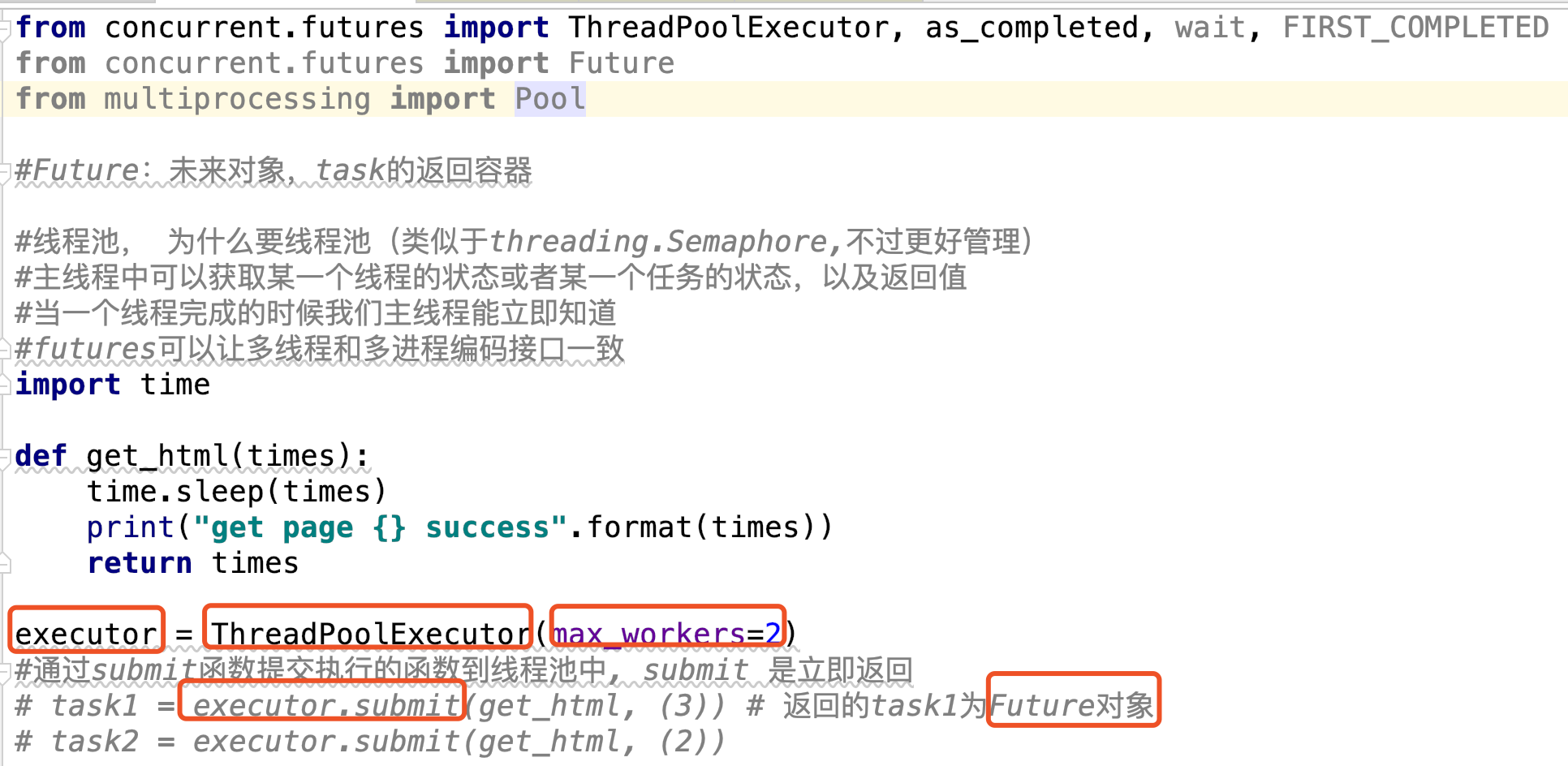
# 线程和进程

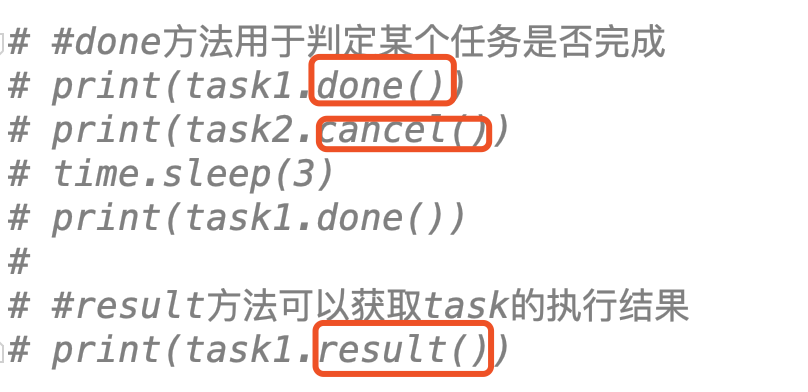
1. 多线程

Semaphore 用于控制进入数量的锁



1. 线程池



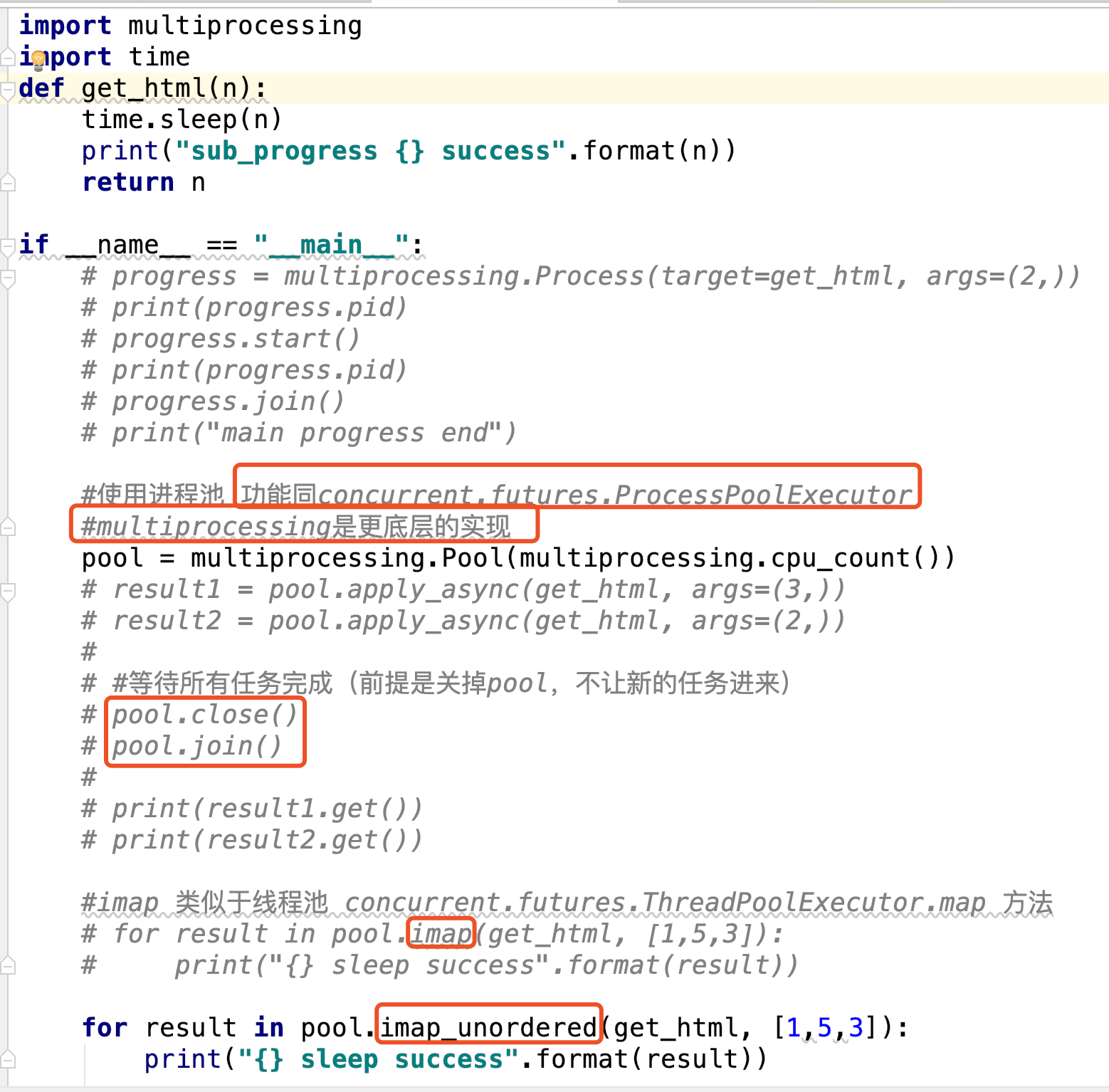




1. 多进程和多线程

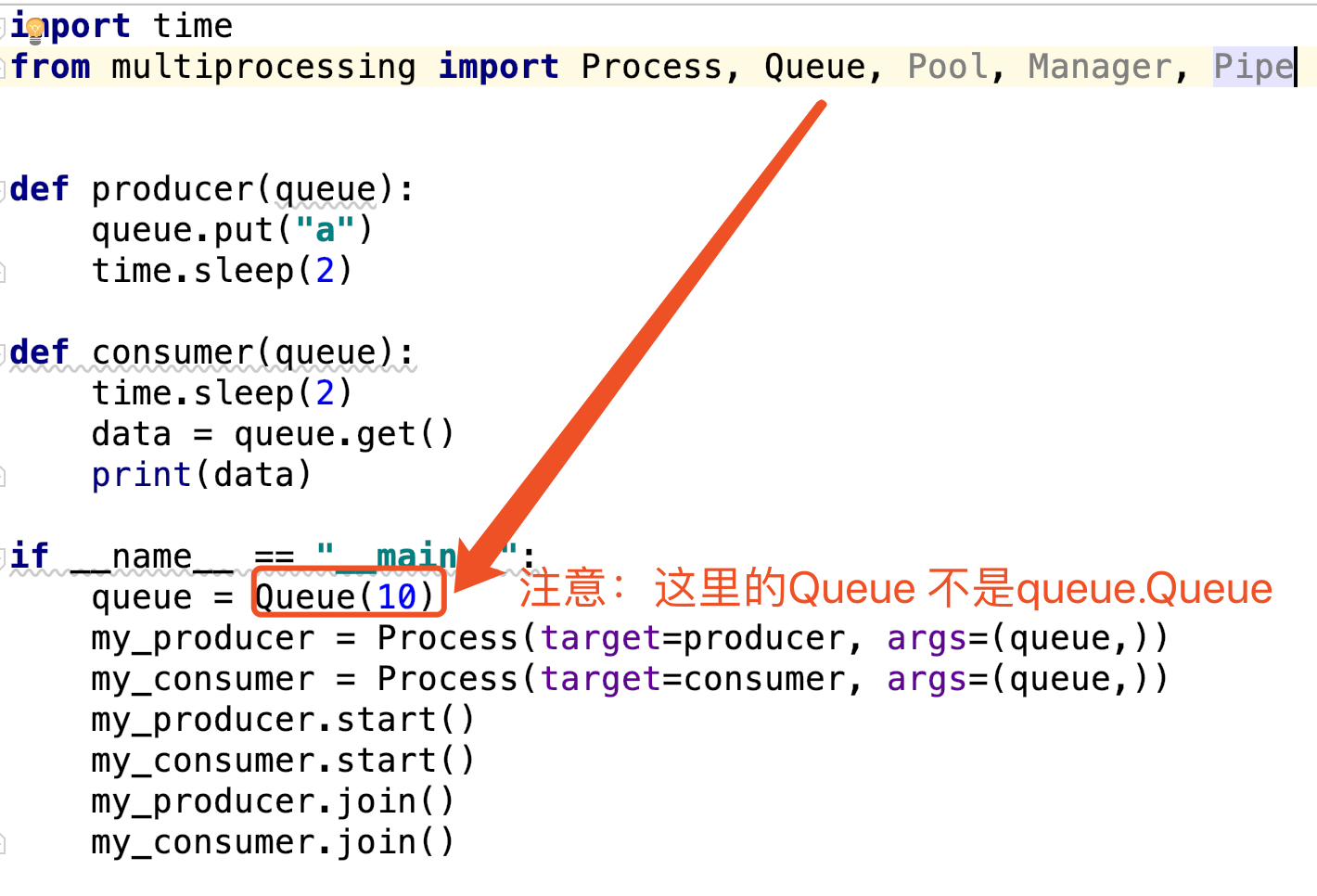


1. 对于耗费CPU的操作，多进程优于多线程
2. 对于io操作来说，多线程优于多进程
3. 多进程编程

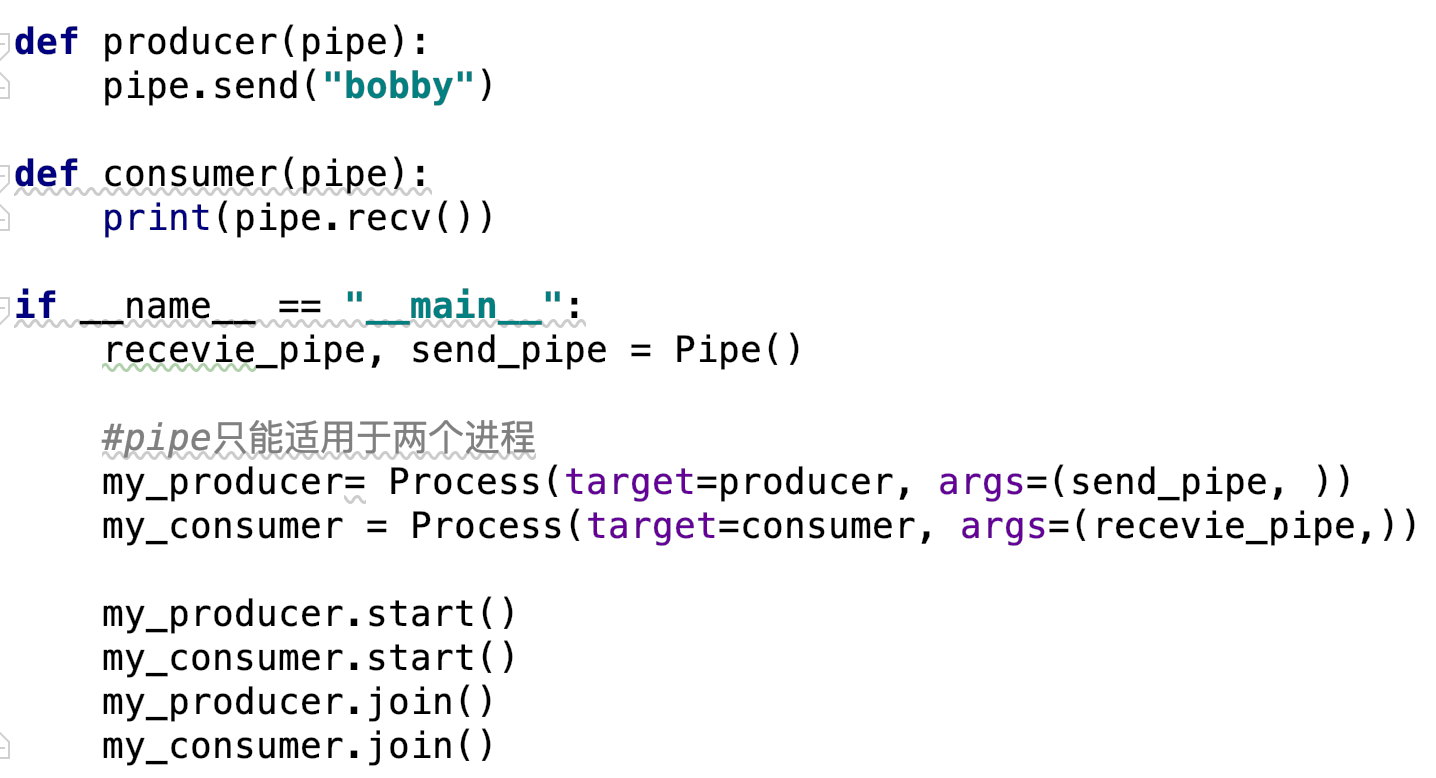


1. 进程间的通信
2. Queue通信

from queue import Queue中的Queue适用于多线程编程，但不适用于多进程编程，多进程编程中应使用multiprocessing.Queue



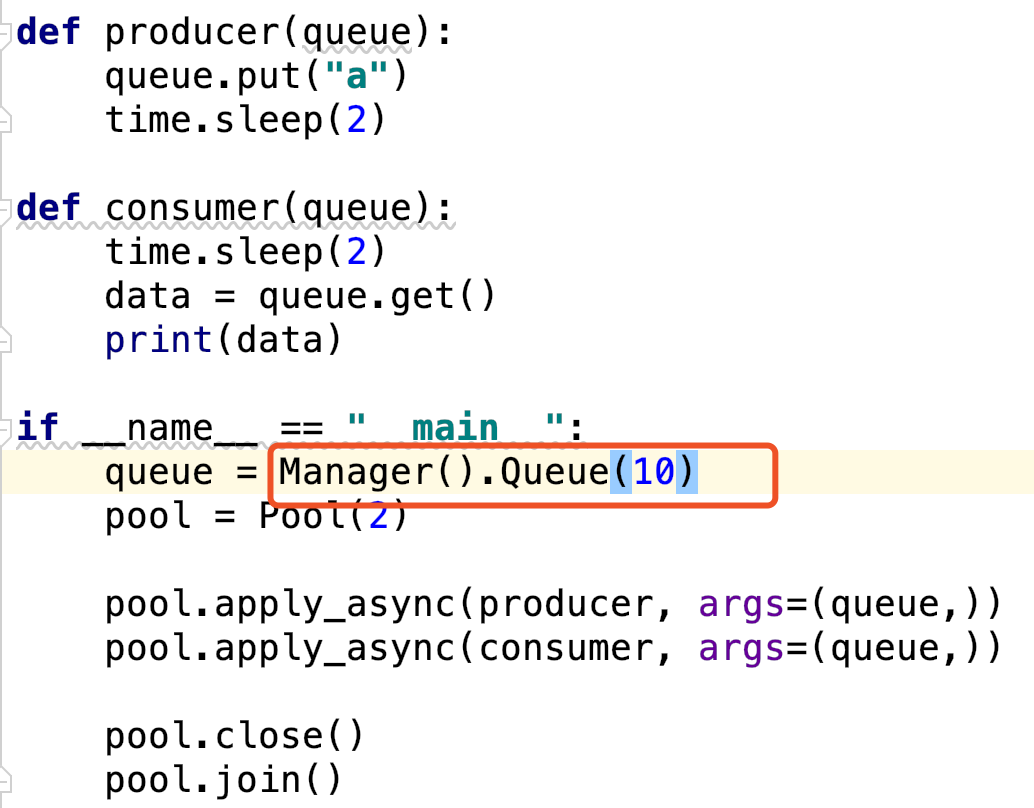
1. 通过pipe实现进程间通信  
   pipe的性能高于queue，但是pip只适用于两个进程



1. 共享全局变量通信

共享全局变量通信不能适用于多进程编程，可以适用于多线程

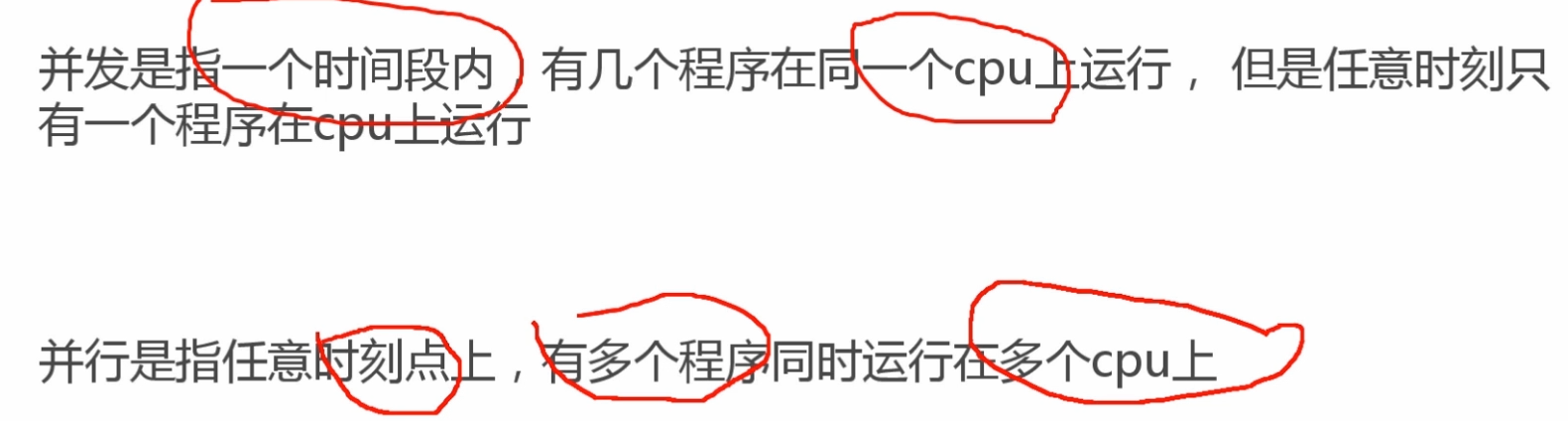
（4）multiprocessing中的Queue不能用于pool进程池  
 pool中的进程间通信需要使用manager中的Queue



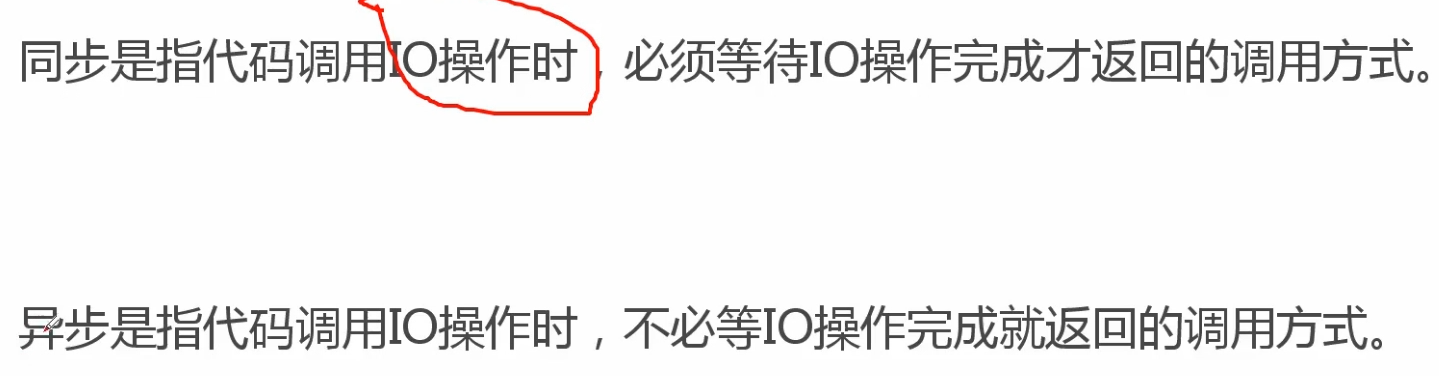
（5）用manager中的数据结构来完成进程间的内存共享（注意数据的同步）



# 2. 协程

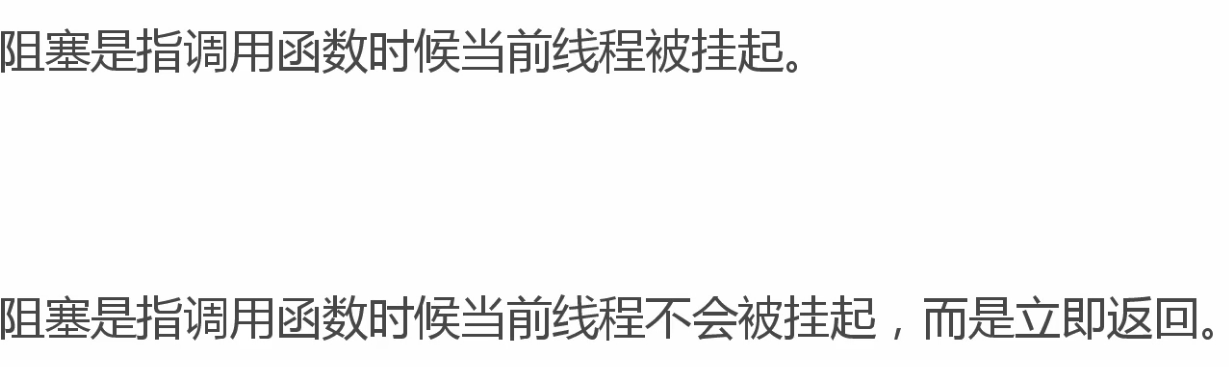


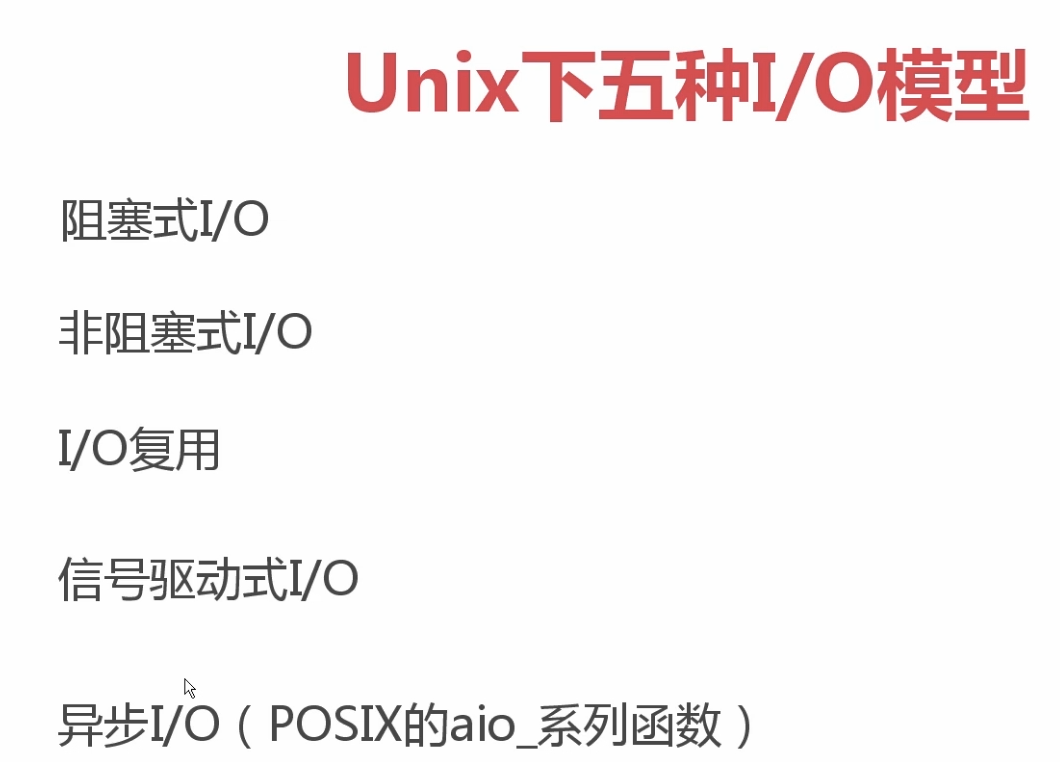
**同步和异步是消息通信的一种机制**



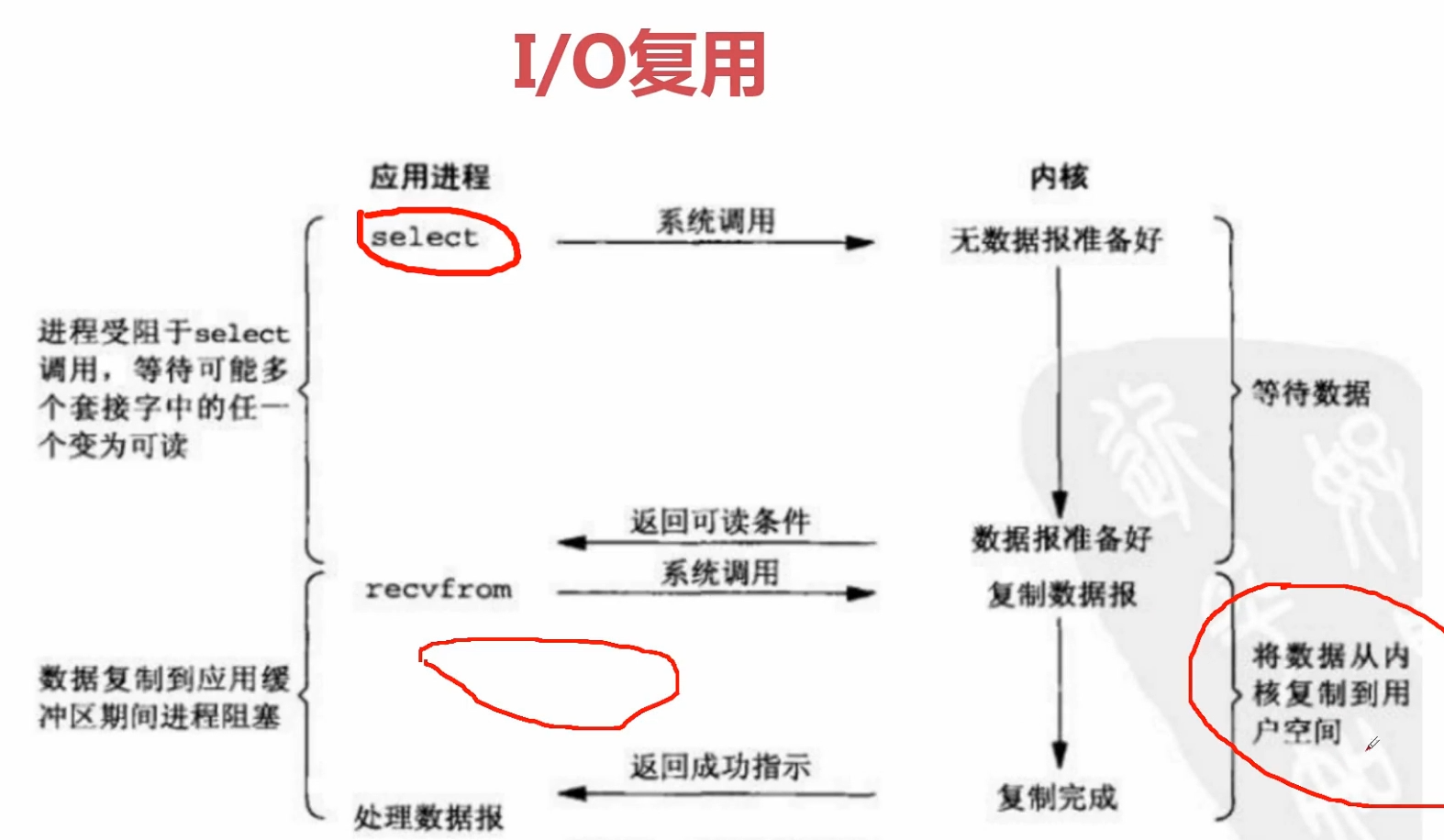
多线程即异步操作

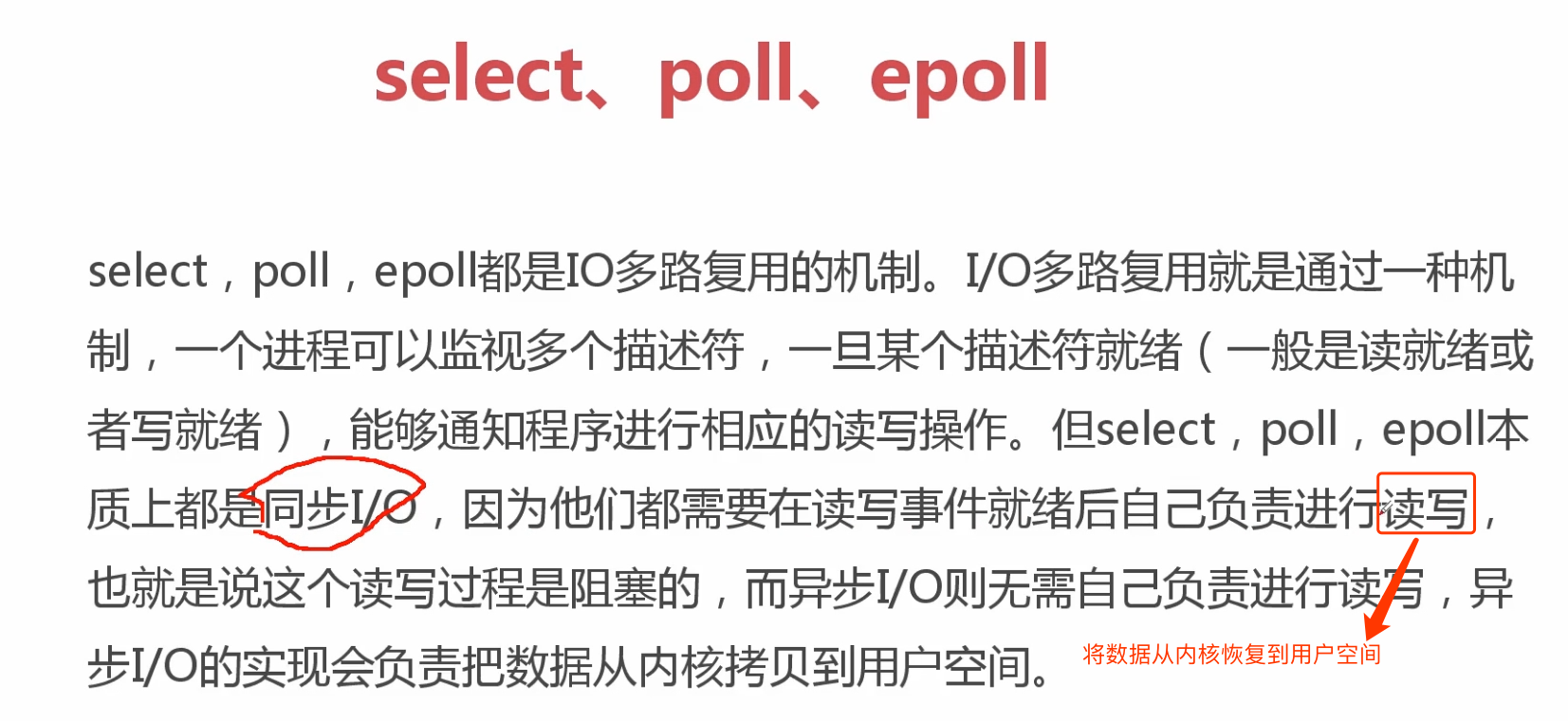
**阻塞和非阻塞是函数调用的一种机制**

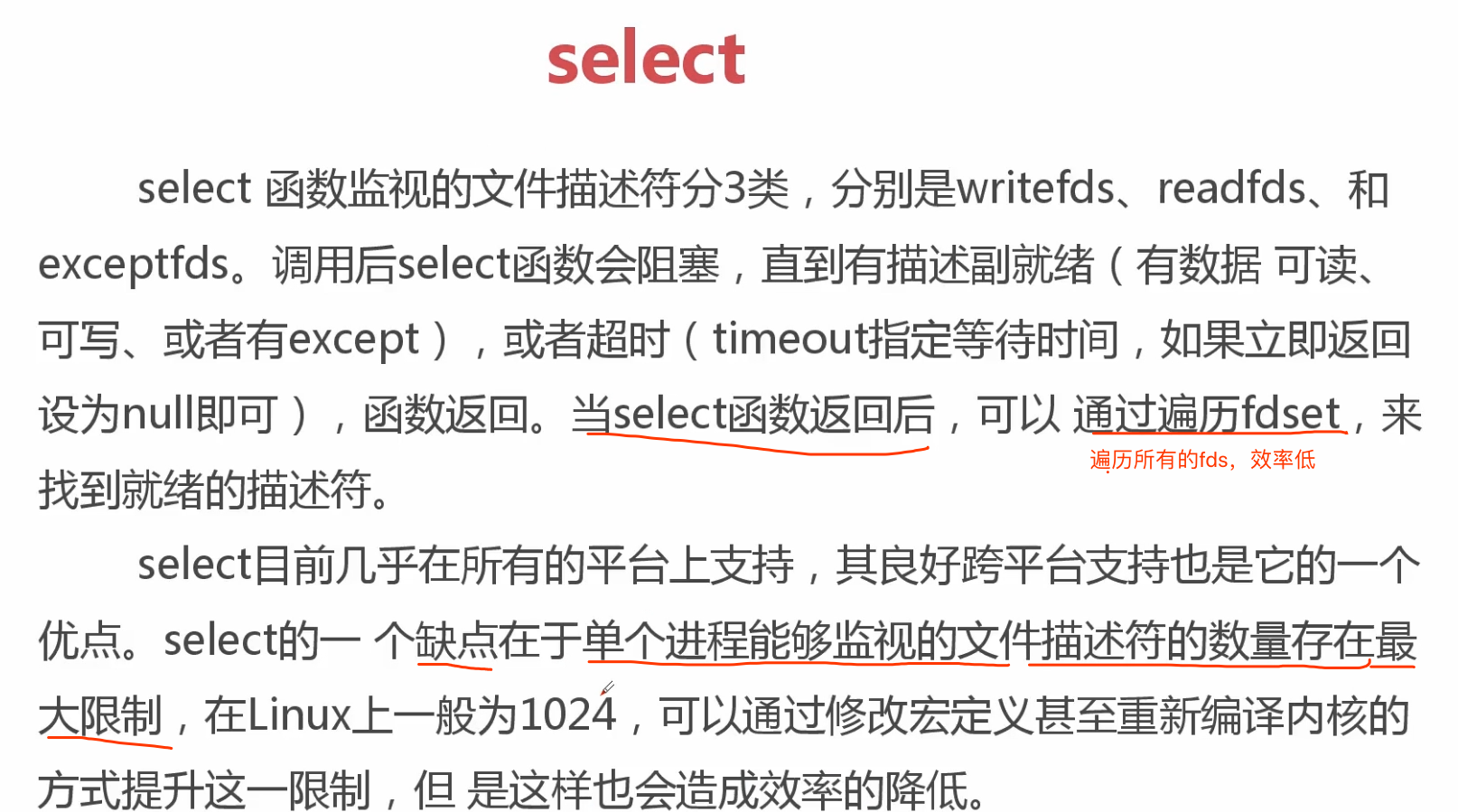


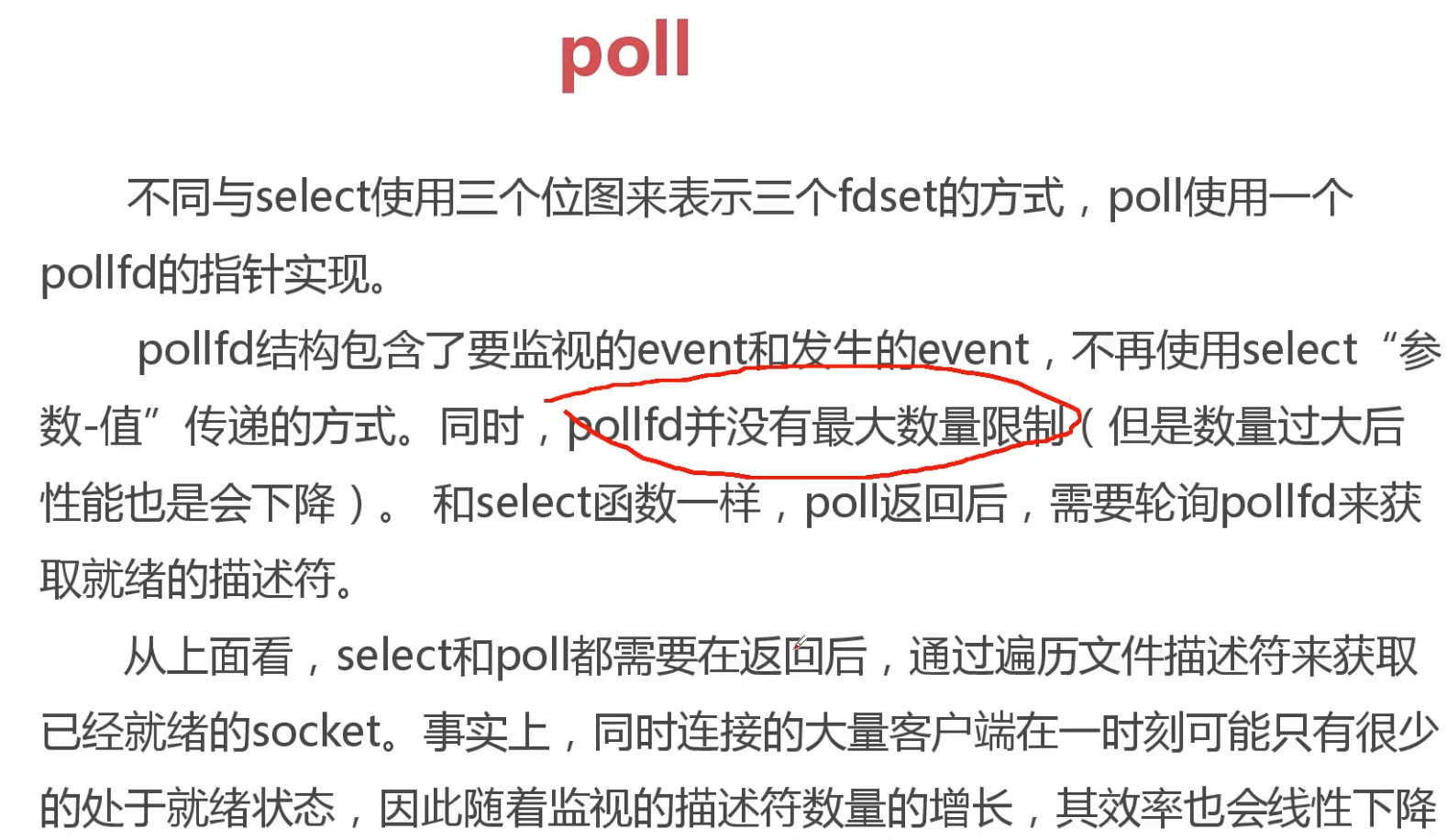


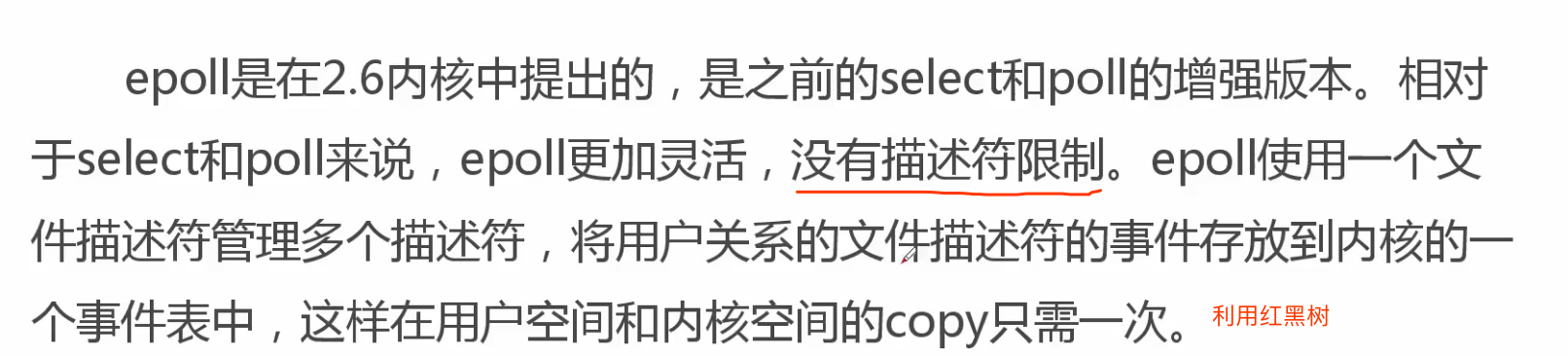
I/O复用：select可以监听多个socket的状态是否发生变化











1. epoll并不代表一定比select好  
在并发高的情况下，连接活跃度不是很高， epoll比select  
并发性不高，同时连接很活跃， select比epoll好

2. 回调+事件循环+select(poll\epoll)