1. tcp为什么三次握手？为什么不能两次？

防止已经失效的链接从新到达服务器。假如采用两次握手，当连接已经完成时，客户端先前的一个syn到达服务端，服务端会产生一个ack到达客户端，此时客户端是关闭的，会丢弃到该ack，而服务端认为连接已经建立，等待客户端传输数据，造成资源浪费。

1. Tcp为什么可靠？

三次握手保证连接是双工的，采用重传机制保证可靠性。

每一端维护两个缓存，一个用来接收数据，一个用来发送数据。

1. 发送缓存：保存将要发送的数据和未收到确认得到已发送数据。
2. 接收缓存：按序到达的但未被应用程序取走的数据；未按序到达的数据。
3. Tcp报文格式？

源端口|目的端口|序号|确认号|标志位（urg，ack，psh，rst，syn，fin）

1. 三次握手？

Syn\_send状态：Syn=1，seq=随机数x

Ack=1，Syn=1，ack=x+1，seq=随机数y ：SYN\_RESV状态

ESTABLISHED状态：Ack=1，ack=y+1 ：ESTABLISHED状态

1. SYN攻击？

短期内伪造ip并发送大量的SYN包，使服务器端大量连接处于SYN\_RESV状态，等待客户端的确认，由于ip并不存在，所以服务端不断重传直到超时。这些半连接将耗费大量的服务器资源，是的正常连接无法得到服务。

1. 四次挥手？

FIN\_WAIT1状态：发送FIN，seq=x包给服务器

服务器进入CLOSE\_WAIT状态

发送ACK=1，ack=x+1

FIN\_WAIT2状态：

发送FIN ，seq=y 服务器进入LAST\_ACK状态

TIME\_WAIT（2MSL）状态：

发送ACK=1,ack=y+1 服务器到达CLOSED状态

1. 为什么需要TIME\_WAIT状态？
2. 保证安全释放连接：如果在此期间没有收到服务端重传的FIN包，就认为服务端已经收到最后一次的ack包，否则，重传Ack包
3. 保证不影响下一次链接：已失效的包在此期间到达的全部简单丢弃，保证下一次链接不会受到这些包的影响。
4. 为什么建立连接三次握手，关闭需要4次挥手？

Syn对应Fin。因为三次握手将ack和syn放到一次传输上。但是释放连接却不能，因为FIN的含义只是表明自己不再发送数据包，但是可以接收数据包，客户端和服务端不需要同时关闭发送。因此一般将ack和fin分开发送。

1. Socket
   1. 客户端：
      1. Socket sockclient = Socket(AF\_INET,SOCK\_STREAM,0);
      2. Connect(sockclient , (SOCKADDR\*)&addrserver,sizeof(addrserver));
   2. 服务端：
      1. Socket sockserver = Socket(AF\_INET,SOCK\_STREAM,0);
      2. Bind(sockserver,(SOCKADDR\*)&addrserver,sizeof(addrserver))
      3. Listen(sockserver,backlogs);//backlog监听数目
      4. Sockclient = Accept(sockserver,（SOCKADDR\*）addrclient,&len);

epoll边沿触发怎么保证数据缓存中的数据被读取？