非阻塞I/O

造成非阻塞I/O的四种情况：

（1）读操作。Read,readv ,recv,revfrom,recvmsg.

对于tcp 和 udp ，其socket在默认情况下都是阻塞的，当socket需要从接收缓冲区读取数据时，若缓冲区为空，则套接字会阻塞。唤醒TCP套接字的方式是有1个byte的数据到达，唤醒UDP套接字的方式是1个完整的数据报到达。

而对于非阻塞的套接字，若输入条件不能被立刻满足，会返回一个EWOULDBLOCK错误。

（2）写操作。这里TCP套接字和UDP不太一样。因为UDP套接字无真正意义上的发送缓冲区。

对于TCP，当发送缓冲区满时，socket就会阻塞；对于非阻塞的socket，会立刻返回一个EWOULDBLOCK错误。

对于阻塞的UDP套接字,由于没有发送缓冲区，不会因为和TCP相同的原因阻塞，但会因为其他的原因阻塞？

（3）接受了外来的链接。具体来说是，TCP的accept函数。默认情况为阻塞的状态。无连接到达，就阻塞。

但是对于非阻塞的套接字，若不能立刻接收则会返回EWOULDBLOCK错误。

（4）向外发送连接请求。

具体来说是connect函数（这个是对TCP来说的，虽然UDP也有这个函数，但是具体的含义不一样，UDP只是将告知内核保存对端的IP和端口号~）。Connect函数发送TCP的三路握手，必须要接收到自己的SYN和ACK分节才可以。所以说一般情况下需要一个RTT的时间才可以被唤醒。

但是对于非阻塞的套接字来说，仍然存在可以立刻建立连接的情况那就是客户端和服务器处于同一个主机上面，若不能立刻建立连接就会返回EINPROGRESS错误（这个错误不同于上述三个~）。

关于非阻塞IO好处的一个具体的例子。客户端从标准输入读取一行文本，使用writen向套接字发送，但是当时套接字的发送缓冲区已经满了，故writen会阻塞；但是于此同时，套接字上的接收缓冲区有数据可供读取，但由于socket 是阻塞的转台只能等待~。