

Re: Socket Programming: How to terminate a thread "listening" for UDP packets?

Source: <http://unix.derkeiler.com/Newsgroups/comp.unix.programmer/2008-01/msg00305.html>

- *From:* David Schwartz <davids@xxxxxxxxxxxxxx>
 - *Date:* Wed, 16 Jan 2008 11:15:49 -0800 (PST)
-

On Jan 16, 7:44 am, Lars Uffmann <a...@xxxxxxxxxxxxxx> wrote:

What's the "natural" approach to "defuse" a function that is listening for incoming packets?

There are several ways, some of which naturally fit your situation and some of which don't:

- 1) Defuse the socket in a safe way that assures that the blocking function will return. This is not possible for UDP, though it is possible with TCP, using 'shutdown'.
- 2) Send the thread a signal to make it abort its operation. I do not recommend this approach.
- 3) Set a shutdown flag and then create the event the thread is waiting for. That is, if the thread is waiting for a UDP packet on port 402, send it one. Make sure it checks the shutdown flag as soon as its blocking function finishes and discards the packet.
- 4) Set the socket non-blocking. Create a pipe. Use 'select' or 'poll' to wait for both the socket and the pipe. Block in 'select' or 'poll'. When you want to shutdown the thread, send a byte on the pipe. This will cause the thread to unblock.

Is there maybe a function in the socket library that will tell me IF there is data to be read (or not) instantly, which I can call and only start a `recvfrom()` call when there is actually data to be read?

Yes, but that won't work. UDP is unreliable. By the time you call 'recvfrom', the packet could be dropped, and you could miss a shutdown. (Precisely this type of bug caused an infamous Linux `inetd` UDP denial of service attack.)

Re: Socket Programming: How to terminate a thread "listening" for UDP packets?

I'm not so much looking for a specific solution to my problem, more the general approach to implementing such stuff – I'm still a beginner ;) don't wanna succumb to the dark side of the force.

See the list of approaches above.

DS

.