## 2.5 RT-Thread操作系统开发

### 2.5.1 设计方案

（1）学习在线学习平台上的RT-Thread操作系统原理及应用，了解RT-Thread操作系统线程、内存管理、定时器、信号量、互斥等原理；

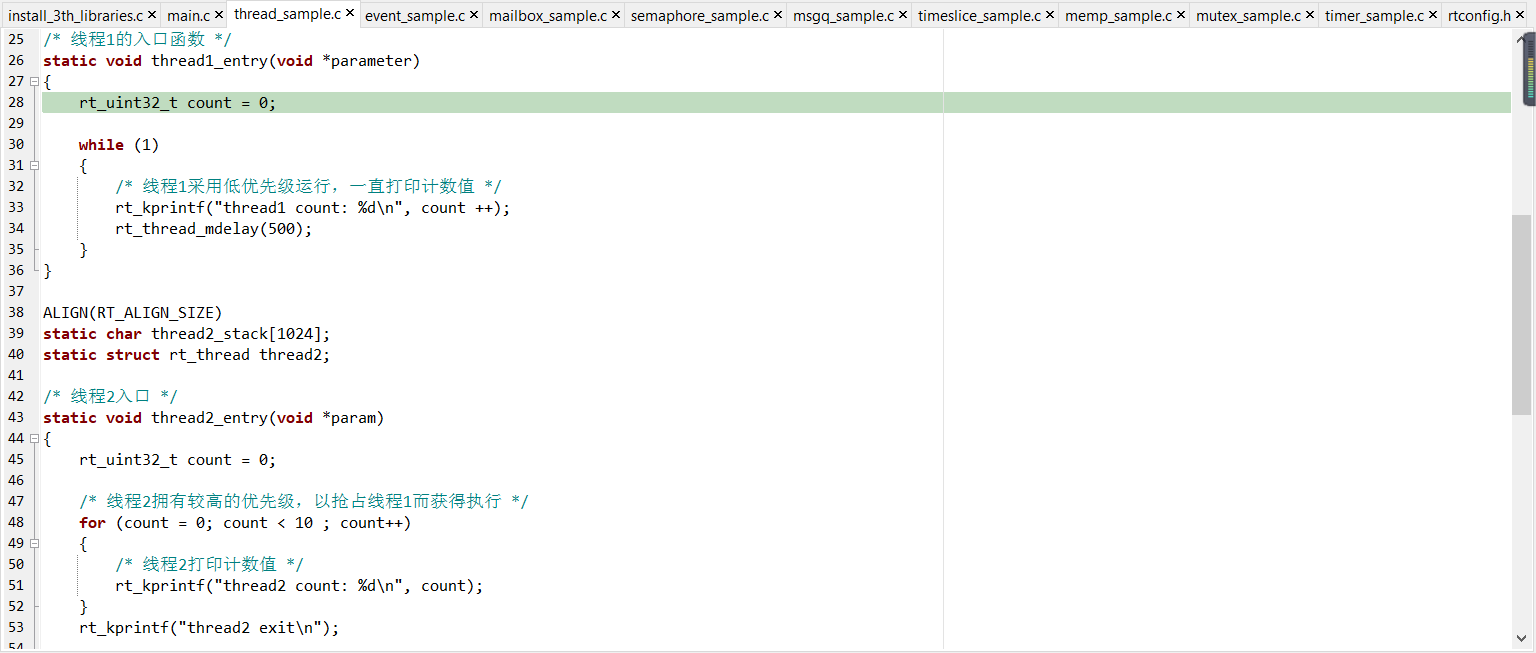
（2）在龙芯1B开发板上完成对RT-Thread操作系统的线程实现，信号量实现，互斥实现等操作；

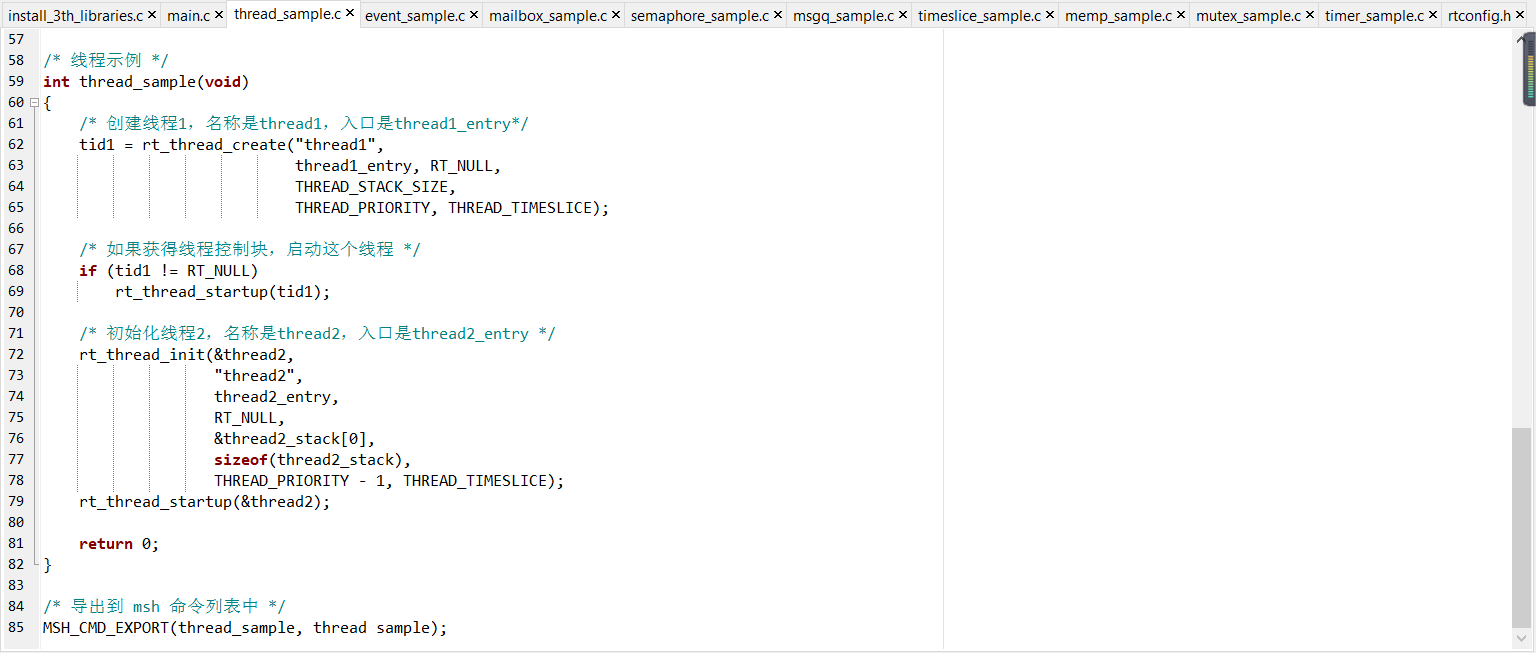
（3）编写操作系统经典实例读者写者问题，实现写者、读者互斥访问文件资源，多个读者同时访问文件资源，有读者访问资源时，写者必须等待，读者优先级高于写者

### 2.5.2 实现方法

（1）参考实验手册中的示例，在RT-Thread操作系统下实现线程的创建、初始化与脱离，设计thread1、thread2两个线程，具体实现如下

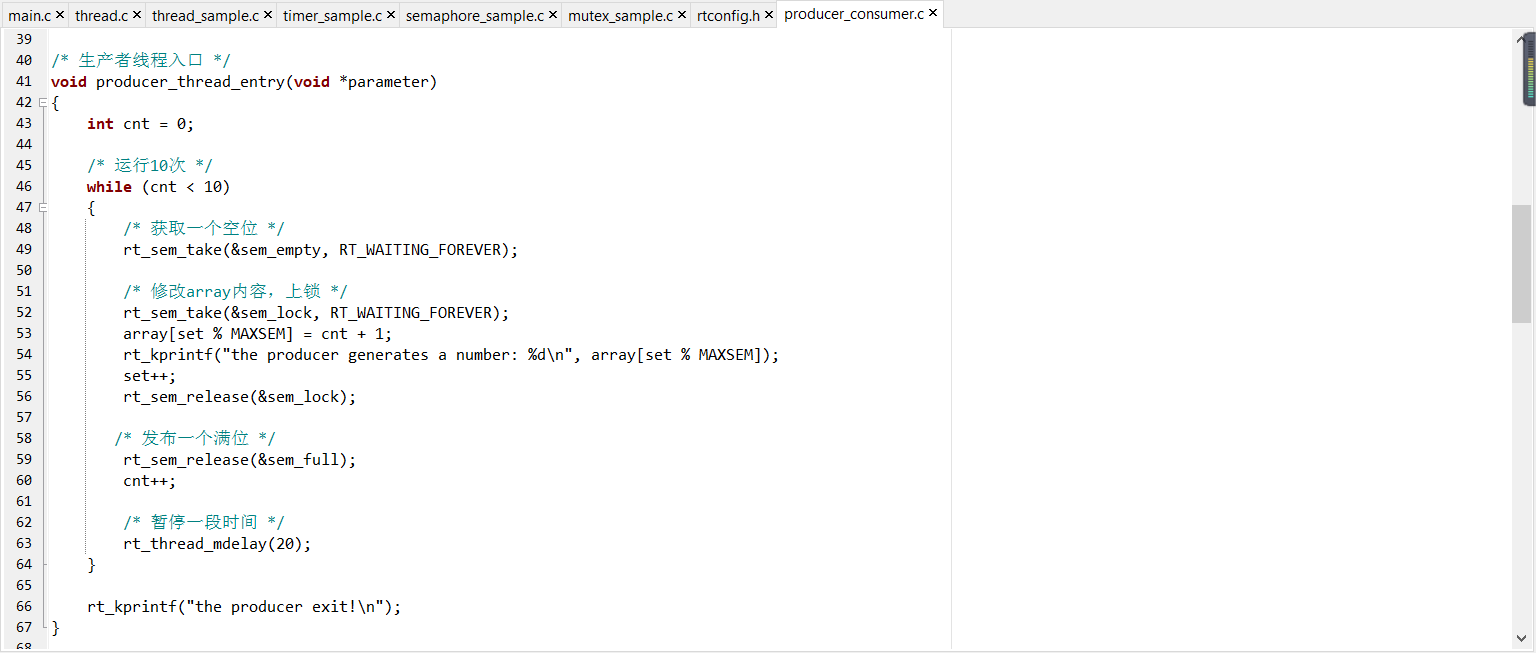


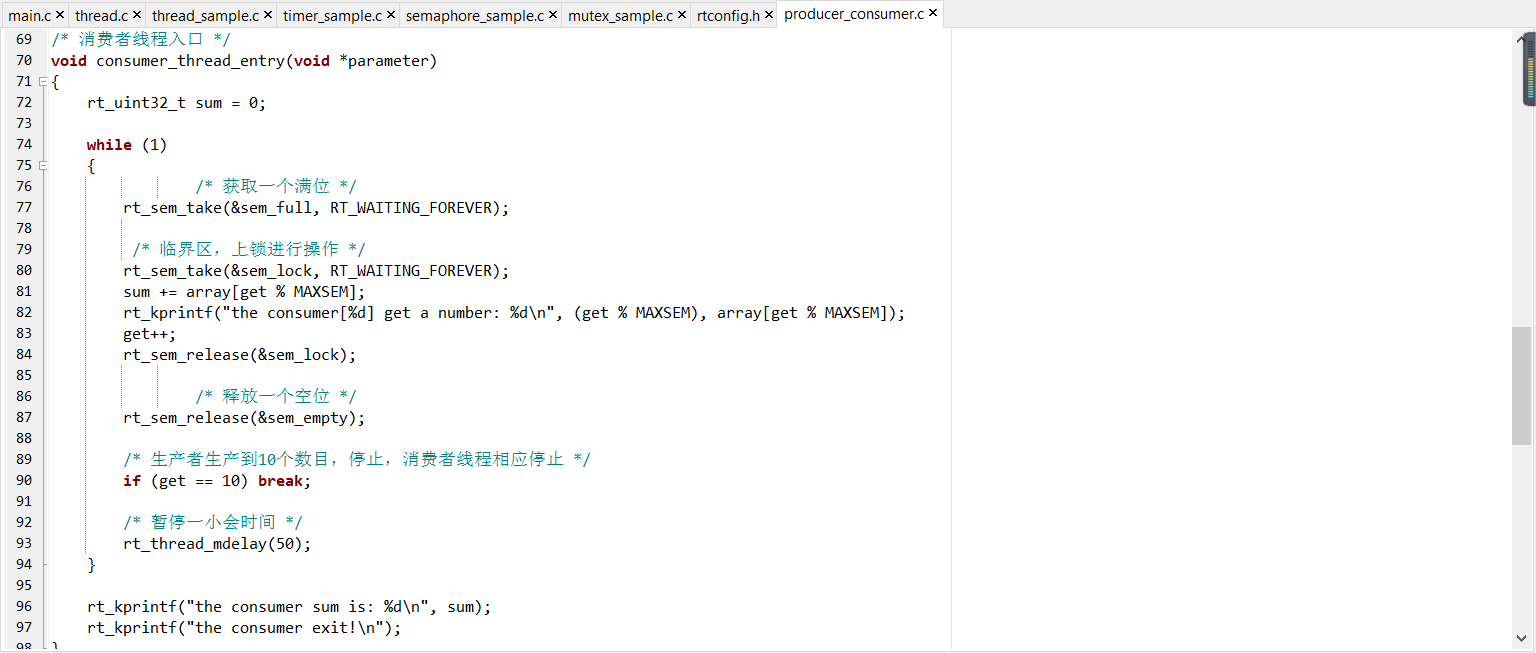


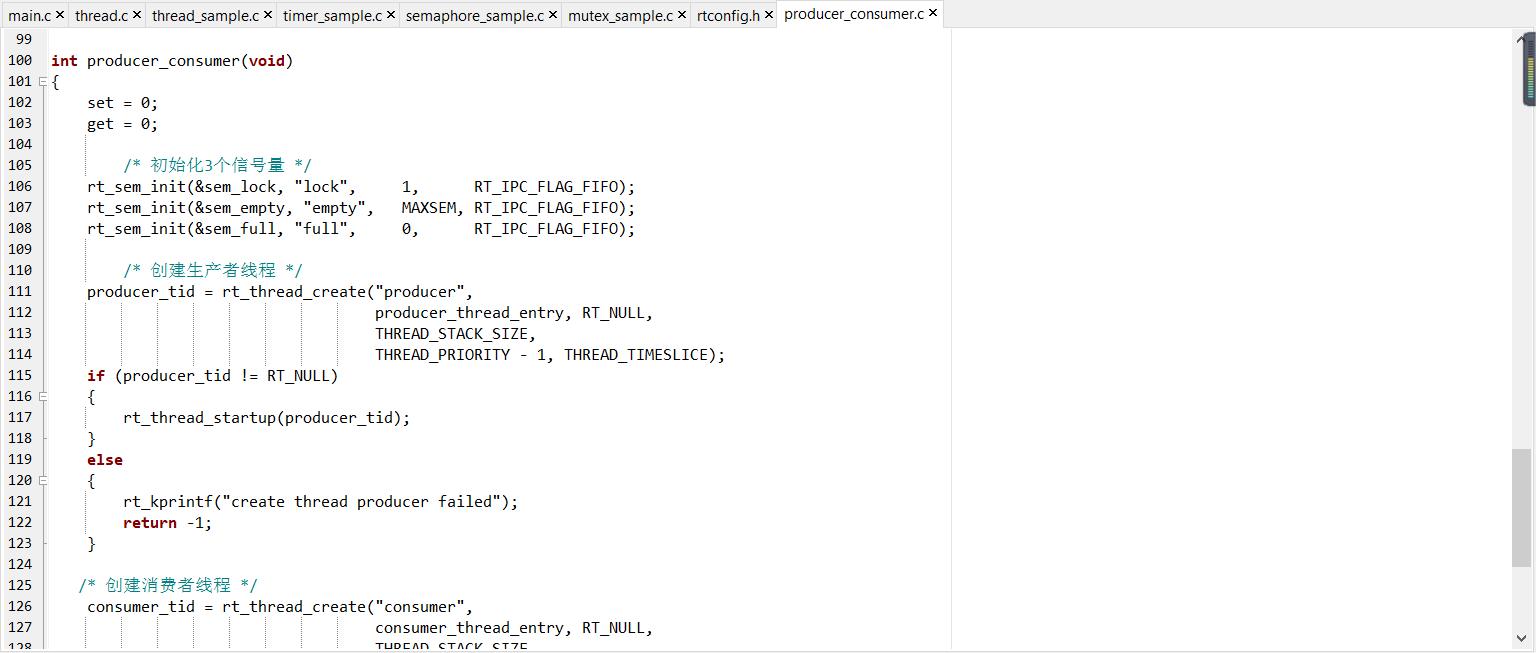


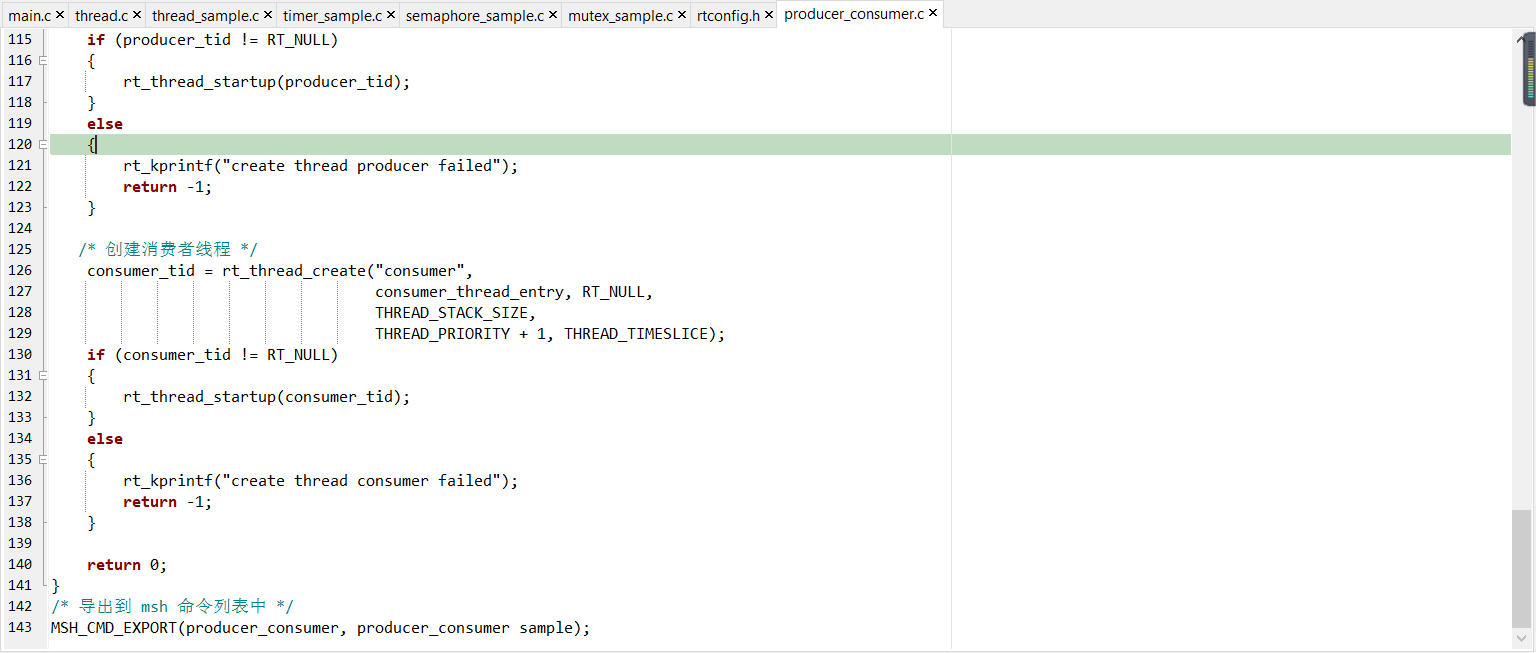
（2）理解信号量的基本原理，参考平台上的实例，使用信号量达到线程间同步，在中断、线程中均可释放信号量，具体代码如下











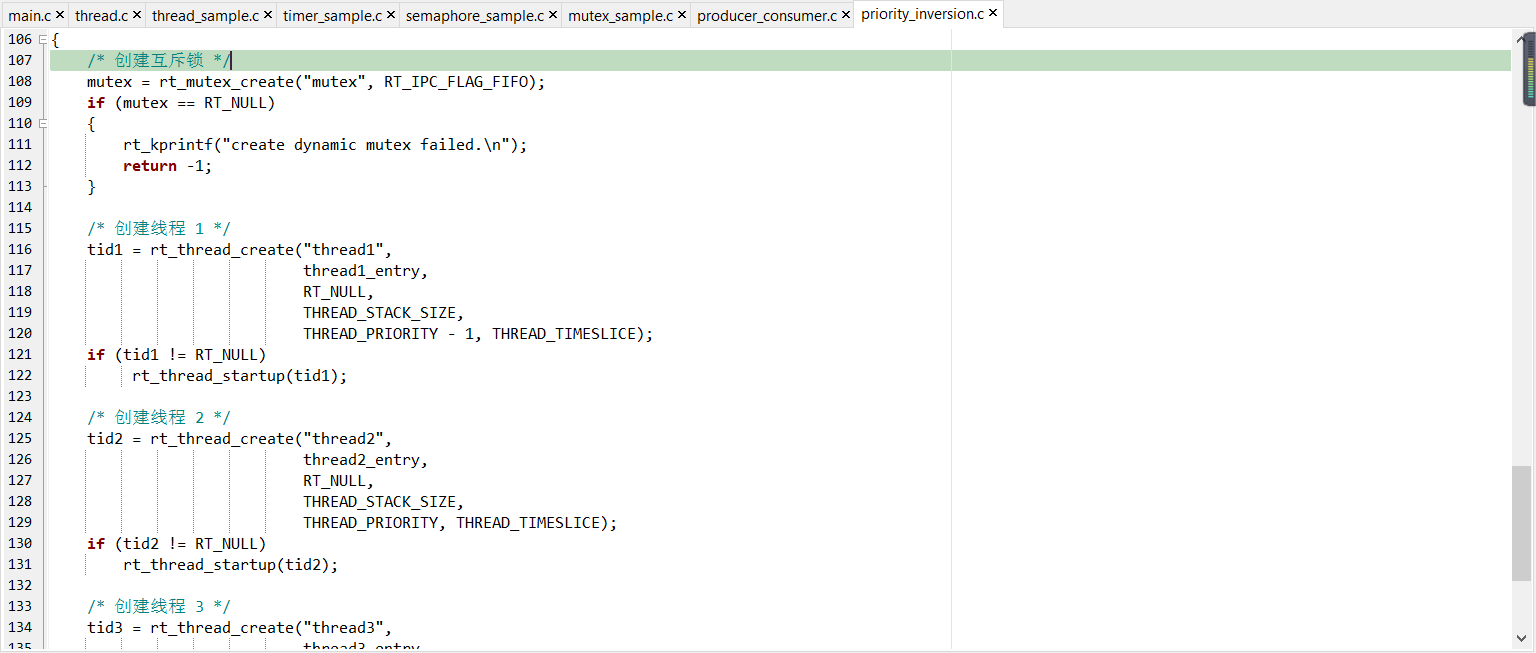
（3）学习理解互斥量的基本原理，使用互斥量来达到线程间同步

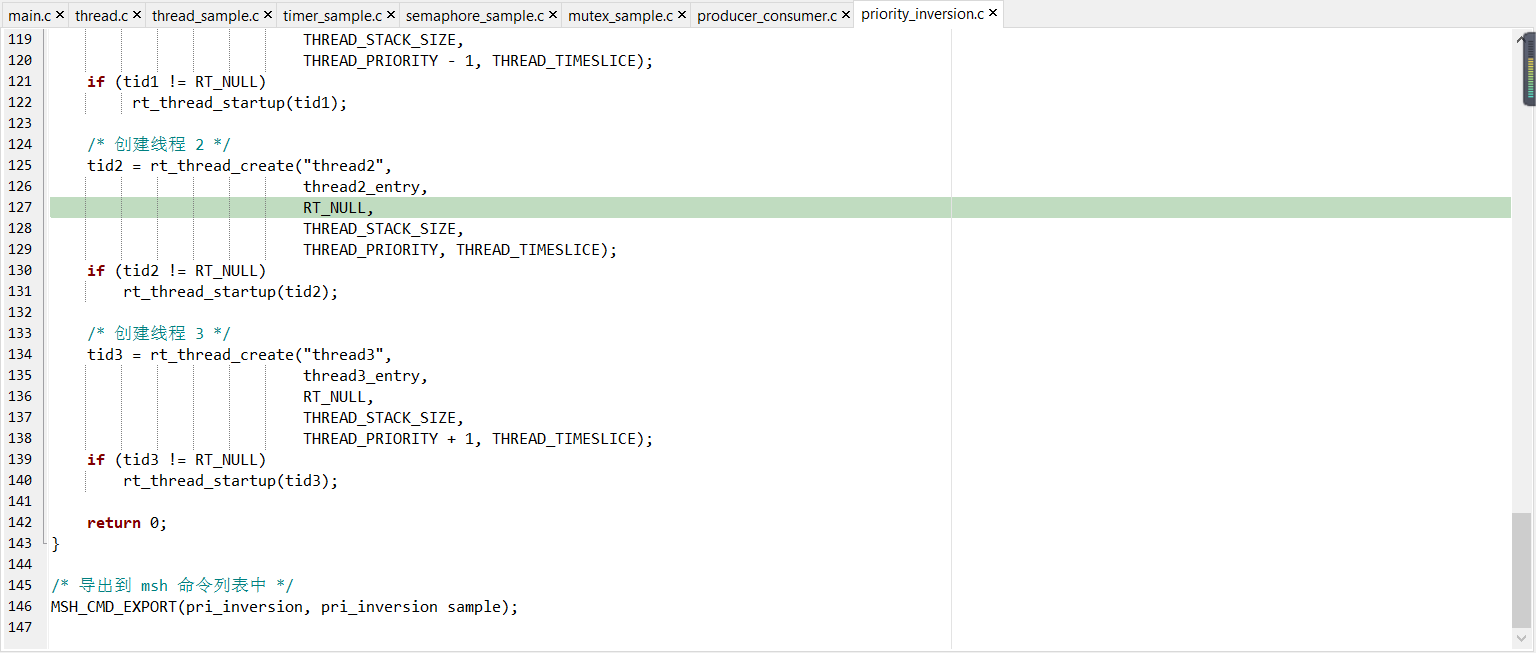
部分代码如下











（4）通过运用信号量、互斥量实现进程同步的理解学习，尝试编写操作系统经典问题读者-写者问题，

1.多个读者同时对文件执行读操作

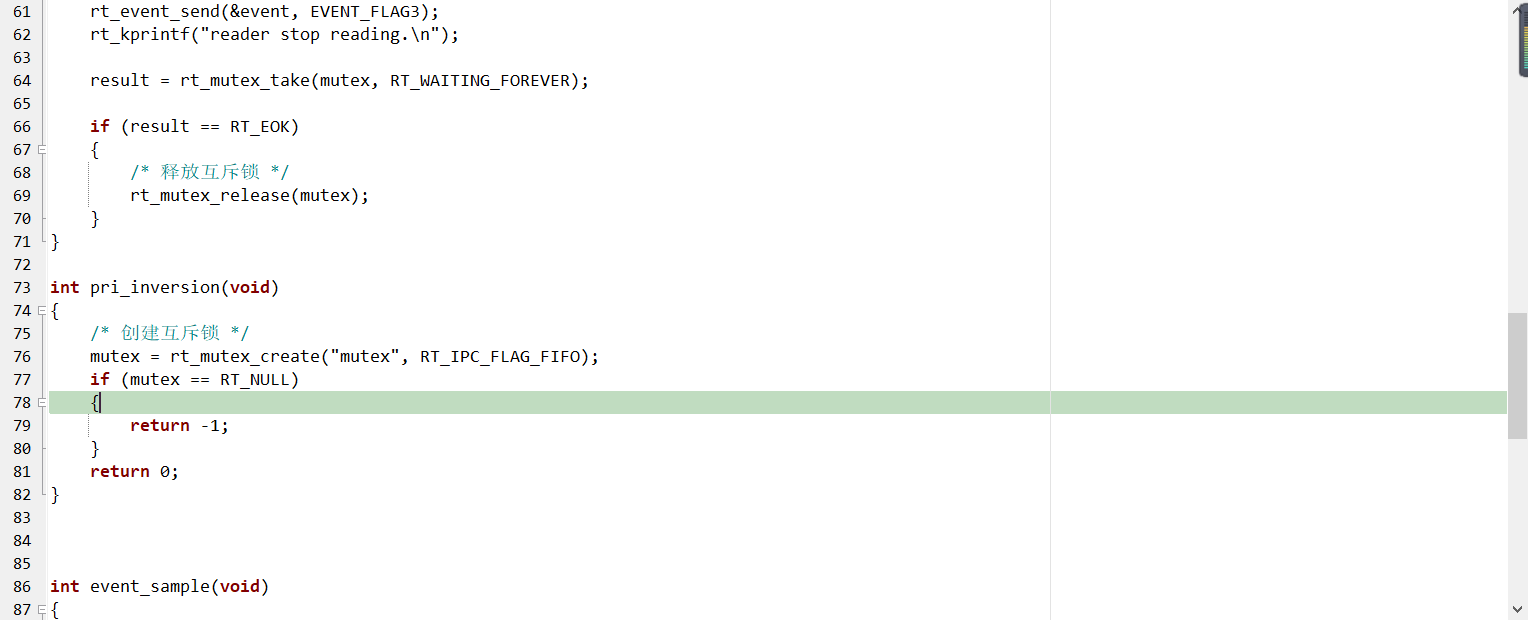
2.写者在完成写操作之前不允许其他读者或写者工作

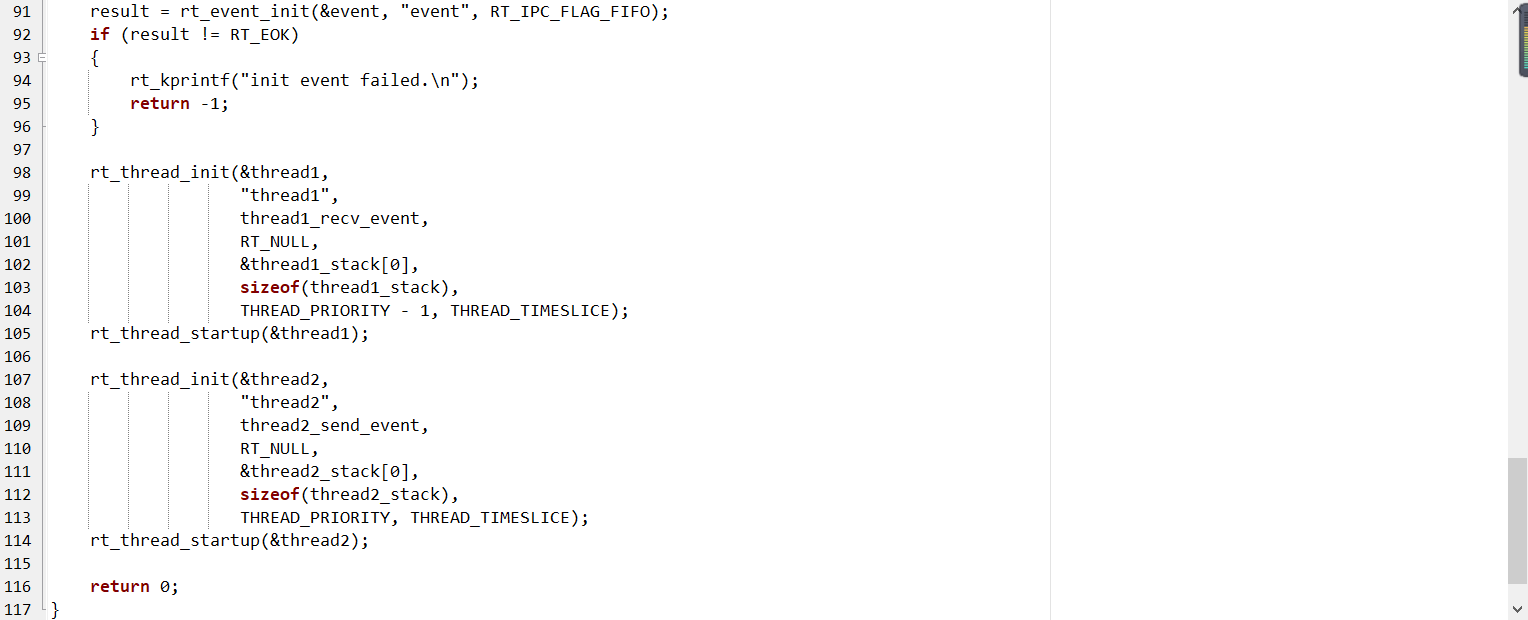
3.两个进程的互斥关系

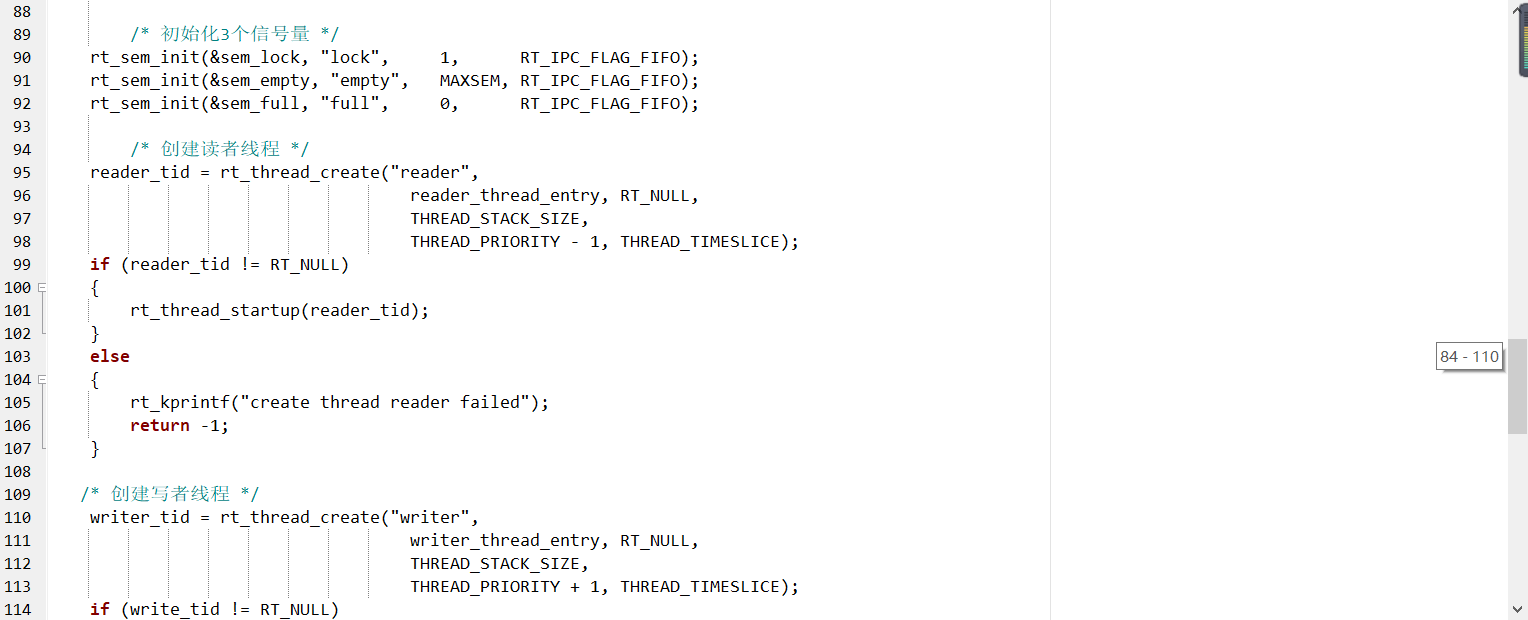
写进程和写进程互斥，写进程和读进程互斥，读进程和读进程不存在互斥问题

设置互斥信号量，写着访问共享文件前后分别执行P、V操作

P和V其实就是对共享文件的“加锁”和“解锁”，让第一个访问文件的读进程“加锁”，让左后一个访问完文件的读进程“解锁”



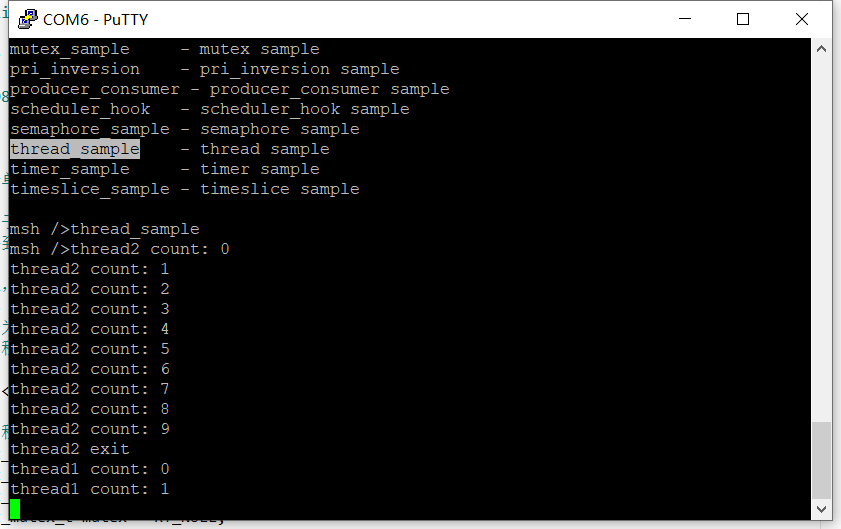




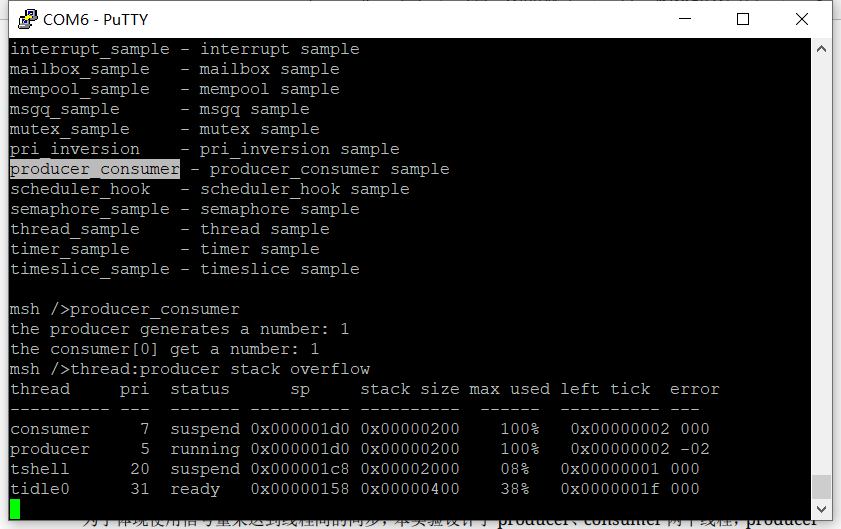
## 3.4 RT-Thread操作系统开发

（1）终端测试结果

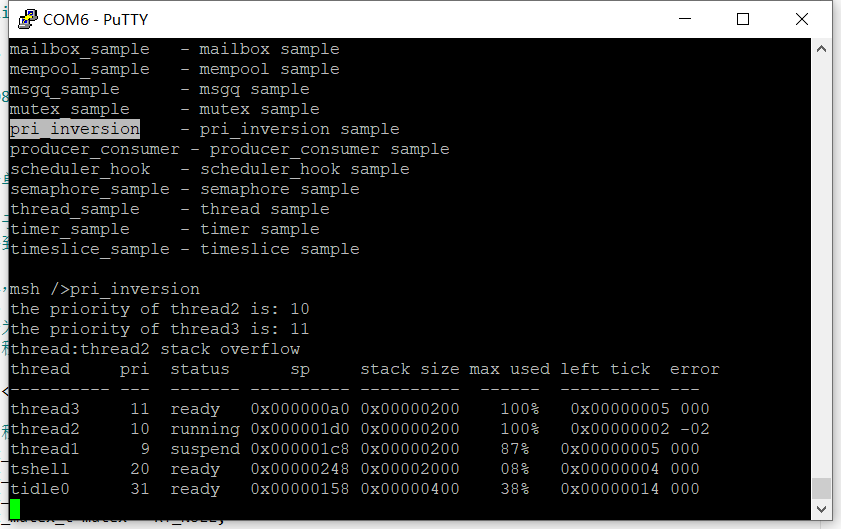
1.信号量实现，终端输入thread\_sample 命令启动，效果如图



2.使用信号量实现线程间的同步，终端输入producer\_consumer 命令启动，效果如图

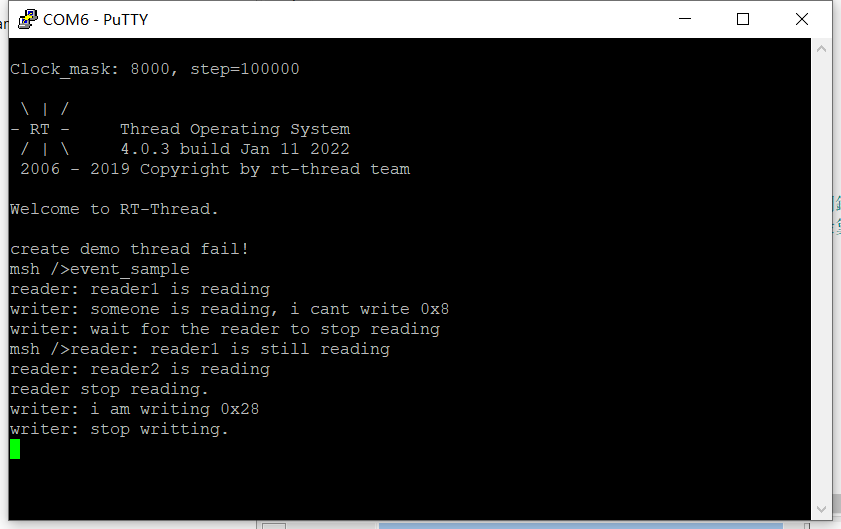


3使用互斥量实现线程间的同步，终端输入



4.运用信号量、多线程和互斥等相关知识实现读者-写者问题

测试结果如图



读者进程进入内存资源时，输出reader1 is reading

写者进程与读者进程互斥访问内存，写者进程无法占用内存，输出someone is reading，I cant write同时输出wait for the reader to stop reading

又一读者进程开始访问内存资源是，第一个进程仍旧占用内存资源，输出reader：reader1 is still reading

读者进程间不需互斥访问内存，则读者二进入同时访问同一内存地址，输出reader2 is reading

占用内存结束，输出：reader stop reading

此时读者进程结束，写者进程开始访问内存，输出writer：I am writing

写者进程占用内存结束，输出：writer：stop writing

（2）遇到的问题

1.第一次运行时，putty终端运行时循环执行，命令无法打断

main.c中的for循环代码没有停止命令，修改main.c中的demo\_thread函数使其能够跳出循环，终端便可以通过tab键查看命令。

2.两个读进程并发执行，有可能两个读进程先后执行，第二个读进程阻塞

设置另一个互斥信号量保证互斥访问