

# 《进程间通信》章节练习

**1. 对于以下四个场景，请从“使用阻塞的消息传递进行进程间的直接通讯”、“使用非阻塞的消息传递进行进程间的直接通讯”、“使用信箱的方式进行进程间进行间接通讯”、“通过轮询共享内存的方式进行进程间的通讯”中选择最合适的进程间通信方法**

a) 电商网站中的反向代理进程希望通过进程间通信的方式将收到的用户请求转发给一系列服务进程，使得某服务进程空闲后即可处理该请求。

- 使用信箱的方式进行进程间进行间接通讯

b) 电商网站中的服务进程希望通过进程间通信的方式从锁服务（Lock Service）进程中获取一把锁，从而进入临界区（Critical Section）执行商品购买逻辑。

- 使用阻塞的消息传递进行进程间的直接通讯

c) 电商网站中的服务进程希望通过进程间通信的方式，将包含用户请求执行结果的网络包通过用户态网络驱动服务进程，以尽可能低的时延发送出去。

- 通过轮询共享内存的方式进行进程间的通讯

d) 电商网站中的服务进程希望通过进程间通信的方式将一条用户购买记录发送给后台推荐分析进程。

- 使用非阻塞的消息传递进行进程间的直接通讯

**2. 在xv6的管道线（PIPE）实现中，pipe这一结构体中的lock这一属性的作用是什么？为什么在sleep函数中存在放锁与拿锁操作，而在wake函数中却没有？**

- 使用锁以确保管道互斥，从而可以原子方式访问共享数据，因为一次只能有一个进程可以持有该锁
- 因为wake只需要唤醒一个进程，让他去尝试获得锁，自己还要继续向下执行，不能放锁。而自己在sleep之前要先放掉锁，并且在醒前再取得锁，因此在sleep中存在放锁与拿锁，wake中没有

**3. 对于基于共享内存的进程间通信而言，存在一个常见的安全性问题：Time-to-check to time-to-use（TOCTOU）。请查阅外部资料，简要说明这一问题的含义。**

- 代码先检查某个前置条件（例如认证），然后基于这个前置条件进行某项操作，但是在检查和操作的时间间隔内条件却可能被改变，如果代码的操作与安全相关，那么就很可能产生漏洞，这种安全问题被称做TOCTTOU

#### 4. 对于课程中所介绍的轻量级进程间通信（LIPC），请回答以下问题：

(1) 为什么要将栈分成参数栈与执行栈两种？

- 参数栈同时映射在调用者进程和被调用者进程地址空间，不需要内核额外的数据拷贝，用来传参
- 执行栈是根据调用者和被调用者临时生成的，是用来真正执行逻辑的，是真正进程运行过程中的栈

(2) LRPC中控制流转换的主要开销来自哪？

- 主要开销来自切换地址空间、更新参数栈和执行栈、更新linkage record等

(3) 不考虑多线程的情况下，共享参数栈安全吗？

- 安全，被调用者正在占用线程时，调用者不会做任何操作，被调用者返回后调用者才能继续运行，类似于顺序执行

#### 5. 在进程间通信的实现中，通常需要采用某种命名机制来确定某个进程间通信的目标进程（如xv6例子中的nread/nwrite指针）。这类命名机制的设计是否会成为影响进程间通信性能的决定性因素？为什么？

不会，命名机制对性能的影响不是决定性的，切换地址空间、更新参数栈和执行栈、更新linkage record等才是主要开销