

1. pipe实现同一台主机中不同进程间的管道通信

1.1. 原理

A pipe is a section of shared memory that processes use for communication. The process that creates a pipe is the pipe server. A process that connects to a pipe is a pipe client. One process writes information to the pipe, then the other process reads the information from the pipe. This overview describes how to create, manage, and use pipes.

命名管道就是一块共享的存储区，由一个进程(pipe server)开辟管道并定义管道的相关属性，然后等待另外一个或多个进程通过提前商量好的PipeName连接上来（类似于一个应用层的协议），然后就可以将管道作为数据中转，实现通信。

管道有三种工作模式，主要体现在pipe server对管道的权限上，包括server只读、只写、可读可写三种，客户端在连接上管道时，需要指定自己的工作模式，是只读，是只写，还是可读可写。

1.2. 本程序过程

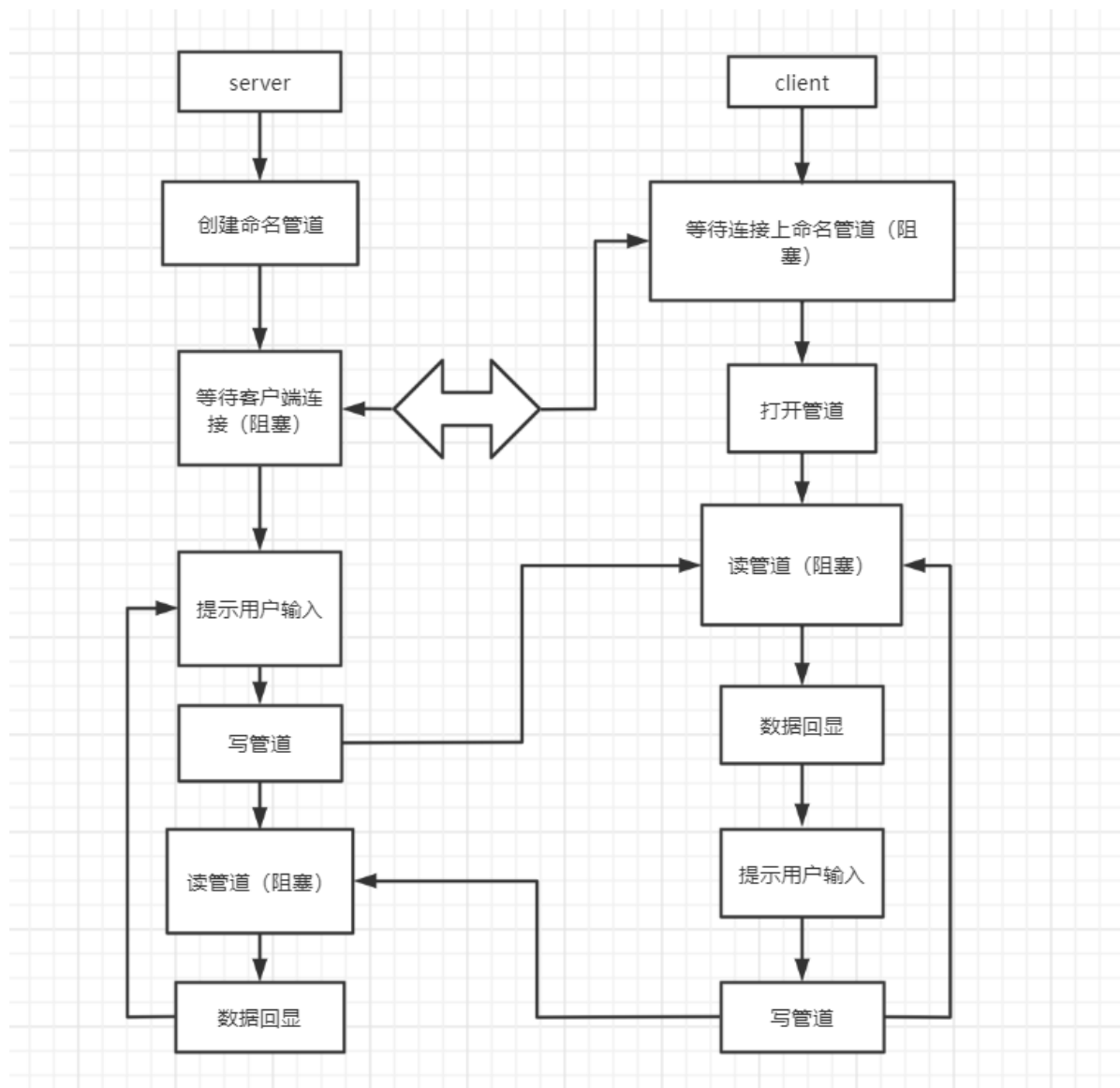
本程序首先让pipe server创建一个管道，然后用 `ConnectNamedPipe` 阻塞自己，直到有进程来连接该管道，管道设置为对pipe server可读可写。

然后让pipe client用 `CreateFile` 连接上这个管道，设置为对client可读可写。

pipe client连接上来后，pipe server由阻塞态进入就绪态，然后运行，提示用户进行输入，然后将用户输入发送给pipe client。

pipe client连接上来后，就调用`ReadFile`，也是阻塞自己，直到从管道里面获取数据，该函数才返回，client进程就进入就绪态。之后pipe client打印出pipe server发送的消息，提示用户输入，然后将消息送入管道，完成对服务器的响应。

同时，pipe server发送完后，调用`ReadFile`，也阻塞了自己，直到管道中有了新的数据，然后打印并回显。



1.3. 感悟

- 通过这几个api阻塞的函数，进一步加深了对进程三基本状态的理解
- 用管道实现了同一主机上不同进程间的通信