链滴

解决了一个困惑很久的 bug

作者：localvar

原文链接：https://ld246.com/article/1522145083197

来源网站：链滴

让这个bug困扰了很久，前一段太忙只找了个临时解决方案而没有追究原因，今天终于把它搞清楚了
由于测试时只在多CPU系统上出现，我甚至一度怀疑它是CPU的bug
oy 。

两个c/s结构的网络通讯程序，服务器端使用完成端口模型，客户端使用阻塞模型，双方以一种客户
发送命令，服务器端处理，然后返回应答的方式通讯。问题出在服务器端。以下是服务器端代码的大
处理逻辑：

```
long volatile g_busy = 0;
void iocp_thread()
{
   while( GetQueuedCompletionStatus() )
   {
      if( InterlockedCompareExchange( &g_busy, 1, 0 ) != 0 )
         WSASend( "服务器忙" );
      // 处理命令
      ProcessCommand();
      WSASend( "应答信息" );
      InterlockedExchange( &g_busy, 0 );
   }
}
```

其中ProcessCommand需要互斥运行（这是简化的逻辑，实际上有很多不同的命令，有些需要互斥
有些可以并行，否则就没必要用完成端口了），并且需要一定的时间才能处理完毕。为了避免多个客
端同时执行命令，导致所有的iocp线程都等在那，我把g_busy当成了一个锁，第一个线程可以成功
入，其它的都直接向客户端返回“服务器忙”。

程序一直都运行的很好，直到有一天把服务器程序装到了一台双核的机器上。我发现，如果让客户端
续发送命令，即收到上一条命令的应答后立即发送下一条命令，就会随机的返回“服务器忙”，而这
只有一个客户端连接上去，按照我设想的逻辑是不可能出这种情况的。检查了半天代码，没觉得有什
问题，调试吧，又遇到了另一个难题，海森堡的测不准原理起作用了，做的工作太多问题就消失了，
的太少又得不到什么有价值的信息。搞得我很是头疼。

今天再次看这个问题，突然想到：它肯定和线程切换相关，所以我应该记录下每次处理命令的线程的I
，这样出错时就可以看看上次成功执行命令的那个线程在干什么了。方法正确了，问题也就迎刃而解
，我发现，出问题时，上一个线程的WSASend居然还没有返回，也就是说，客户端已经收到应答并
送了下一条命令，服务器端也收到了命令并准备处理，但上一条命令的应答却还没有完全发送完成，
怪出错了！

总结经验教训，感觉自己一开始被两点给误导了，一是实际程序中的WSARecv/WSASend藏的比较
，没这么明显，所以没注意到。二是当时粗略检查代码觉得没问题，就把主要精力放在ProcessComm
nd上了，由于我把它里面一段访问数据库的代码注释掉以后，问题就不出了，所以还看了半天ATL OL
DB的源码，最后精疲力尽，其它事情又比较多就放弃了。