

****

《实用操作系统》实验报告

（四）

**姓　　名 ：宋泽涛**

**学　　号 ：25120222201292**

**学 院 ：信息学院**

**专 业 ：软件工程**

**2024年 10 月**

# 实验目的

* 加强对进程概念的理解。
* 进一步了解并发执行的实质。
* 分析进程争用资源的现象，学习解决进程互斥的方法
* 了解Linux系统中进程通信的基本原理

# 实验环境

* + PC + Linux Red Hat操作系统
  + GCC

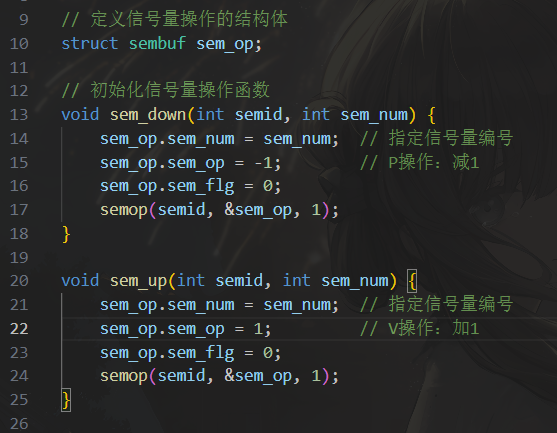
# 实验内容

* 使用信号量实现有限缓冲区的生产者和消费者问题
* 使用信号量实现读进程具有优先权的读者和写者问题

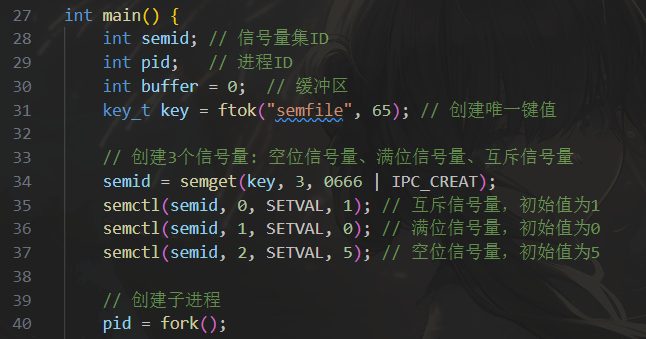
# 实验具体实现

1.使用信号量实现有限缓冲区的生产者和消费者问题

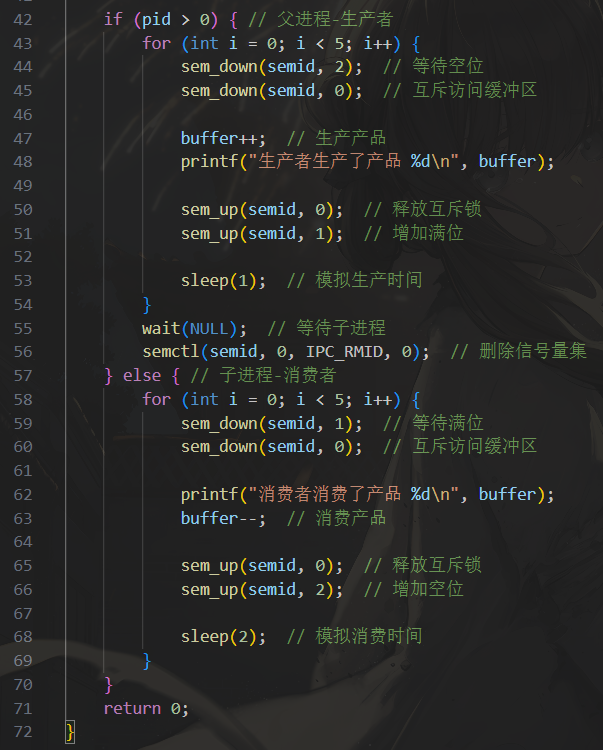
(1).编写如下c语言程序，首先定义信号量操作的结构体，再编写两个信号量操作函数，sem\_down对应信号量的P操作，sem\_up对应信号量的V操作。



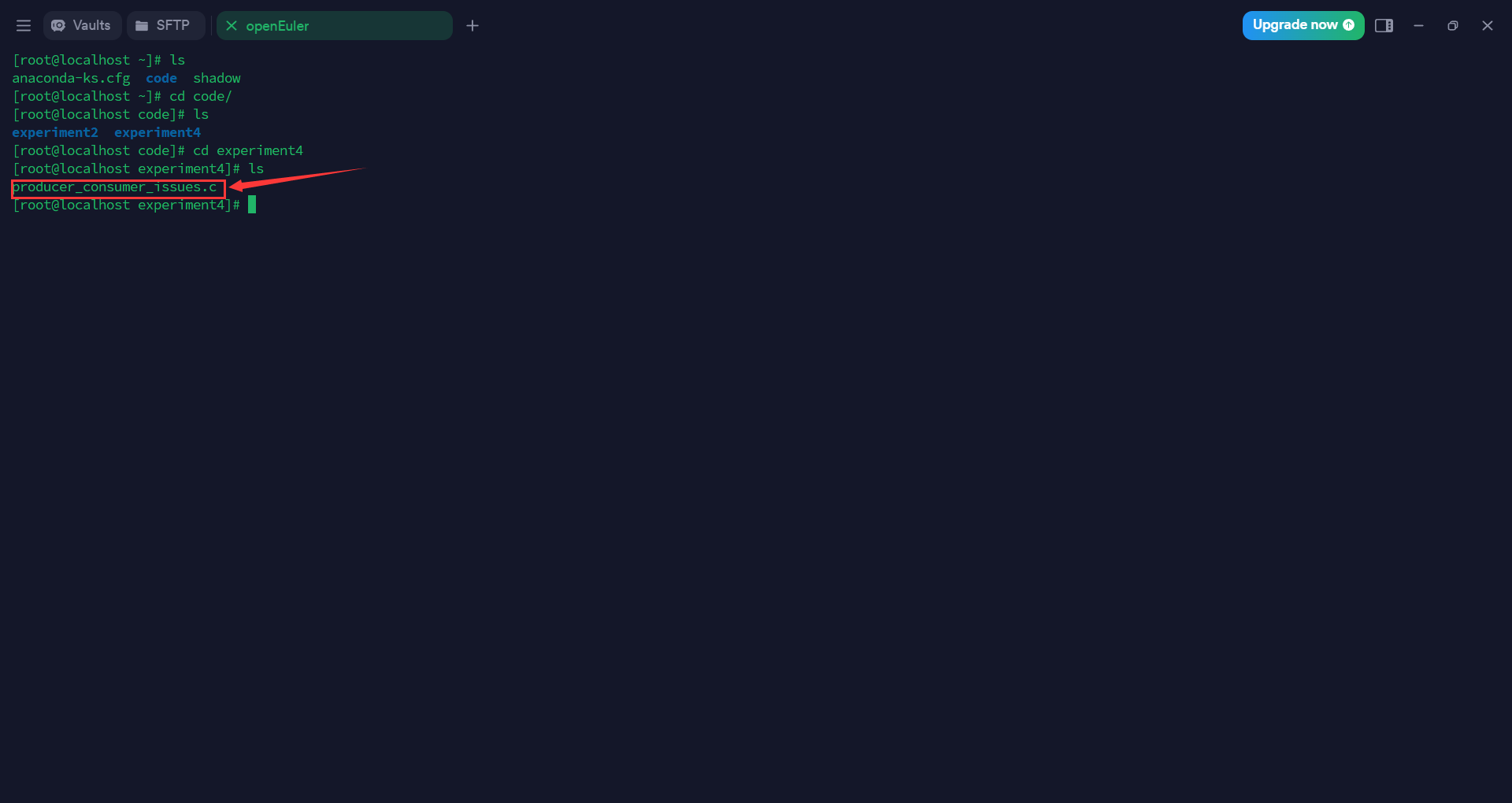
(2).定义信号量集ID、进程ID和缓冲区，创建3个信号量并初始化，再使用fork()函数创建子进程。

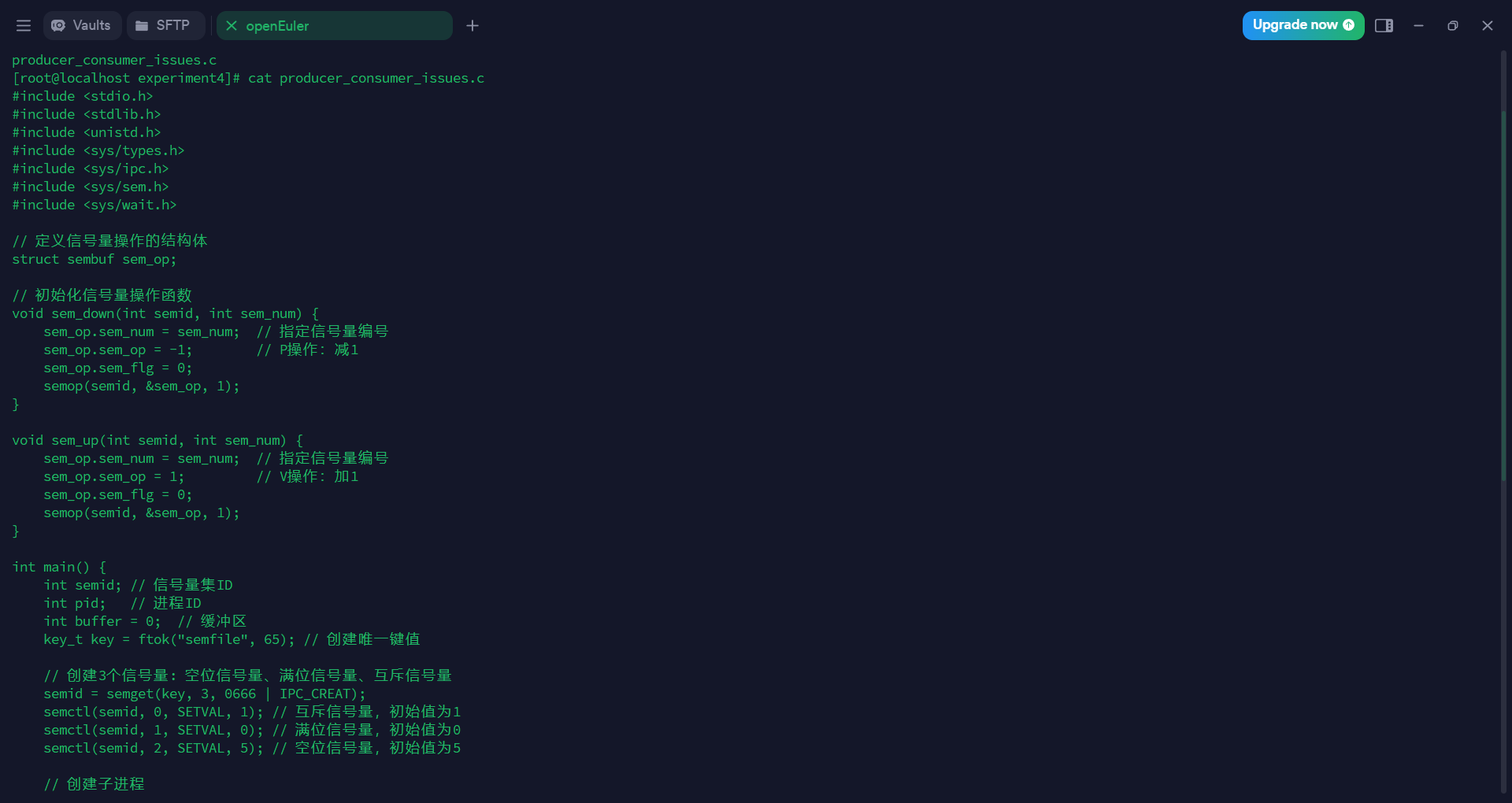


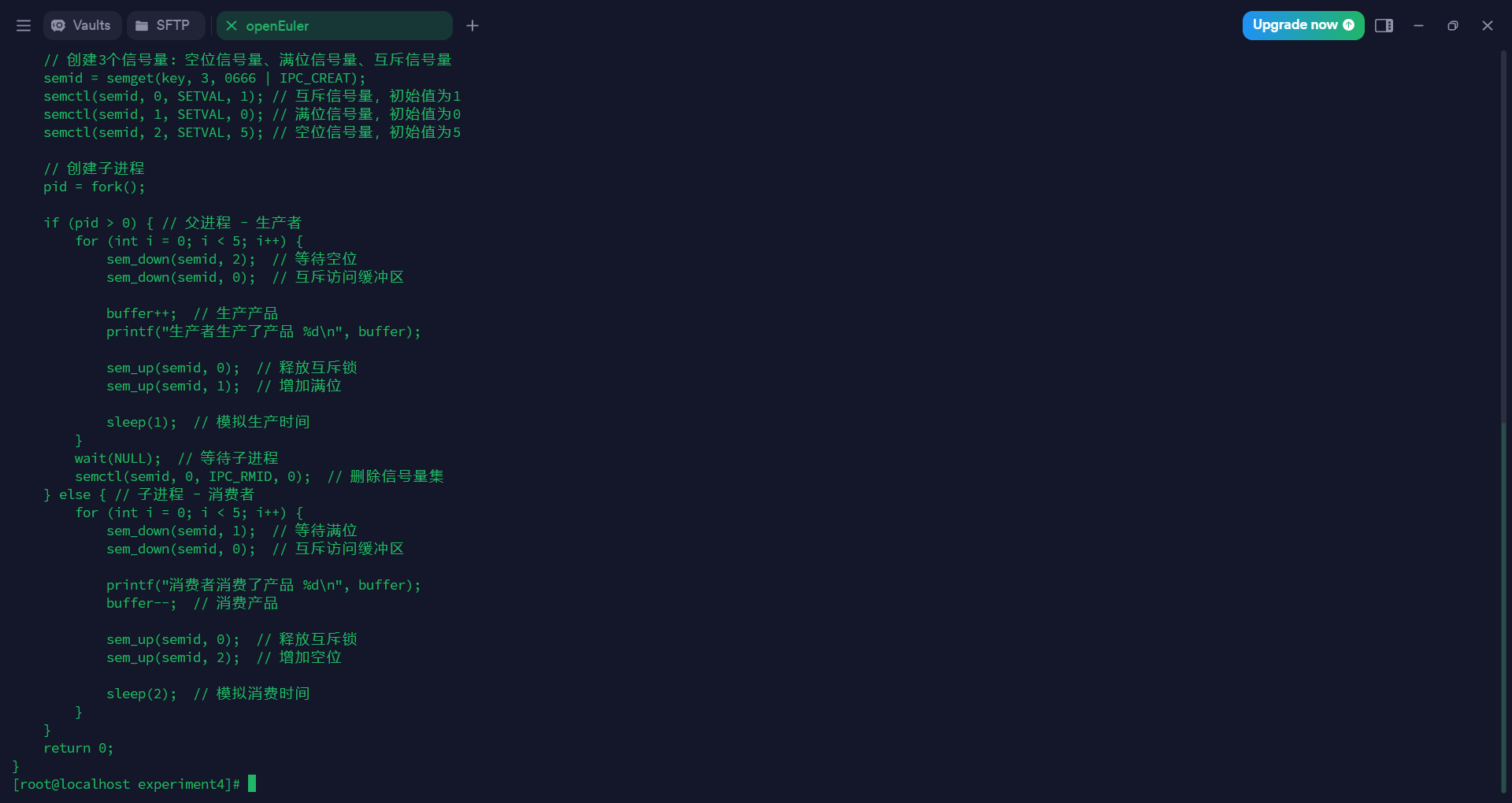
(3).模拟生产者-消费者访问缓冲区的过程，生产者往缓冲区写入，消费者从缓冲区拿出。当缓冲区满，生产者需要等待消费者拿出数据；当缓冲区空，消费者需要等待生产者写入数据。



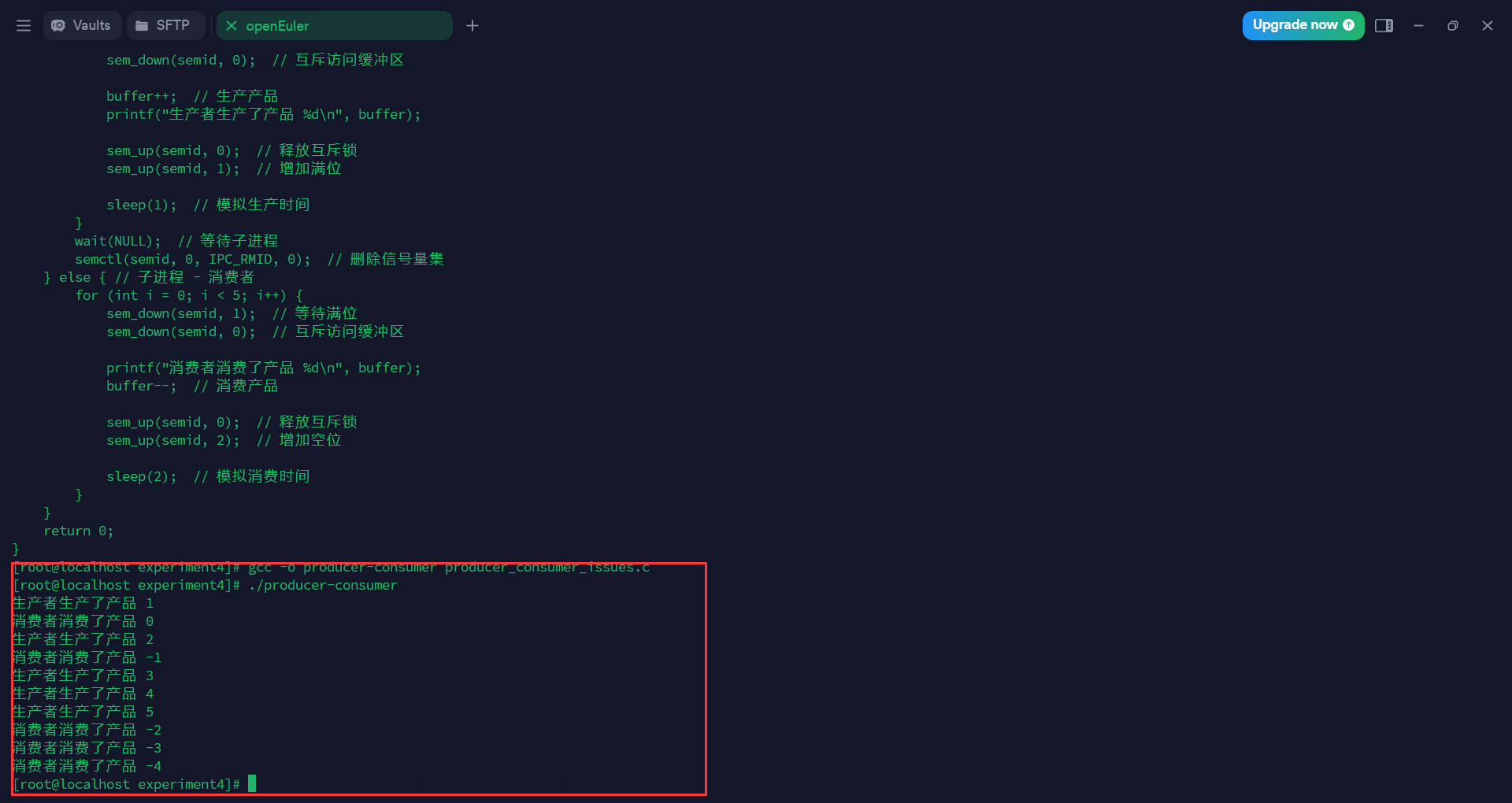
(4).将本地编写好的c语言程序复制到openEuler-Linux虚拟机中，并查看内容。

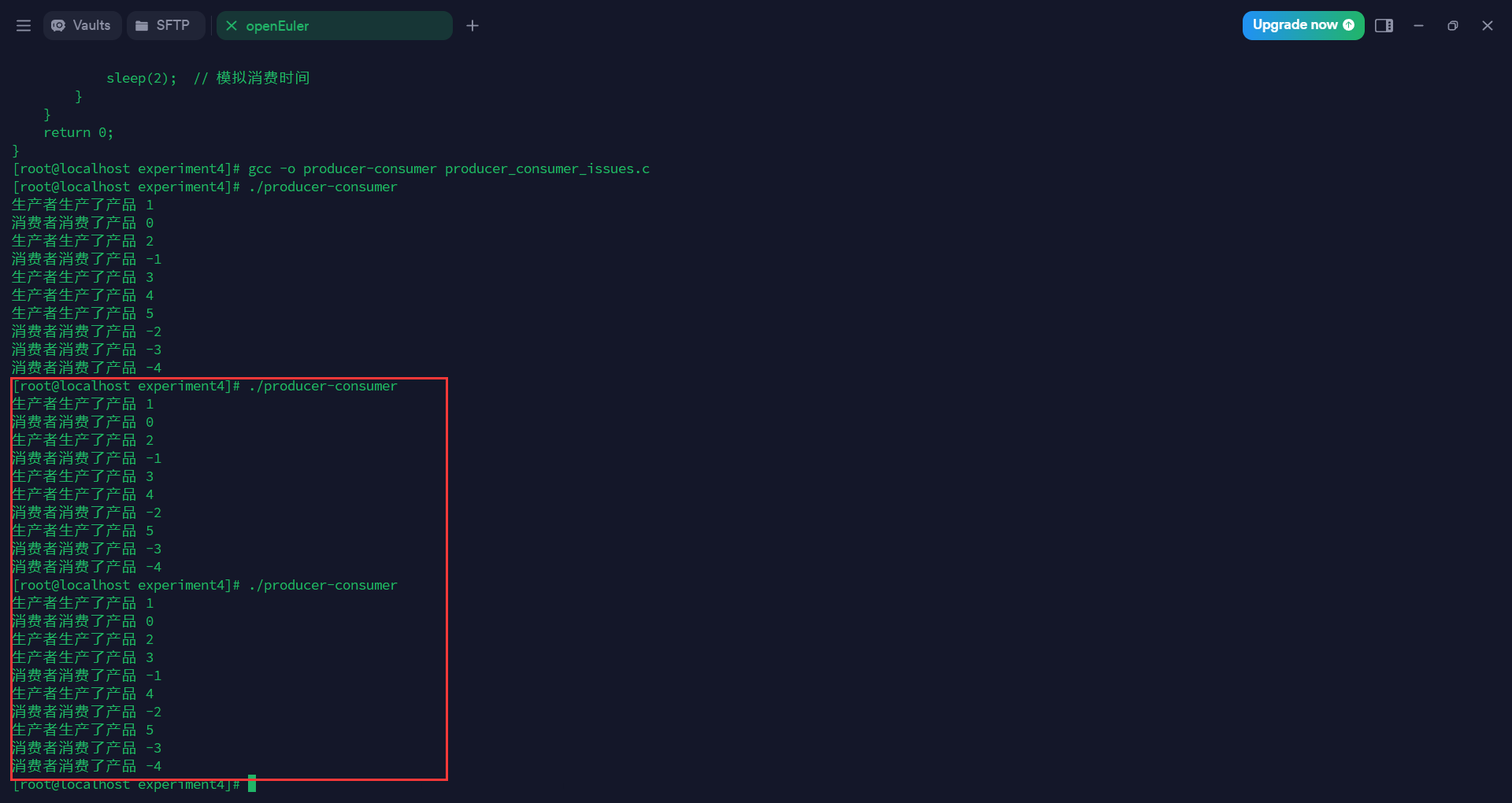






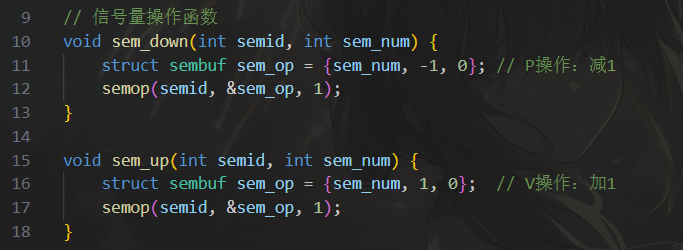
(5).使用gcc命令编译运行代码。



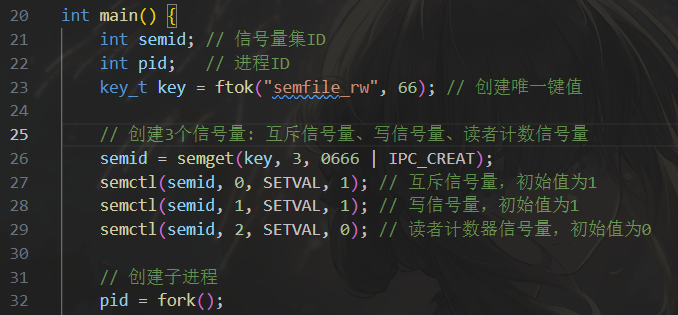


2.使用信号量实现读进程具有优先权的读者和写者问题

(1).编写如下c语言程序，定义信号量操作函数，sem\_down对应P操作，sem\_up对应V操作。



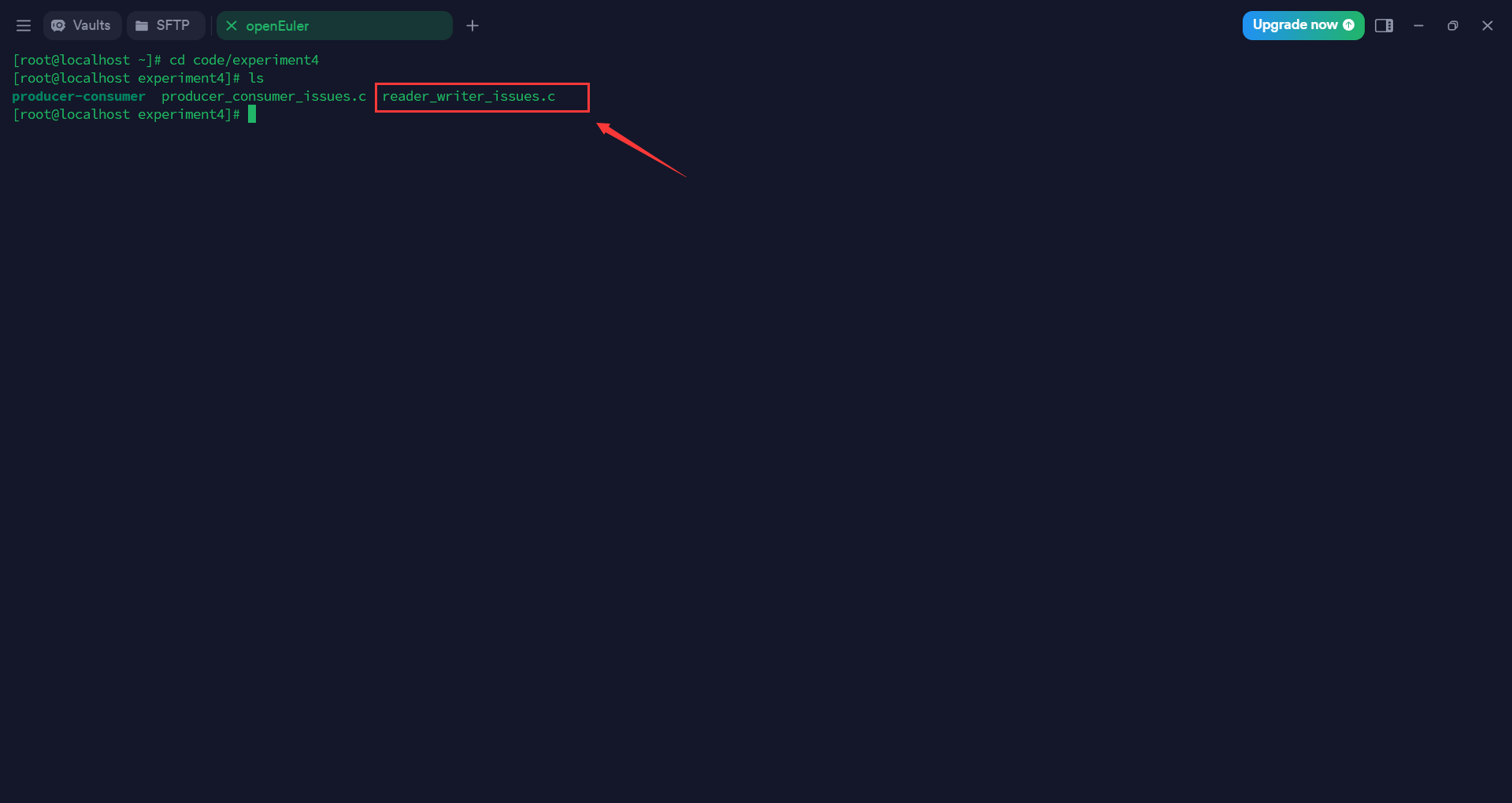
(2).定义信号量集ID、进程ID和缓冲区，创建3个信号量并初始化，再使用fork()函数创建子进程。信号量0用于控制对读者计数器的互斥访问，信号量1用于控制写者的互斥访问，信号量2记录当前正在进行读操作的读者数量。

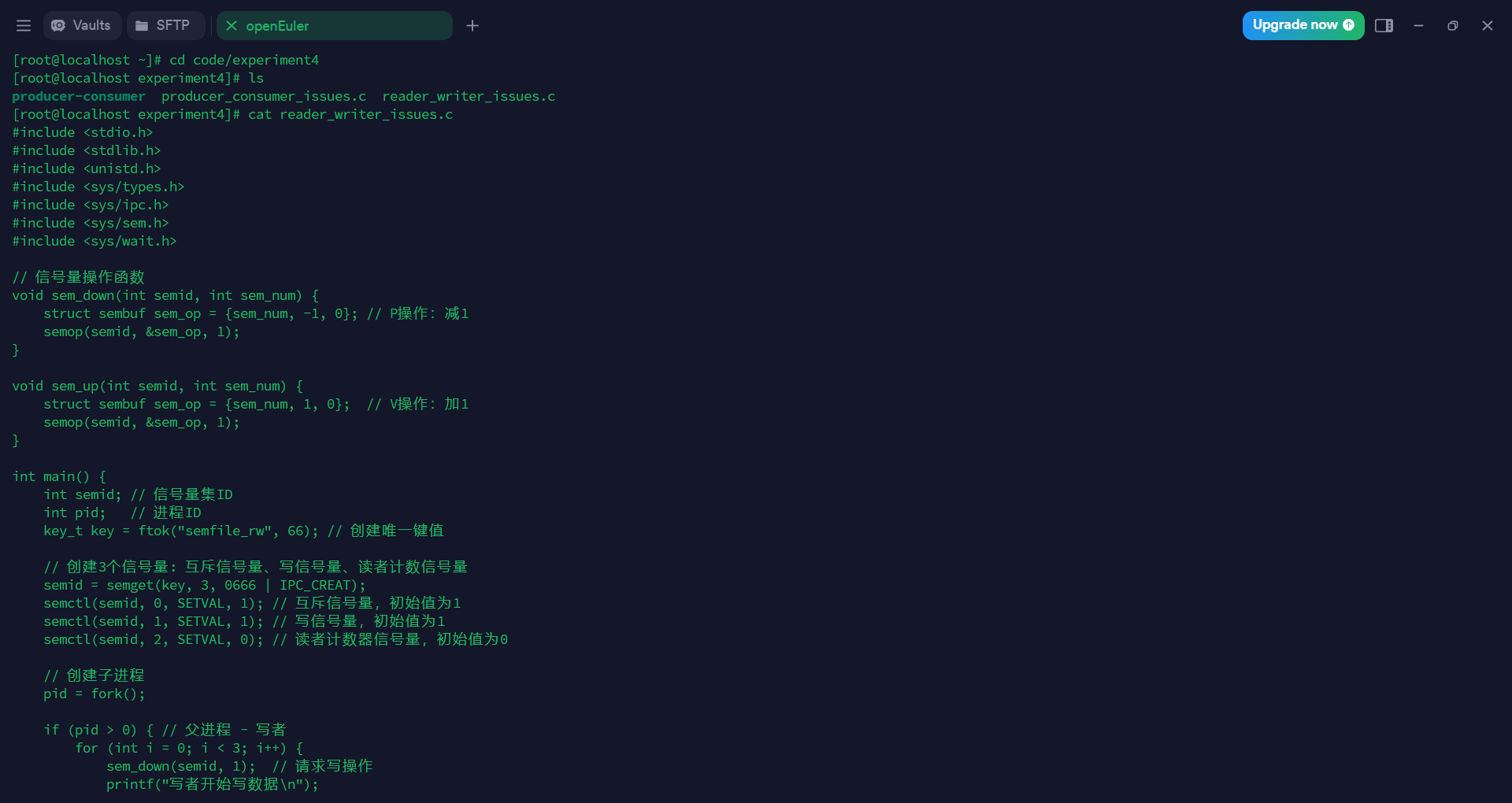


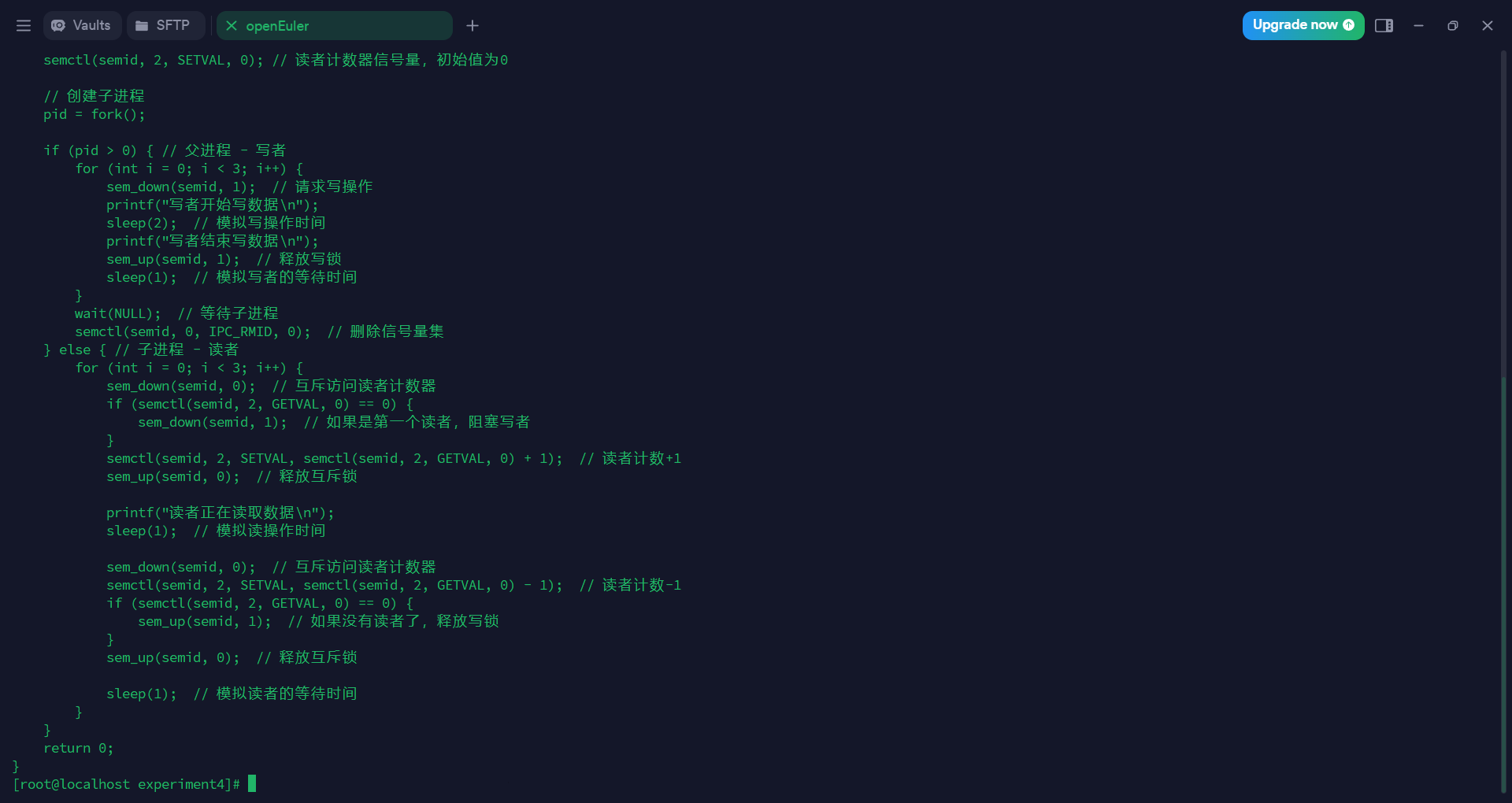
(3).父进程作为写者，之前创建的子进程作为读者。由于是读者优先，当读者在读取数据时写者不能进行写入操作。通过信号量控制读者的数量以及写者的互斥锁，确保数据的同步与并发读写的正确性。



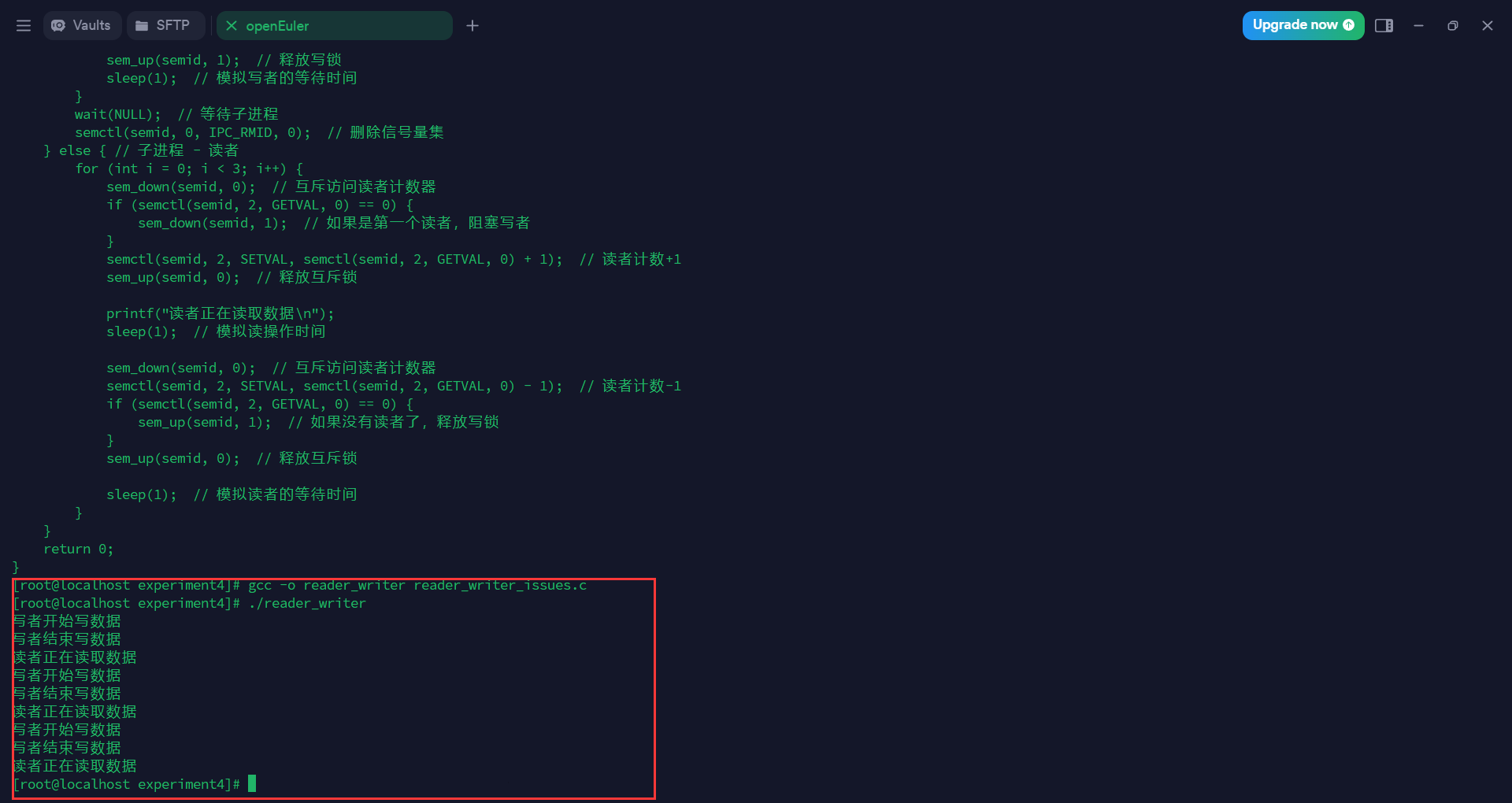
(4).将本地编写好的c语言程序复制到openEuler-Linux虚拟机中，并查看内容。

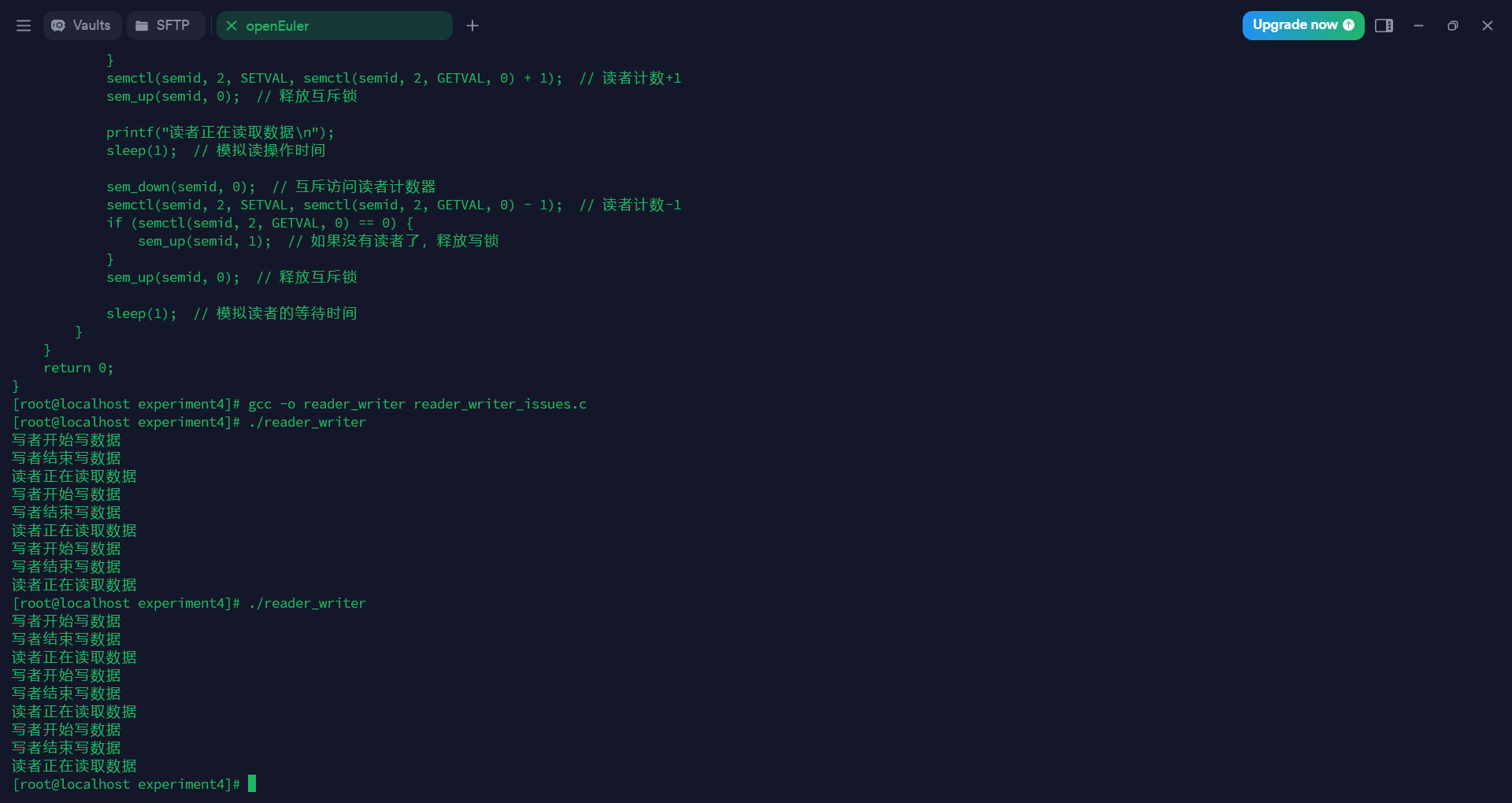






(5).使用gcc命令编译运行代码。





# 实验分析与总结

(1).**进程并发与互斥的本质**： 本实验通过信号量机制解决了生产者-消费者和读者-写者的经典并发问题，帮助理解了多进程间资源争用的本质。信号量通过对共享资源的互斥访问和同步控制，避免了进程间因同时访问同一资源而产生的不一致性。生产者-消费者问题中的互斥锁和空满信号量保障了生产和消费过程的协调，读者-写者问题中的互斥信号量和读写优先控制确保了并发读写操作的安全性。

(2).**信号量的使用与系统调用**： 信号量通过 semget()、semop() 和 semctl() 系统调用进行创建、操作和控制。本实验清楚地展示了信号量的初始化和 P/V 操作，通过操作信号量的值来控制资源的占用和释放。在生产者-消费者问题中，空位和满位信号量分别控制缓冲区的使用；而在读者-写者问题中，通过对读写信号量的控制，实现了读者优先的并发模型。

(3).**并发控制的原子性**： 通过信号量操作，可以保证进程对共享资源的操作是原子性的。每一次信号量的减（P操作）和加（V操作）都能够避免竞态条件的发生，从而确保多个进程在并发情况下不会出现资源争用导致的错误。例如，生产者-消费者问题中的缓冲区访问操作和读者-写者问题中的读写操作都得到了正确的同步控制。