1 **中断信号的处理**

对于某些信号，比如SIGCHLD，必须处理，以避免已终止子进程成为僵尸进程。通常为了程序的可移植性，在信号处理函数返回后，都假定系统调用将不会自动重启，而是返回错误并置errno为EINTR。因此，我们必须判断errno为EINTR的情况，并手动重启被信号中断的系统调用。

2 **connect**：

* connect调用失败后，相应的套接字将不再可用，必须关闭，**我们不能对这样的套接字再次调用connect函数**。
* 对于一个正常的阻塞式套接字，如果其上的connect调用在TCP三次握手完成前被中断（**例如收到了某个信号**），如果内核不会自动重启被中断的connect，那么connect调用将返回EINTR，在这种情况下，我们**不能再次调用connect等待未完成的连接继续完成。这样做将导致返回EADDRINUSE错误**。

这种情形下我们**只能调用select**，当连接建立成功时select返回套接字可写条件，当连接建立失败时select返回套接字即可读又可写条件。

3 **连接异常的处理**：

* **服务器正常关闭连接**：即服务器进程调用shutdown关闭输出端，此时**服务器端TCP首先将发送缓冲区的数据都发送完毕后**，在发送FIN给客户端。客户端收到FIN（read返回0）后，知道服务器已经关闭连接，此时read已经读不到数据了，因为服务器已经停止发送数据了。但是write还是可以的，也就是说服务器还可以接收客户端发送的数据。这是因为TCP是全双工的，shutdown只是关闭一个方向的数据流动而已。但如果服务器是调用close，那么两个方向的数据流都将关闭，客户端write会出错。
* **服务器进程终止**：服务器进程关闭后，将发送FIN给客户端，需要保证客户端能够及时的获取到服务器关闭连接的消息（即read返回0）。这就要求客户端在某些场景下必须使用IO多路复用技术（select、epoll）。当客户端read返回0后，如果继续尝试write发送数据（理论上是可以的），这时将会：
* **第一次write**：客户端第一次write的数据将发送给服务器，服务器端TCP收到后检测到对应端口的服务器进程已终止，因此给客户端回RST。此时write出错并设置errno为ECONNRESET并且**清空TCP发送和接收缓冲区**。
* **第二次write**：客户端内核收到第二次write请求后，并不会尝试将数据发送到服务器，而是会发送一个SIGPIPE信号给客户端，该信号默认行为为终止进程。如果客户端忽略该信号，那么write调用返回错误并设置errno为EPIPE。
* **accept返回前连接中止**：此时accept将返回错误，并设置errno为ECONNABORTED
* **服务器主机崩溃**：客户端发送数据超时（收不到服务端TCP的ack），write返回错误并设置errno为ETIMEOUT。
* **服务器主机崩溃后重启**：服务器重启时会清空所有的tcp连接信息，重启后如果客户端write发送数据，服务器tcp收到后认为该连接报文是无效的，因此回复RST给客户端，客户端write返回错误并设置errno为ECONNRESET。
* **服务器主机关机**：服务器关机时，init进程会发信号SIGTERM和SIGKILL来杀掉每一个进程，此后进程正常终止，流程同上。

4 **读写0字节**

* **读0字节**：int n = read(fd, buf, 0); 直接返回0，并且不会置errno。
* **写0字节**：int n = write(fd,buf,0); 直接返回0；write调用也就只有这一种情况会返回0；

5 **IO复用的可读可写触发条件**：

* **可读触发**：当关注的套接字描述符出错时
* **可写触发**：**非阻塞的connect的套接字已建立连接**或出错；关注的套接字有错误待处理时

6 **UDP套接字不存在真正的发送缓冲区**，内核只是复制应用进程数据并把它沿着协议栈向下传递，渐次冠以UDP首部和IP首部。因此对一个UDP套接字（默认设置），输出函数调用将不会因为与TCP套接字一样的原因而阻塞，不过有可能会因为其他的原因而阻塞。

另一方面，**UDP套接字有接收缓冲区**，因此对于接收函数，其阻塞机制类似于TCP接收函数。

7 **阻塞式accept**和**select**结合使用时在如下所示的业务场景下将出现问题：

1. 客户建立一个连接后，立马终止它，由于客户端打开了SO\_LINGER选项，并且设置逗留时间为0，这导致客户端将不会发送正常的FIN序列来结束TCP连接，而是**直接发送RST给服务器**
2. 当select返回监听套接字可读时，服务器由于业务繁忙，还没有来得及调用accept获取已连接套接字，此时**服务器TCP协议栈收到客户端发送的RST，服务器会直接将监听套接字描述符从已就绪描述符集中剔除掉**。之后，服务器再调用accept时将可能会阻塞在此，直到一个新的连接到达。

为了解决上面这个问题，我们可以在**select返回监听套接字可读后，立即将监听套接字设置为非阻塞**，并且在后续调用accept时忽视EWOULDBLOCK(Berkeley)\ECONNABORTED(POSIX实现，客户终止连接时)、EPROTO（SVR4实现，客户终止连接时）和EINTR。