TINY+ 词法分析程序

- 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：

```
cd tiny  # 进入tiny文件夹
make       # Makefile
./bin/main test.tny    #输出TOKEN序列到屏幕(控制台)
./bin/main test.tny tokens #输出TOKEN序列到tokens文件中
```

- 文件结构

```
tiny
├── .vscode
├── bin
|     ├── main //可执行程序
├── include  //.h文件夹
|     ├── errors.h  //声明枚举错误类型
|     ├── global.h  //声明全局变量等
|     ├── scan.h  //声明getToken获取序列函数
|     ├── print.h  //声明实现printToken函数
|     ├── ... //其他两实验相关文件，此处可忽略
├── src   //.cpp文件夹
|     ├── main.cpp  //主函数
|     ├── scan.cpp  //实现getToken获取序列
|     ├── print.cpp  //实现输出Token的函数
|     ├── ... //其他两实验相关文件，此处可忽略
├── Makedfile  //Makefile文件
├── test.tny  //测试tiny源程序
├── tokens  //输出文件
```

- 输出示例
  以下为部分示例，更多测试详见REPORT.pdf。

```
root@DESKTOP-6L638NB:/mnt/e/code/Complier/tiny# ./bin/main test2.tny
TOKENS序列：
(KEY, int)
(ID, x)
```

```
(TK_COMMA, ,)
(ID, fact)
(TK_COMMA, ,)
(ID, A)
(TK_COMMA, ,)
(ID, B)
(TK_COMMA, ,)
(ID, C)
(TK_COMMA, ,)
(ID, D)
(TK_SEMICOLON, ;)
(KEY, read)
(ID, x)
(TK_SEMICOLON, ;)
(KEY, if)
(ID, x)
(TK_LSS, <)
(NUM, 10)
(KEY, and)
(ID, x)
(TK_GTR, >)
(NUM, 5)
(KEY, or)
(ID, x)
(TK_LSS, <)
(NUM, 9)
(KEY, then)
(ID, fact)
(TK_ASSIGN, :=)
(NUM, 4)
(KEY, else)
(ID, fact)
(TK_ASSIGN, :=)
(NUM, 6)
(KEY, end)
(TK_SEMICOLON, ;)
(KEY, repeat)
(ID, A)
(TK_ASSIGN, :=)
(ID, A)
(TK_MUL, *)
(NUM, 2)
(TK_SEMICOLON, ;)
(KEY, until)
(TK_LP, ()
(ID, A)
(TK_ADD, +)
(ID, C)
(TK_RP, ))
(TK_LSS, <)
(TK_LP, ()
(ID, B)
(TK_ADD, +)
(ID, D)
(TK_RP, ))
(TK_SEMICOLON, ;)
(KEY, while)
(TK_LP, ()
(ID, A)
(TK_ADD, +)
(ID, B)
(TK_ADD, +)
(ID, C)
(TK_RP, ))
(TK_LSS, <)
(NUM, 10)
(KEY, do)
(ID, B)
(TK_ASSIGN, :=)
(ID, B)
(TK_ADD, +)
(NUM, 3)
(TK_SEMICOLON, ;)
(KEY, end)
DONE
root@DESKTOP-6L638NB:/mnt/e/code/Complier/tiny# ./bin/main test2.tny tokens
DONE
```