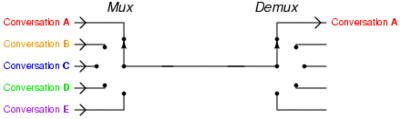
<https://www.cnblogs.com/wajika/p/6581104.html>

## [IO 多路复用是什么意思？](http://www.cnblogs.com/wajika/p/6581104.html)

在同一个线程里面， 通过拨开关的方式，来同时传输多个I/O流， (学过EE的人现在可以站出来义正严辞说这个叫“时分复用”了）。

什么，你还没有搞懂“一个请求到来了，nginx使用epoll接收请求的过程是怎样的”， 多看看这个图就了解了。提醒下，ngnix会有很多链接进来， epoll会把他们都监视起来，然后像拨开关一样，谁有数据就拨向谁，然后调用相应的代码处理。



要弄清问题 先要知道问题的出现原因  
原因:  
由于进程的执行过程是线性的(也就是顺序执行),当我们调用低速系统I/O(read,write,accept等等),进程可能阻塞,此时进程就阻塞在这个调用上,不能执行其他操作.阻塞很正常. 接下来考虑这么一个问题:一个服务器进程和一个客户端进程通信,服务器端read(sockfd1,bud,bufsize),此时客户端进程没有发送数据,那么read(阻塞调用)将阻塞直到客户端调用write(sockfd,but,size)发来数据. 在一个客户和服务器通信时这没什么问题,当多个客户与服务器通信时,若服务器阻塞于其中一个客户sockfd1,当另一个客户的数据到达套接字sockfd2时,服务器不能处理,仍然阻塞在read(sockfd1,...)上;此时问题就出现了,不能及时处理另一个客户的服务,咋么办?   I/O多路复用来解决!  
I/O多路复用:  
继续上面的问题,有多个客户连接,sockfd1,sockfd2,sockfd3..sockfdn同时监听这n个客户,当其中有一个发来消息时就从select的阻塞中返回,然后就调用read读取收到消息的sockfd,然后又循环回select阻塞;这样就不会因为阻塞在其中一个上而不能处理另一个客户的消息

具体参考 https://www.zhihu.com/question/32163005