

TINY+ 语法分析程序

- 运行环境: Linux(Ubuntu) g++ make
- 可执行程序路径: ./tiny/bin/main
- 源程序文件夹: ./tiny
- 报告: REPORT.pdf
- 运行方法:

```
1 cd tiny # 进入tiny文件夹
2 make # Makefile
3 ./bin/main test.tny #输出语法树到屏幕(控制台)
4 ./bin/main test.tny tree #输出语法树到屏幕(控制台)
5 ./bin/main test.tny tokens #输出TOKEN序列到屏幕(控制台)
6 ./bin/main test.tny tokens tokens #输出TOKEN序列到tokens文件
```

- 文件结构

```
1 tiny
2 |— .vscode
3 |— bin
4 |   |— main //可执行程序
5 |— include // .h文件夹
6 |   |— errors.h //声明枚举错误类型
7 |   |— global.h //声明全局变量等
8 |   |— scan.h //声明getToken获取序列函数
9 |   |— print.h //声明实现printToken函数
10 |   |— parser.h //声明语法分析器头文件
11 |   |— ... //其他实验相关文件, 此处可忽略
12 |— src // .cpp文件夹
13 |   |— main.cpp //主函数
14 |   |— scan.cpp //实现getToken获取序列
15 |   |— print.cpp //实现输出Token的函数
16 |   |— parser.cpp //实现语法分析
17 |   |— ... //其他实验相关文件, 此处可忽略
18 |— makedfile //Makefile文件
19 |— test.tny //测试tiny源程序
20 |— tokens //输出文件
```

- 输出示例
以下为部分示例, 更多测试详见REPORT.pdf。

```
1 root@DESKTOP-6L638NB:/mnt/e/code/Compiler/tiny# ./bin/main
test2.tny tree
2 语法树如下:
3   READ x
4   If
5       LogicOp: (KEY,or)
6       LogicOp: (KEY,and)
7       Op: (SYM,<)
8       ID: x
9       Const: 10
10      Op: (SYM,>)
11      ID: x
12      Const: 5
13      Op: (SYM,<)
14      ID: x
15      Const: 9
16      Assign to fact
17      Const: 4
18      Assign to fact
19      Const: 6
20  Repeat
21      Assign to A
22      Op: (SYM,*)
23      ID: A
24      Const: 2
25      Op: (SYM,<)
26      Op: (SYM,+)
27      ID: A
28      ID: C
29      Op: (SYM,+)
30      ID: B
31      ID: D
32  while
33      Op: (SYM,<)
34      Op: (SYM,+)
35      ID: A
36      Op: (SYM,+)
37      ID: B
38      ID: C
39      Const: 10
40      Assign to B
41      Op: (SYM,+)
42      ID: B
43      Const: 3
```