 隐患 2．对等套接字闭包 UNIX 有趣的一面是您几乎可以把任何东西看成是一个文件。文件本身、目录、管道、设备和套接字都被当作文件。这是新颖的抽象，意味着一整套的 API 可以用在广泛的设备类型上。考虑 read API 函数，它从文件读取一定数量的字节。read 函数返回读取的字节数（最高为您指定的最大值）；或者 -1，表示错误；或者 0，如果已经到达文件末尾。 如果在一个套接字上完成一个 read 操作并得到一个为 0 的返回值，这表明远程套接字端的对等层调用了 close API 方法。该指示与文件读取相同 没有多余的数据可以通过描述符读取（参见 清单 2）。 清单 2．适当处理 read API 函数的返回值 int sock, status.sock = socket(AF_INET, SOCK_STREAM, 0)....status = read(sock, buffer, buflen).if (status > 0) { /* Data read from the socket */} else if (status == -1) { /* Error, check errno, take action... */} else if (status == 0) { /* Peer closed the socket, finish the close */ close(sock). /* Further processing... */}同样，可以用 write API 函数来探测对等套接字的闭包。在这种情况下，接收 SIGPIPE 信号，或如果该信号阻塞，write 函数将返回 -1 并设置 errno 为 EPIPE。