Lab 3 Writeup

My name: Zuo, Qikun

My Student number : 201830013

This lab took me about 3 hours to do. I did attend the lab session.

1. Program Structure and Design:

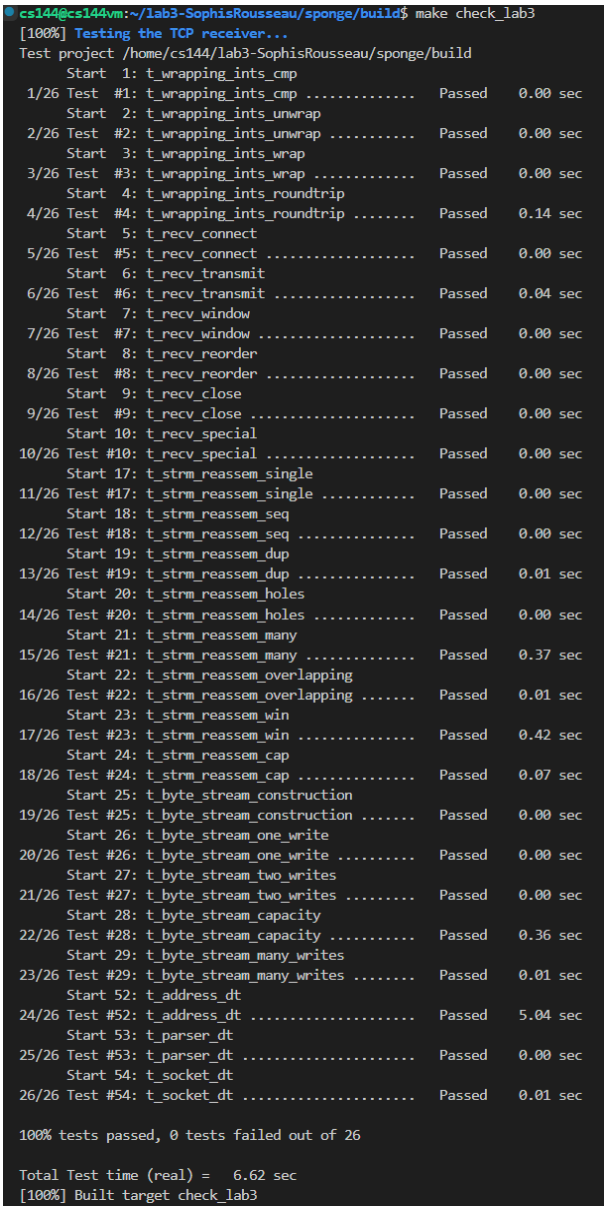
In wrapping_integers.cc, we write wrap() and unwrap() to implement the conversion between sequence numbers and absolute sequence numbers. The relationship between sequence numbers and absolute sequence numbers is: abs_seqno + isn (mod 1 << 32) = seqno. So the implementation of the function wrap() is easy: just add abs_seqno and isn together according to the operator overloaded in the class WrappingInt32. Note that a seqno n corresponds to many absolute seqnos, and we need to find the absolute seqno that is closest to the checkpoint. Thus to implement function unwrap(), we could first find the seqno of the checkpoint, i.e., checkpoint + isn (mod 1 << 32). And then we need to discuss the quantitative relationship between n and the seqno of the checkpoint to determine the distance between the absolute seqno of n and checkpoint. Several different cases and edge conditions should be taken into consideration in order to make the result strict.

In tcp_receiver.cc, we write segment_received() to: (1)listen to 'syn' to record the beginning of a stream, the ISN of the stream (i.e., seg.header().seqno), and the FIN signal; (2)calculate the checkpoint, the absolute seqno, and the stream index of the stream; (3)use the function push_substring() of the reassembler to reassemble the byte stream. Writing function akcno() and window_size() is easy: just use the right member functions of the class Bytestream. And the ISN and FIN signal should also be taken into consideration when calculating the acknowledgment number (ackno).

2. Implementation Challenges:

The hardest part of the lab is to unwrap the seqno, i.e., to convert the seqno into the absolute seqno according to the given checkpoint. Several different cases and edge conditions should be taken into consideration in order to make the result strict. We drew a picture of the mapping relationship between the absolute seqnos and seqnos to make the conversion algorithm clear and comprehensible.

3. Remaining Bugs:



```
cs144@cs144vm:~/lab3-SophisRousseau/sponge/build$ make check_lab3
[100%] Testing the TCP receiver...
Test project /home/cs144/lab3-SophisRousseau/sponge/build
      Start  1: t_wrapping_ints_cmp
 1/26 Test  #1: t_wrapping_ints_cmp .............   Passed    0.00 sec
      Start  2: t_wrapping_ints_unwrap
 2/26 Test  #2: t_wrapping_ints_unwrap ..........   Passed    0.00 sec
      Start  3: t_wrapping_ints_wrap
 3/26 Test  #3: t_wrapping_ints_wrap ............   Passed    0.00 sec
      Start  4: t_wrapping_ints_roundtrip
 4/26 Test  #4: t_wrapping_ints_roundtrip .......   Passed    0.14 sec
      Start  5: t_recv_connect
 5/26 Test  #5: t_recv_connect ..................   Passed    0.00 sec
      Start  6: t_recv_transmit
 6/26 Test  #6: t_recv_transmit .................   Passed    0.04 sec
      Start  7: t_recv_window
 7/26 Test  #7: t_recv_window ...................   Passed    0.00 sec
      Start  8: t_recv_reorder
 8/26 Test  #8: t_recv_reorder ..................   Passed    0.00 sec
      Start  9: t_recv_close
 9/26 Test  #9: t_recv_close ....................   Passed    0.00 sec
      Start 10: t_recv_special
10/26 Test #10: t_recv_special ..................   Passed    0.00 sec
      Start 17: t_strm_reassem_single
11/26 Test #17: t_strm_reassem_single ...........   Passed    0.00 sec
      Start 18: t_strm_reassem_seq
12/26 Test #18: t_strm_reassem_seq ..............   Passed    0.00 sec
      Start 19: t_strm_reassem_dup
13/26 Test #19: t_strm_reassem_dup ..............   Passed    0.01 sec
      Start 20: t_strm_reassem_holes
14/26 Test #20: t_strm_reassem_holes ............   Passed    0.00 sec
      Start 21: t_strm_reassem_many
15/26 Test #21: t_strm_reassem_many .............   Passed    0.37 sec
      Start 22: t_strm_reassem_overlapping
16/26 Test #22: t_strm_reassem_overlapping ......   Passed    0.01 sec
      Start 23: t_strm_reassem_win
17/26 Test #23: t_strm_reassem_win ..............   Passed    0.42 sec
      Start 24: t_strm_reassem_cap
18/26 Test #24: t_strm_reassem_cap ..............   Passed    0.07 sec
      Start 25: t_byte_stream_construction
19/26 Test #25: t_byte_stream_construction ......   Passed    0.00 sec
      Start 26: t_byte_stream_one_write
20/26 Test #26: t_byte_stream_one_write .........   Passed    0.00 sec
      Start 27: t_byte_stream_two_writes
21/26 Test #27: t_byte_stream_two_writes ........   Passed    0.00 sec
      Start 28: t_byte_stream_capacity
22/26 Test #28: t_byte_stream_capacity ..........   Passed    0.36 sec
      Start 29: t_byte_stream_many_writes
23/26 Test #29: t_byte_stream_many_writes .......   Passed    0.01 sec
      Start 52: t_address_dt
24/26 Test #52: t_address_dt ....................   Passed    5.04 sec
      Start 53: t_parser_dt
25/26 Test #53: t_parser_dt .....................   Passed    0.00 sec
      Start 54: t_socket_dt
26/26 Test #54: t_socket_dt .....................   Passed    0.01 sec

100% tests passed, 0 tests failed out of 26

Total Test time (real) =   6.62 sec
[100%] Built target check_lab3
```

Until now, no bugs are found in the code submitted.