

Linux操作系统套接字编程的5个隐患 (1) 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_144587.htm Socket API 是网络应用程序开发中实际应用的标准 API。尽管该 API 简单，但是开发新手可能会经历一些常见的问题。本文识别一些最常见的隐患并向您显示如何避免它们。在 4.2 BSD UNIX 操作系统中首次引入，Sockets API 现在是任何操作系统的标准特性。事实上，很难找到一种不支持 Sockets API 的现代语言。该 API 相当简单，但新的开发人员仍然会遇到一些常见的隐患。本文识别那些隐患并向您显示如何避开它们。

隐患 1 . 忽略返回状态

第一个隐患很明显，但它是开发新手最容易犯的一个错误。如果您忽略函数的返回状态，当它们失败或部分成功的时候，您也许会迷失。反过来，这可能传播错误，使定位问题的源头变得困难。捕获并检查每一个返回状态，而不是忽略它们。考虑清单 1 显示的例子，一个套接字 send 函数。

清单 1. 忽略 API 函数返回状态

```
int status, sock, mode./* Create a new
stream (TCP) socket */sock = socket( AF_INET, SOCK_STREAM,
0 )....status = send( sock, buffer, buflen, MSG_DONTWAIT ).if
(status == -1) { /* send failed */ printf( "send failed: %s\n",
strerror(errno) ).} else { /* send succeeded -- or did it? */}
```

清单 1 探究一个函数片断，它完成套接字 send 操作（通过套接字发送数据）。函数的错误状态被捕获并测试，但这个例子忽略了 send 在无阻塞模式（由 MSG_DONTWAIT 标志启用）下的一个特性。send API 函数有三类可能的返回值：如果数据成功地排到传输队列，则返回 0。如果排队失败，则返回 -1（通

过使用 `errno` 变量可以了解失败的原因)。如果不是所有的字符都能够在函数调用时排队，则最终的返回值是发送的字符数。由于 `send` 的 `MSG_DONTWAIT` 变量的无阻塞性质，函数调用在发送完所有的数据、一些数据或没有发送任何数据后返回。在这里忽略返回状态将导致不完全的发送和随后的数据丢失。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