

中间代码生成实验报告

因为时间关系，最后还是选择阅读老师提供的 project 的代码。按照 readme 的实验步骤进行完成整个实验。

1、lexer.l（词法分析）

该文件通过各种正则表达式对源代码进行解读。首先一部分是将关键字作为保留的部分。然后是对标识符 IDENTIFIER，以及十六进制常量、字符常量、十进制常量、浮点数常量等进行正则表达式的书写。下一步就是对注释的解读。再往下就是运算符的匹配。

这一步的运行命令是 `lex lexer.l`。

2、parser.y（语法分析）

首先是声明了很多的 token，第二部分就是对文法的代码实现。最后的几个函数负责功能的实现，包括类型检查、生成中间代码等。

这一步的命令是 `yacc -d parser.y -v`

3、最后一步的命令是 `g++ -std=c++11 -g y.tab.c -ly -ll -o icg`

不使用上面三步的命令就需要使用 readme.md 文件中所提到的 compile 来完成，需要提前使用 `chmod` 命令给予权限。

最后附上所生成的中间代码：

```
0: main:
1: arg argc
2: i = 0
3: if i < 10 goto 7
4: goto 39
5: i++
6: goto 3
7: x = 120
8: i = 2
9: t0 = 2 + 3
10: if j < t0 goto 12
11: goto 38
12: i = 3
13: t1 = i / 6
14: t2 = 3 + t1
15: t3 = t2 + 2
16: t4 = t3 + 5
17: j = t4
18: j = arr[7]
19: if j < 5 goto 21
20: goto 23
21: if i == 3 goto 25
22: goto 23
23: if j != 5 goto 25
24: goto 37
25: j = 2
26: if 2 < 4 goto 28
27: goto 30
28: goto 38
29: goto 37
30: if 2 > 3 goto 32
31: goto 34
32: goto 9
33: goto 37
34: t5 = 4 + 5
35: t6 = minus t5
36: i = t6
37: goto 9
38: goto 5
39: exit
```