# 进程通信

## 什么是进程通信

每个进程各自有不同的用户地址空间,任何一个进程的全局变量在另一个进程中都看不到。 所以进程之间要交换数据必须通过内核,在内核中开辟一块缓冲区,进程 A 把数据从用户空间拷到内核缓冲区, 进程 B 再从内核缓冲区把数据读走,内核提供的这种机制称为进程间通信。

## 进程通信的方式

1. 管道通信

管道是一种半双工的通信方式，数据只能单向流动。管道通信分为命名管道和匿名管道。

匿名只能在具有亲缘关系的进程间使用。进程的亲缘关系通常是指父子进程关系。 命名管道则可以在任何关系的进程之间通讯。 命名管道虽然可以在任意关系之间通信， 但是它会长期存在于系统之中，因此使用不当会容易对系统造成负担。

父进程创建管道，得到两个⽂件描述符指向管道的两端 父进程 fork 出子进程，⼦进程也有两个⽂件描述符指向同⼀管道。 ⽗进程可以往管道⾥写,⼦进程可以从管道⾥读,管道是⽤环形队列实现的,数据从写端流⼊从读端流出,这样就实现了进程间通信。

1. 信号及信号量通信

信号 ( sinal ) ： 信号是一种比较复杂的通信方式，用于通知接收进程某个事件已经发生。

信号量( semophore ) ： 信号量是一个计数器，可以用来控制多个进程对共享资源的访问。 它常作为一种锁机制，防止某进程正在访问共享资源时，其他进程也访问该资源。因此，主要作为进程间以及同一进程内不同线程之间的同步手段。

同样的，信号量是有限的，不能无限使用。 3. 消息队列

消息队列( message queue ) ： 消息队列是由消息的链表，存放在内核中并由消息队列标识符标识。 消息队列克服了信号传递信息少、管道只能承载无格式字节流以及缓冲区大小受限等缺点。

1. 共享内存

共享内存( shared memory ) ：共享内存就是映射一段能被其他进程所访问的内存，这段共享内存由一个进程创建，但多个进程都可以访问。 共享内存是最快的 IPC 方式，它是针对其他进程间通信方式运行效率低而专门设计的。它往往与其他通信机制，如信号量配合使用来实现进程间的同步和通信。

1. socket

套接字( socket ) ： 套接口也是一种进程间通信机制，与其他通信机制不同的是，它可用于不同机器间的进程通信。

* node进程通信的方式

Node进程间通信有4种方式：

1. 通过stdin/stdout传递：最直接的方式，适用于能够拿到“子”进程句柄的场景
2. Node原生IPC支持：node封装了IPC的具体实现，暴露了exec, fork等方法，可以用于父子进程通讯
3. 通过sockets：最通用的方式，可以跨机器通讯
4. 借助message queue：最强大的方式，但是使用起来考虑的问题是最多的