.edu.cn) (负责Project)
  - 林青 [18110240028@fudan.edu.cn](mailto:18110240028@fudan.edu.cn) (负责课程作业)

# 实验环境

- OS: linux, 推荐Ubuntu
- 依赖: gcc/g++ (版本不限), flex, bison
- flex与bison安装(以Ubuntu为例):
  - `sudo apt-get install flex`
  - `sudo apt-get install bison`
- 实验环境也可在MAC OS以及WINDOWS下配置, 具体配置请自行搜索
- 简单来说, 就是C/C++配合flex与bison两个工具

# 实验目的

- 通过flex与bison两种工具，分析目标PCAT语言，并生成目标语言的语法树
- PCAT语言可看作一种简化版的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介绍及demo:

<https://github.com/WangJY06/CompilePJ>

## Project 1 (40%)

- 使用flex, 对于给定的PCAT语言的12个样例做词法分析, 打印出所有的tokens
  - 完成基本功能, 计算tokens的总数 (10%)
  - 输出每一个token的起始行号与列号 (5%)
  - 对于**无需语法分析的基本错误**, 如整型溢出, 能提供报错信息 (5%)
  - 编写文档, 包含flex的使用用法, 用到的正则表达式以及实现的细节等等, 最后说明项目的成员分工及**贡献百分比** (20%)
- 
- 提交项目代码以及项目文档PDF
  - DDL: **2019年11月1日** 11:59PM
  - 该项目需展示给TA, 先完成可先展示, 时间与TA预约即可

# Project 1 示例

- 输出格式参考，**请遵循图片格式！**

```
wjy@FudanT630:~/CompilerPJ$ ./lexer tests/test01.pcat
Row  Col  Type                Token/Error
4     1    reserved keyword    PROGRAM
4     9    reserved keyword    IS
5     5    reserved keyword    VAR
5     9    identifier           i
5    10    delimiter            ,
5    12    identifier           j
5    14    delimiter            :
5    16    identifier           INTEGER
5    24    operator             :=
5    27    integer              1
5    28    delimiter            ;
6     5    reserved keyword    VAR
6     9    identifier           x
6    11    delimiter            :
6    13    identifier           REAL
6    18    operator             :=
6    21    real                 2.0
6    24    delimiter            ;
7     5    reserved keyword    VAR
6    11    delimiter            :
6    13    identifier           REAL
6    18    operator             :=
6    21    real                 2.0
6    24    delimiter            ;
7     5    reserved keyword    VAR
7     9    identifier           y
7    11    delimiter            :
7    13    identifier           REAL
7    18    operator             :=
7    21    real                 3.0
7    24    delimiter            ;
8     1    reserved keyword    BEGIN
9     5    reserved keyword    WRITE
9    11    delimiter            (
9    12    string               "i = "
```

# 提交方式

- 如果文件太大，可先上传至百度云或者复旦云，再将网盘分享地址发送到TA邮箱；文件小则可直接发送到TA邮箱。
- TA邮箱： [18110240005@fudan.edu.cn](mailto:18110240005@fudan.edu.cn)
- 若对PJ有疑问，或想在上机课外时间验收，可与TA联系
- TA办公地址：新金博大厦1201室（邯郸路539号）

# Flex简介

lexer.lex

```
1  %{
2  #include "lexer.h"
3  %}
4  %option      nounput
5  %option      noyywrap
6
7  DIGIT        [0-9]
8  INTEGER      {DIGIT}+
9  REAL         {DIGIT}+"."{DIGIT}*
10 WS          [ \t\n]+
11
12 %%
13 {WS}          return WS; // skip blanks and tabs
14 <<EOF>>       return T_EOF;
15 {INTEGER}     return INTEGER;
16 {REAL}        return REAL;
17 %%
18
```

定义区

规则区

用户代码区

lexer.h

```
1  #ifndef _LEXER_H_
2  #define _LEXER_H_
3
4  #define T_EOF      0
5  #define INTEGER    1
6  #define REAL       2
7  #define WS         3
8
9  #endif
10
```



# Flex简介

编译:

flex -o lexer.c lexer.lex

g++ -c lexer.c -o lexer.o

g++ main.cpp lexer.o -o demo

```
1  #include <iostream>
2  #include <string.h>
3  #include <stdio.h>
4  #include <iomanip>
5  #include "lexer.h"
6  using namespace std;
7
8  int yylex();
9  extern "C" FILE* yyin;
10 extern "C" char* yytext;
11
12 int main(int argc, char** argv) {
13     if (argc > 1) {
14         FILE *file = fopen(argv[1], "r");
15         if (!file) {
16             cerr << "Cannot open file." << endl;
17             return 1;
18         } else {
19             yyin = file;
20         }
21     } else {
22         yyin = stdin;
23     }
24
25     while (true) {
26         int n = yylex();
27         string type = "";
28         string token = "";
29         if (n == T_EOF) break;
30         switch(n) {
31             case INTEGER:
32                 // overflow? (see file test20.pcat)
33                 type = "integer";
34                 token = yytext;
35                 break;
```

# 参考资料

- 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)