

SO_LINGER brokenness...

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/net/2008-01/msg00214.html>

- *From:* Sean Chittenden <sean@xxxxxxxxxxxxxxxx>
 - *Date:* Wed, 23 Jan 2008 11:56:33 -0800
-

Howdy. Looks like SO_LINGER is broken on loopback. Here's the test case:

0. Install memcached (databases/memcached)

1. Download test_linger.c: fetch http://sean.chittenden.org/pubfiles/freebsd/test_linger.c

2. Compile: gcc test_linger.c -o test_linger

3. Fire up memcached in a new window: memcached -vvv -p 11211

4. Test: ./test_linger localhost 11211

5. Watch for either:

```
<16 new client connection
<16 set valid_key 0 0 8
>16 STORED
<16 set valid_key 0 0 8
>16 STORED
<16 connection closed.
```

Fail.

or

```
<16 new client connection
<16 set valid_key 0 0 8
>16 STORED
<16 set valid_key 0 0 8
>16 STORED
<16 get ENOENT_key
>16 END
<16 connection closed.
```

Win.

Can someone verify that my test case is correct? If it is, I'll port it to src/tools/regression/sockets/.

Broken or not, I don't think the behavior suggested in src/netinet/tcp_usrreq.c:1498 is correct. A The test above write(2)'s and read(2)'s some data successfully before testing the validity of the half-close(2)'ed state, so I'm not worried about whether or not a SYN has been received. I haven't looked too closely yet, but it looks like tcp_close() needs to call to sbflush() if SO_LINGER is set (in fact, it doesn't look like there's any special handling of linger in tcp_subr.c, so I'm not sure if I'm glancing in the right area).

SO_LINGER brokenness...

/*

- * User issued close, and wish to trail through shutdown states:
 - * if never received SYN, just forget it. If got a SYN from peer,
 - * but haven't sent FIN, then go to FIN_WAIT_1 state to send peer a FIN.
 - * If already got a FIN from peer, then almost done; go to LAST_ACK
 - * state. In all other cases, have already sent FIN to peer (e.g.
 - * after PRU_SHUTDOWN), and just have to play tedious game waiting
 - * for peer to send FIN or not respond to keep-alives, etc.
 - * We can let the user exit from the close as soon as the FIN is acked.
- */

Thoughts/guidance? -sc

--

Sean Chittenden

sean@xxxxxxxxxxxxxx

<http://sean.chittenden.org/>

freebsd-net@xxxxxxxxxxxx mailing list

<http://lists.freebsd.org/mailman/listinfo/freebsd-net>

To unsubscribe, send any mail to "freebsd-net-unsubscribe@xxxxxxxxxxxx"