

comp.unix.programmer: read on socket shows EOF... but only when app is running in background

read on socket shows EOF... but only when app is running in background

Source: <http://unix.derkeiler.com/Newsgroups/comp.unix.programmer/2004-08/0418.html>

From: Ruud Schramp (*r.schramp_at_telecom.dot.tno.dot.nl*)

Date: 08/19/04

Date: Thu, 19 Aug 2004 08:32:18 +0200

Hello All,

Does anybody know other reasons a TCP Socket will read eof besides being closed locally or receiving a FIN packet?

My multi threaded server application doesn't work when run in the background. The application is intended to run as a daemon so I want to create a commandline option that puts it in the background. It does work when run in the foreground, even when started with a "&" option, but not when an internal fork is used.

I did this (in lots of other programs as well) by putting a fork() function almost at startup in which the parent exits and the child continues running.

This forking itself works, so the application runs in the background, but now when I connect to it with TCP (the app has a server socket) the client socket is immediately "closed" again (saying it will unblock for read, reading the EOF (read returns 0)). The client connecting has not terminated the TCP connection, also tcpdump doesn't show a FIN packet being sent to the server. After verification with strace I see that no signals generated within the timeframe of these events. Also no unexpeced closes are created by the server application.

comparison of the strace logging of the application run in the foreground with the strace logging when forked show no difference upto the reception of the EOF.

Unfortunately the application is too big to show in the mailing list. I am afraid that I will not be able to reproduce the problem on a micro scale because, well I can't isolate the event. To my best knowledge it should work....

Anyone seen this type of events before?

regards,

comp.unix.programmer: read on socket shows EOF... but only when app is running in background

Ruud Schrap