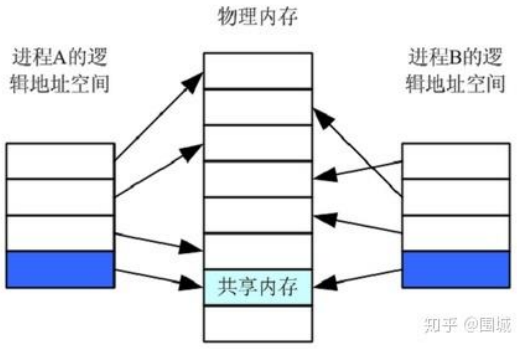
# Lecture 7 作业

## 问题：列举 5 种 IPC 的方法并阐述其原理。

1. 共享内存：

进程通过调用 shmget 想系统内核获取一个共享内存块，然后每个进程通过 shmat 将进程的逻辑虚拟地址空间指向共享内存块中。当一个进程向这个共享内存块中写入时其他所有共享的进程都可以看到写入的内容。如果共享内存的容量被占满，需要发送内容的进程就会进入等待直到有空闲空间出现。由于进程之间不能直接通信，在发送者没有发送新消息的时候，接收者会进入盲目等待状态，造成轮询，资源开销较大。




2. **消息传递通信：**

基本原理是建立一个通信连接，然后通过 **send** 和 **recv** 接口进行信息传递。

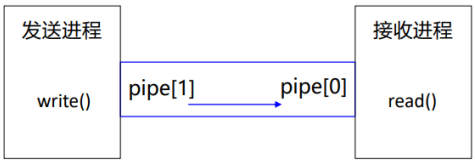
消息传递通信分为**直接通信**和**间接通信**。

直接通信下的连接建立是自动的，通过发送者和接收者之间的唯一标识；在直接连接中一个连接唯一地对应一对进程，一对进程之间也只会存在一个连接，大部分的连接是双向的。

间接通信下消息的发送和接收都需要经过一个信箱，信箱的大小决定了信箱中可以容纳的信件数，每个信箱都有自己唯一的标识。实现间接通信首先要创建一个新的信箱，比如 A 和 B 之间，A 要先通过 send 原语把信件放入信箱，然后 B 通过 receive 原语从信箱中取出这封信件，结束通信后销毁这个信箱。

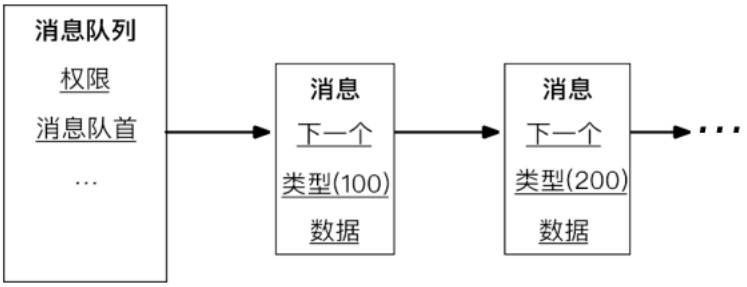
3. **管道通信：**

管道通信是 unix 等宏内核系统中非常重要的进程间通信机制。管道（pipe）连接一个发送端和一个接收端，一个管道只能有两个端口，一个负责输入，一个负责输出，传送的数据为字节流。



4. **消息队列：**

消息队列通过**链表**的方式组织消息，任何有权限的进程都可以访问队列，写入或者读取。消息队列的读写规则遵循 FIFO 先进先出原则，在队伍尾部写入，默认从队伍首部获取信息；可以通过 recv(A,type,message)按照类型进行查找，类型为 0 时返回第一个信息（首部信息），类型为某个 type 时（type 为正数）返回第一个类型为 type 的信息。




5. **信号量：**

信号量方式的基本原理类似计数器，信号量具有一个初始值，每当有进程申请使用信号量，信号量自减一，进程进入临界区工作。当信号量减到 0 的时候说明资源已经用完，此时进程需要等待进入临界区，当有进程执行完退出临界区后信号量自增一，等待的下一个进程可以开始执行。

# Lecture 8 作业

① 假设有4个进程想要互斥地使用某临界资源，该资源的数目为2，S为信号量，则S.value的值的最大值与最小值分别为多少?
② 简述发生死锁的必要条件，以及解决死锁的办法。
③ 若某一系统在某时刻的资源分配表和进程等待表如下所示，请问在该系统中存在死锁吗？如果存在，终止哪个进程解决死锁问题所付出的代价最小?

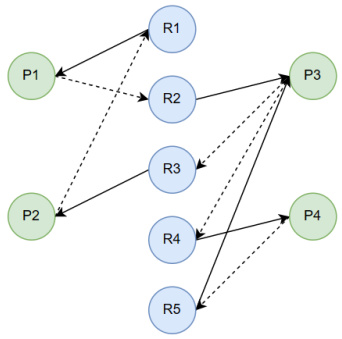
**资源分配表**

| 进程号 | 资源号 |
|---|---|
| $P_1$ | $R_1$ |
| $P_2$ | $R_3$ |
| $P_3$ | $R_2$, $R_5$ |
| $P_4$ | $R_4$ |

**进程等待表**

| 进程号 | 资源号 |
|---|---|
| $P_1$ | $R_2$ |
| $P_2$ | $R_1$ |
| $P_3$ | $R_3$, $R_4$ |
| $P_4$ | $R_5$ |

1. **答**：最大值为 2，最小值为 0
2. **答**：
   1) 必要条件：互斥访问、持有并等待、资源非抢占、循环等待
   2) 解决方法：
   a. 打破循环等待：出现问题后检测是否出现了循环等待，如果出现可以通过杀死循环中的进程打破环
   b. 预防出现死锁：
      - 避免互斥访问：通过其他手段（如服务器代理执行）
      - 不允许持有并等待：一次性申请所有资源
      - 允许资源被抢占：需要考虑如何将被抢占的进程恢复到抢占之前的状态
      - 避免循环等待：按照特定顺序获取资源
   c. 避免运行时出现死锁：使用银行家算法，所有的进程在获取资源时都需要通过管理者同意
3. **答**：




根据表格画出资源分配图，由图可以看出该系统在这个时刻存在：

（1） R1->P1->R2->P3->R3->P2->R1

（2） P3->R4->P4->R5->P3

两个环路，说明系统中死锁存在。由于 P3 存在于两个环路中，所以终止进程 P3 付出的代价最小。