基础题

1. recv的第四个参数MSG\_PEEK有什么作用？MSG\_DONTWAIT有什么作用？

MSG\_PEEK：若加上MSG\_PEEK，则Recv函数读取时只会看这次数据一眼（只读不取），而要等到下一次再recv不加MSG\_PEEK参数时才会真正读取。

MSG\_DONTWAIT：参数会让本次recv函数实现非阻塞的功能，他与设置fd非阻塞的区别在于、前者只是一次性的设置，而后者可以实现进程操作期间的永久非阻塞设置。

1. epoll和select相比有什么优势？是如何实现的？
   1. Epoll没有select对fd的限制（select最大1024个fd，epoll的1G内存可以监控10万fd）
   2. Epoll每次只需要检查链表中的就绪结点，若有则wait拷贝句柄信息并返回，而不像select那样线性遍历所有描述符
   3. 添加描述符时，只需把fd挂在红黑树上。内核会一直记录这个描述符，即一次注册，永久保留

3、什么是epoll的边沿触发？什么是水平触发？

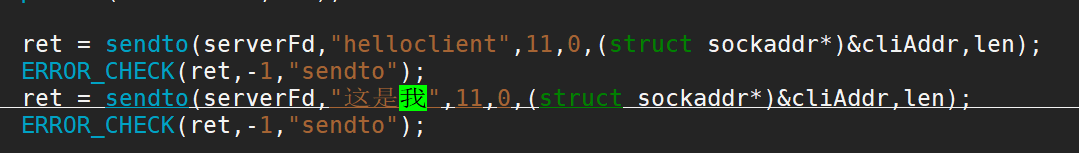
水平触发：关心缓冲区有无数据，有数据时，epoll\_wait会一直被触发。Select和epoll默认都是水平触发

边沿触发：关心的是新数据的到达，每次有新数据到达时，才会触发epoll\_wait()

-----------------------------------------

编程题

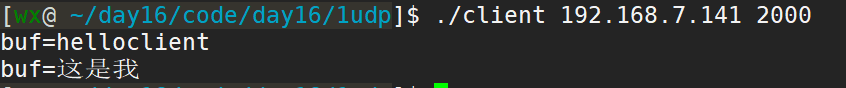
1. 用UDP实现一下客户端服务器各自收发一次消息，对比一下tcp，UDP多次发出去的消息可以被一次接收全部接收到吗？为什么？

不可以被全部收到，而只会收到第一次的消息。原因在于send和recv都会在没有对方成对出现时发生阻塞。这里出现在第二次sendto时发生了阻塞。

效果：

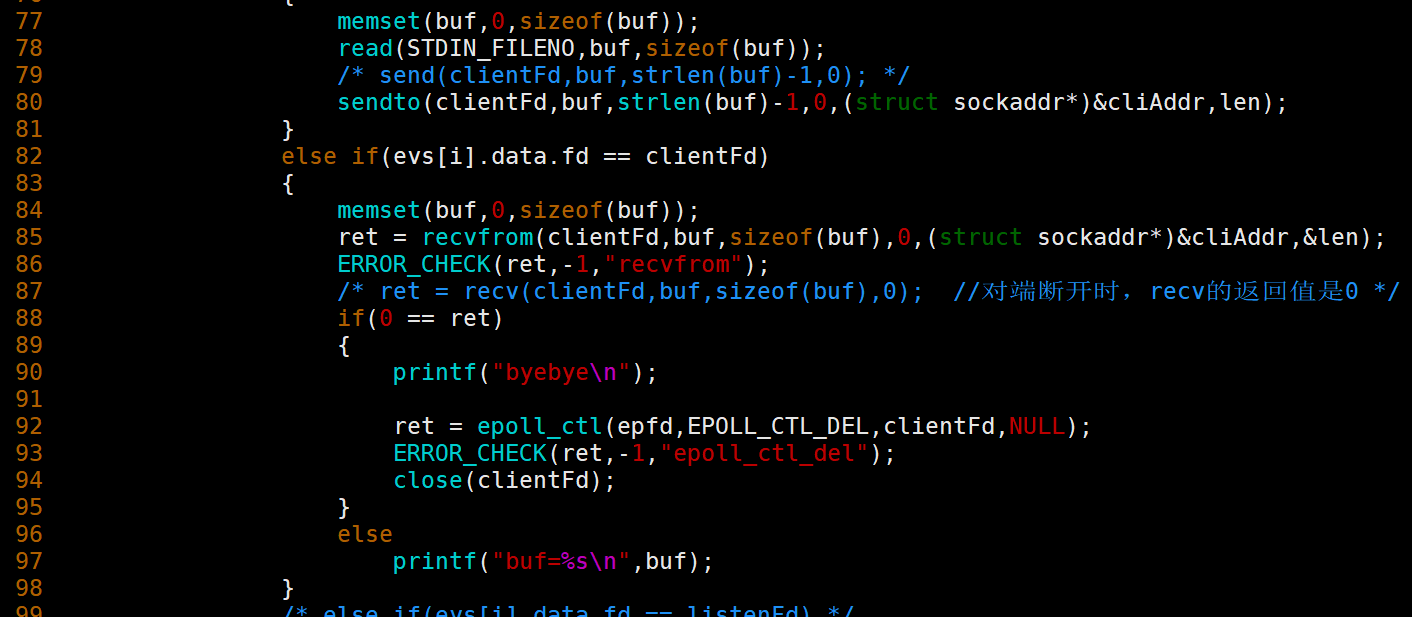
（第二次打印没有成功）

当修改client。C也让recvfrom接受并打印buf两次时，才正常打印

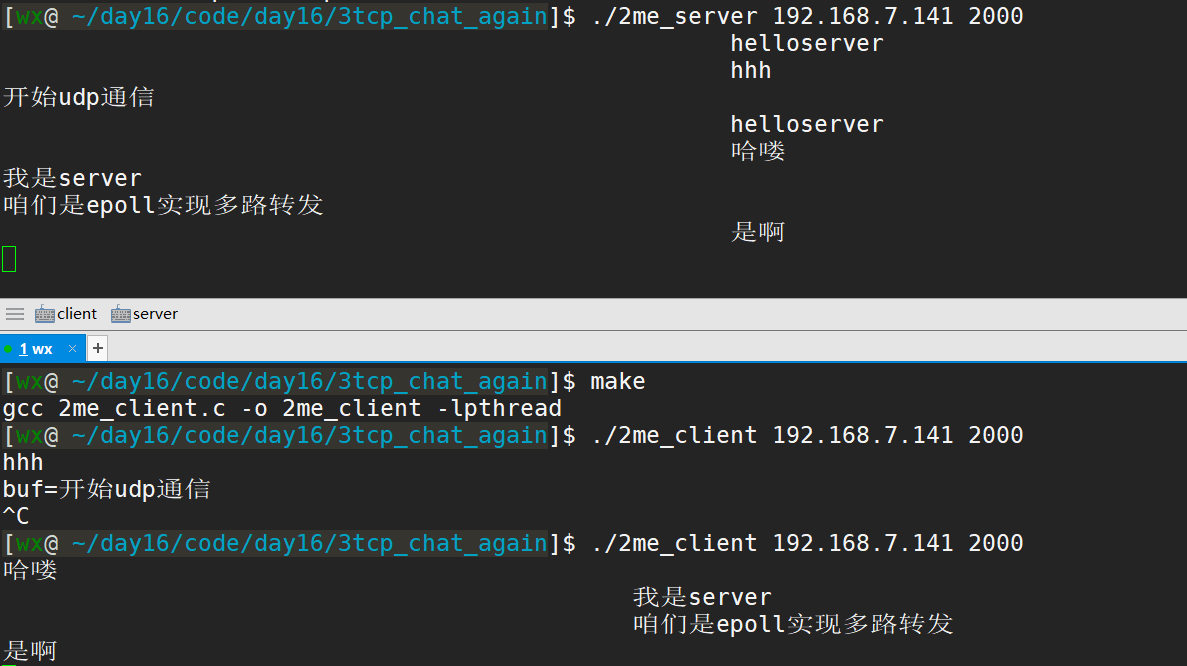


完整程序如下：

1. 实现epoll+UDP的聊天程序。



效果如下：



3、通过epoll的边沿触发模式实现即时聊天。注意用非阻塞模式，客户端断开后，能够再次连接。

（注意，把非阻塞和边沿触发都理解之后再来完成这道题目，如果非阻塞不理解就先写一个程序理解非阻塞的效果，如果边沿触发不理解，

就写用边沿触发感受一下触发的时机）

