

Linux操作系统套接字编程的5个隐患（4）PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/144/2021_2022_Linux_E6_93_8D_E4_BD_c103_144582.htm 隐患 4 . 发送结构化数据 套接字

是发送无结构二进制字节流或 ASCII 数据流（比如 HTTP 上的 HTTP 页面，或 SMTP 上的电子邮件）的完美工具。但是如果试图在一个套接字上发送二进制数据，事情将会变得更加复杂。比如说，您想要发送一个整数：您可以肯定，接收者将使用同样的方式来解释该整数吗？运行在同一架构上的应用程序可以依赖它们共同的平台来对该类型的数据做出相同的解释。但是，如果一个运行在高位优先的 IBM PowerPC 上的客户端发送一个 32 位的整数到一个低位优先的 Intel x86，那将会发生什么呢？字节排列将引起不正确的解释。通过套接字发送一个 C 结构会怎么样呢？这里，也会遇到麻烦，因为不是所有的编译器都以相同的方式排列一个结构的元素。结构也可能被压缩以便使浪费的空间最少，这进一步使结构中的元素错位。幸好，有解决这个问题的方案，能够保证两端数据的一致解释。过去，远程过程调用（Remote Procedure Call，RPC）套装工具提供所谓的外部数据表示（External Data Representation，XDR）。XDR 为数据定义一个标准的表示来支持异构网络应用程序通信的开发。现在，有两个新的协议提供相似的功能。可扩展标记语言/远程过程调用（XML/RPC）以 XML 格式安排 HTTP 上的过程调用。数据和元数据用 XML 进行编码并作为字符串传输，并通过主机架构把值和它们的物理表示分开。SOAP 跟随 XML-RPC，以更好的特性和功能扩展了它的思想。参见参考资料小节，获

取更多关于每个协议的信息。 隐患 5 . TCP 中的帧同步假定 TCP 不提供帧同步，这使得它对于面向字节流的协议是完美的。这是 TCP 与 UDP (User Datagram Protocol , 用户数据报协议) 的一个重要区别。UDP 是面向消息的协议，它保留发送者和接收者之间的消息边界。TCP 是一个面向流的协议，它假定正在通信的数据是无结构的，如图 1 所示。图 1

. UDP 的帧同步能力和缺乏帧同步的 TCP 图 1 的上部说明一个 UDP 客户端和服务端。左边的对等层完成两个套接字的写操作，每个 100 字节。协议栈的 UDP 层追踪写的数量，并确保当右边的接收者通过套接字获取数据时，它以同样数量的字节到达。换句话说，为读者保留了写者提供的消息边界。现在，看图 1 的底部 . 它为 TCP 层演示了相同粒度的写操作。两个独立的写操作 (每个 100 字节) 写入流套接字。但在本例中，流套接字的读者得到的是 200 字节。协议栈的 TCP 层聚合了两次写操作。这种聚合可以发生在 TCP/IP 协议栈的发送者或接收者中任何一方。重要的是，要注意到聚合也许不会发生 TCP 只保证数据的有序发送。对大多数开发人员来说，该陷阱会引起困惑。您想要获得 TCP 的可靠性和 UDP 的帧同步。除非改用其他的传输协议，比如流传输控制协议 (STCP) ，否则就要求应用层开发人员来实现缓冲和分段功能。 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com