

Check2 实验报告

王昕浩 221502005

1. Structure and Design

在 wrapping_integer 中,

wrap 方法直接返回 zero_point+n。

unwrap 方法首先计算出一个可能的结果 ans, 再通过 judge 的大小来修改 ans 的结果。

在 TCPReceiver 中,

receive 方法中, 我将 SYN_flag==true 作为插入时的常态, message 中 SYN 为 true 时作为特判, 维护状态并插入。我利用 reassebler 的 insert 方法会忽略大于 available_capacity 的字符的性质简化了代码。

send 方法中, 我按照实验手册要求生成了 message 并返回。

2. Implementation Challenges

本次实验中, 我遇到了一个原因是 reassembler 中的 bug 的 bug, 这是由于 bytes_pending_数据的维护失效。在 recv_reorder 的 many gaps, then subsumed 测试中, 我的 reassembler 因为在拼接时未考虑到之前存储的多个元素, 因此未正确修改 bytes_pending 的值。

所以建议增加 check1 的测试, 其内容为 recv_reorder 的 many gaps, then subsumed 测试中 tcp_receiver 所需对 reassembler 的操作。

除此之外, 需注意到 bytes_stream 的 error_需要和 tcp_receiver 的 RST_flag_绑定。

3. Remaining Bugs

没有剩余的 bug。

4. Experimental results and performance

完成截图如下:

```

Start 6: byte_stream_two_writes
5/29 Test #6: byte_stream_two_writes ..... Passed 0.01 sec
Start 7: byte_stream_many_writes
6/29 Test #7: byte_stream_many_writes ..... Passed 0.04 sec
Start 8: byte_stream_stress_test
7/29 Test #8: byte_stream_stress_test ..... Passed 0.02 sec
Start 9: reassembler_single
8/29 Test #9: reassembler_single ..... Passed 0.01 sec
Start 10: reassembler_cap
9/29 Test #10: reassembler_cap ..... Passed 0.01 sec
Start 11: reassembler_seq
10/29 Test #11: reassembler_seq ..... Passed 0.01 sec
Start 12: reassembler_dup
11/29 Test #12: reassembler_dup ..... Passed 0.03 sec
Start 13: reassembler_holes
12/29 Test #13: reassembler_holes ..... Passed 0.01 sec
Start 14: reassembler_overlapping
13/29 Test #14: reassembler_overlapping ..... Passed 0.01 sec
Start 15: reassembler_win
14/29 Test #15: reassembler_win ..... Passed 0.20 sec
Start 16: wrapping_integers_cmp
15/29 Test #16: wrapping_integers_cmp ..... Passed 0.01 sec
Start 17: wrapping_integers_wrap
16/29 Test #17: wrapping_integers_wrap ..... Passed 0.01 sec
Start 18: wrapping_integers_unwrap
17/29 Test #18: wrapping_integers_unwrap ..... Passed 0.01 sec
Start 19: wrapping_integers_roundtrip
18/29 Test #19: wrapping_integers_roundtrip ..... Passed 0.84 sec
Start 20: wrapping_integers_extra
19/29 Test #20: wrapping_integers_extra ..... Passed 0.22 sec
Start 21: recv_connect
20/29 Test #21: recv_connect ..... Passed 0.01 sec
Start 22: recv_transmit
21/29 Test #22: recv_transmit ..... Passed 0.21 sec
Start 23: recv_window
22/29 Test #23: recv_window ..... Passed 0.01 sec
Start 24: recv_reorder
23/29 Test #24: recv_reorder ..... Passed 0.01 sec
Start 25: recv_reorder_more
24/29 Test #25: recv_reorder_more ..... Passed 0.69 sec
Start 26: recv_close
25/29 Test #26: recv_close ..... Passed 0.01 sec
Start 27: recv_special
26/29 Test #27: recv_special ..... Passed 0.02 sec
Start 37: compile with optimization
27/29 Test #37: compile with optimization ..... Passed 0.63 sec
Start 38: byte_stream_speed_test
        ByteStream throughput: 5.22 Gbit/s
28/29 Test #38: byte_stream_speed_test ..... Passed 0.08 sec
Start 39: reassembler_speed_test
        Reassembler throughput: 9.43 Gbit/s
29/29 Test #39: reassembler_speed_test ..... Passed 0.13 sec

```

100% tests passed, 0 tests failed out of 29

Total Test time (real) = 5.91 sec

Built target check2

wexther@wxh:~/minnow\$