编译原理实验一「Tiny词法扫描」说明文档

2017141461179 王兆基

# 实验环境：

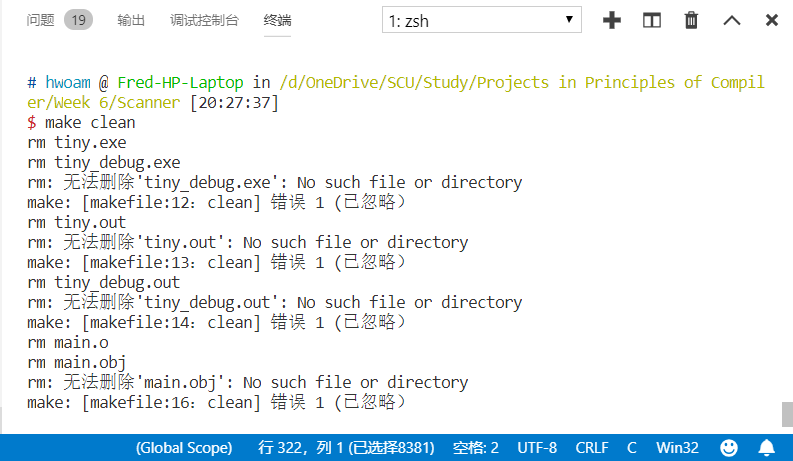
* **设备：**Microsoft Surface Pro 5 i5 8+256G
* **操作系统：**Windows 10 64位
* **Linux模拟环境：**MSYS2 MinGW 64 (mintty)

（和在Linux下操作是完全一样的）

* **Shell：**zsh
* **编程语言：**C Language
* **编译器：**gcc.exe (Rev2, Built by MSYS2 project) 8.3.0
* **Make：**GNU Make 4.2.1 (为 x86\_64-pc-msys 编译)
* **文本编辑器：**VSCode作为编辑器，文本文件采用utf-8编码

# 使用流程

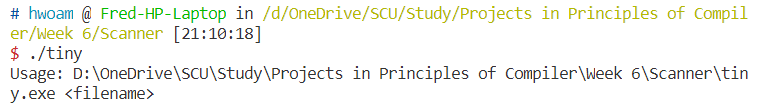
　　首先，运行make clean做一点清理工作，把旧的中间文件删除（已经考虑到了.exe和.out，以及.obj和.o文件可能在不同系统下生成的差异）：

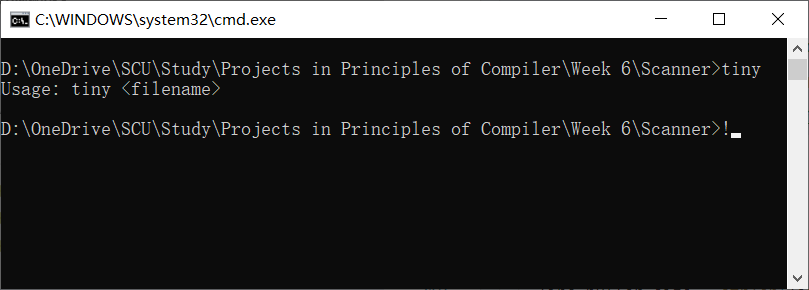


　　然后，运行make命令编译：



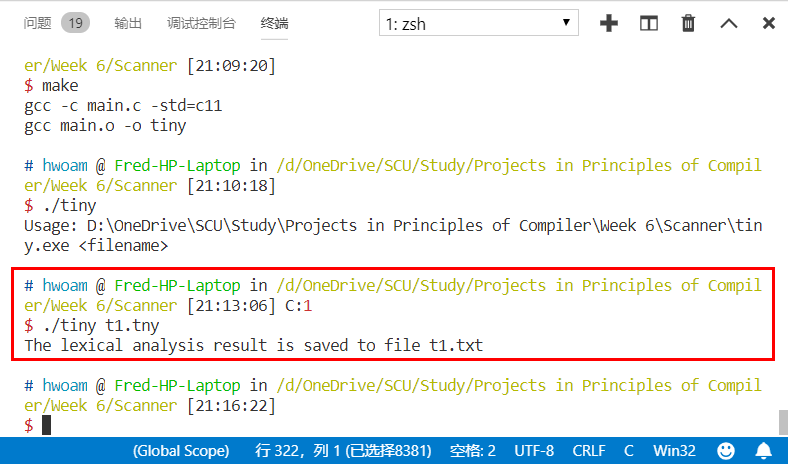
　　编译成功后，运行./tiny（bash/zsh/PowerShell）或者tiny（Windows的cmd）：





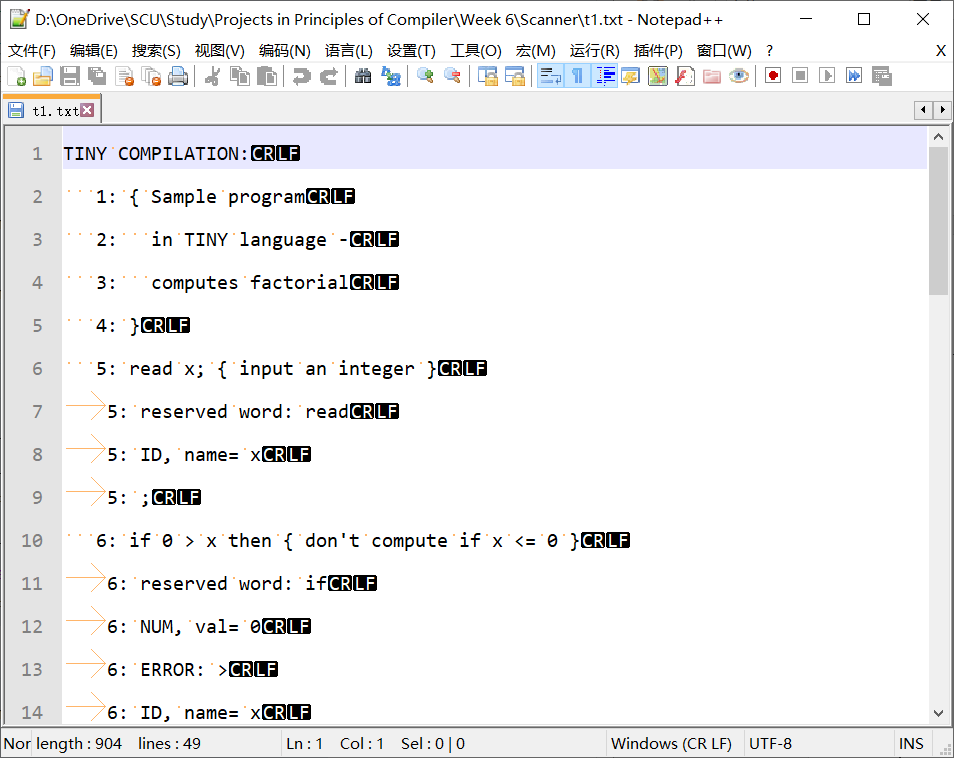
　　如图，如果终端是bash/zsh/PowerShell，那么是上面第一张图显示的提示。如果是cmd，那么是上面第二张图的提示。即提示了用户「正确使用方法」是「tiny文件 <filename>」的格式调用：

　　然后输入./tiny t1.tny调用程序，如果终端不是bash/zsh/PowerShell而是cmd，调用命令同上，不再赘述：

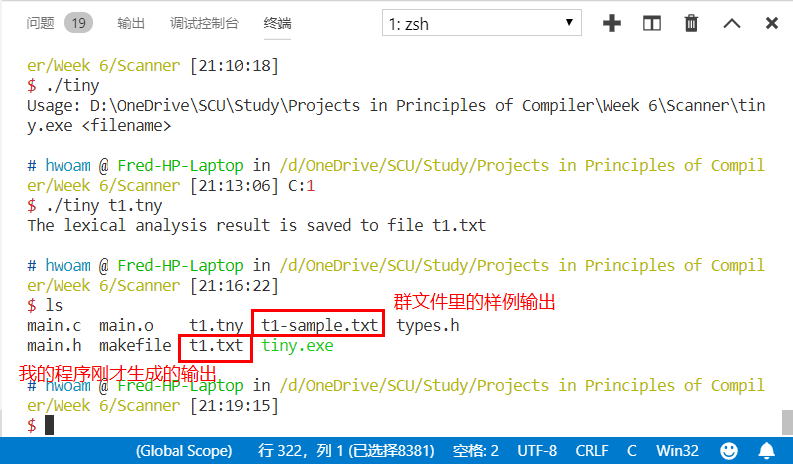


　　如图，程序会提示把词法扫描的结果输出到了t1.txt文件当中。

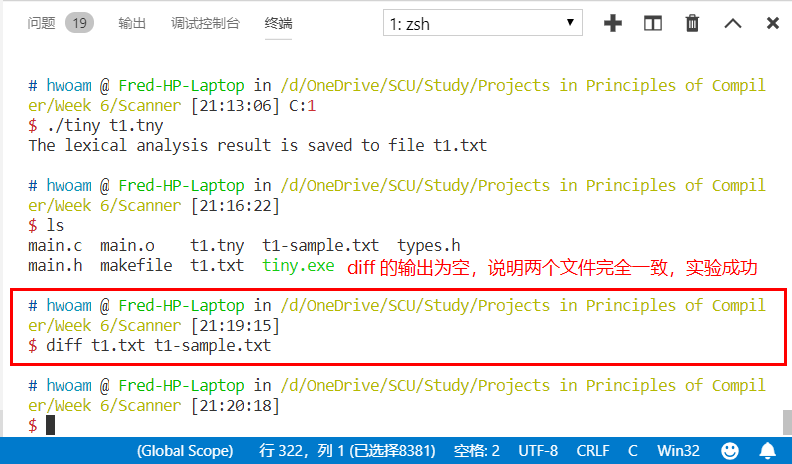
　　打开t1.txt文件：



　　和预期的输出格式一致，不过到底和样例有没有区别还是要比较一下。我将样例命名为t1-sample.txt放在了同一目录下：



　　输入diff t1.txt t1-sample.txt比较：



　　diff的输出为空，说明两个文件完全一致，实验成功！