Lab 2 Writeup

My name: 王聪颖

My Student number : 211220180

This lab took me about 8 hours to do. I did attend the lab session.

1. Program Structure and Design:

wrapping_integers:

- 使用移位运算加快代码运行速度
- left_abs_seqno <= checkpoint < right_abs_seqno
- 使用mask来使right_abs_seqno快速逼近checkpoint，减少计算量

```
uint64_t unwrap(WrappingInt32 n, WrappingInt32 isn, uint64_t checkpoint) {
    const uint64_t mod = static_cast<uint64_t>(1) << 32;
    uint64_t offset = (n - isn >= 0) ? uint64_t(n - isn) : uint64_t(n - isn
+ mod);
    uint64_t right_abs_seqno = 0;
    uint64_t left_abs_seqno = 0;

    if (offset >= checkpoint)
        return offset;

    right_abs_seqno = offset;
    // fastly set value to approximate checkpoint
    right_abs_seqno |= ((checkpoint >> 32) << 32);
    if (right_abs_seqno <= checkpoint)
        right_abs_seqno += mod;
    left_abs_seqno = right_abs_seqno - mod;
    return (checkpoint - left_abs_seqno < right_abs_seqno - checkpoint) ?
left_abs_seqno : right_abs_seqno;
}
```

tcp_receiver:

- 增加三个私有成员

```
bool _flag_SYN{false};
bool _flag_FIN{false};
WrappingInt32 _isn{0};
```

- 根据_flag_SYN，将收到segment时，receiver的status分为： 1.连接建立前；2.连接建立后
- 根据header.syn，区分segment： 1.请求连接segment；2.普通segment

- 根据header.fin，区分segment： 1.连接终止segment；2.普通segment

```
void TCPReceiver::segment_received(const TCPSegment &seg) {
    const TCPHeader &header = seg.header();
    // haven't connected
    if (!_flag_SYN) {
        // connect failure
        if (!header.syn)
            return;
        // create connection
        else {
            _flag_SYN = true;
            _isn = header.seqno;
            bool this_seg_eof = false;
            // close connection
            if (header.fin)
                _flag_FIN = this_seg_eof = true;
            string data = seg.payload().copy();
            _reassembler.push_substring(data, 0, this_seg_eof);
        }
    }
    // connection already exists
    else {
        // create connection twice
        if (header.syn)
            return;

        bool this_seg_eof = false;
        // close connection
        if (header.fin)
            _flag_FIN = this_seg_eof = true;
        // reassemble data
        string data = seg.payload().copy();
        uint64_t checkpoint = _reassembler.expect_index();
        uint64_t abs_seqno = unwrap(header.seqno, _isn, checkpoint);
        Index index = abs_seqno - 1;
        _reassembler.push_substring(data, index, this_seg_eof);
    }
}
```

- 判断是否ack FIN，必须使用_flag_FIN && _reassembler.stream_out().input_ended()

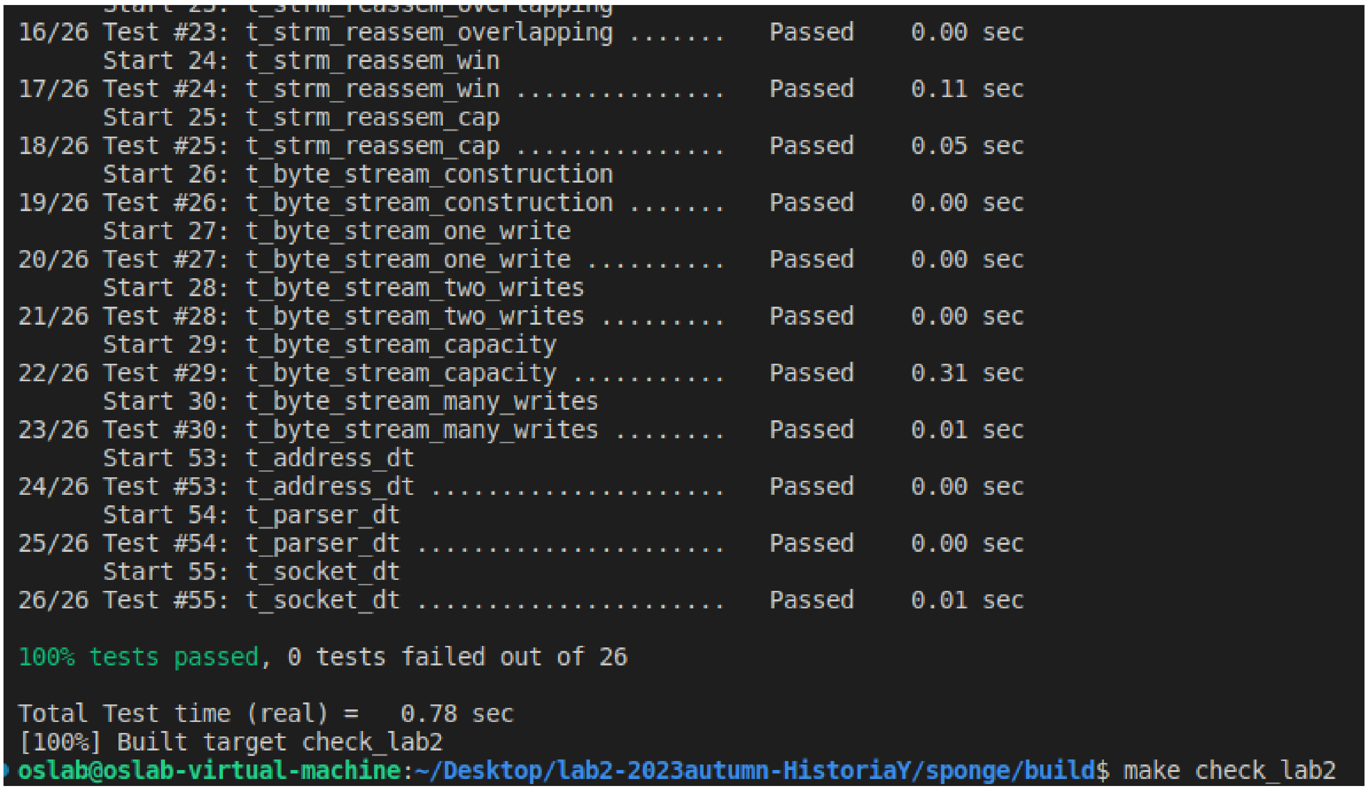
```
optional<WrappingInt32> TCPReceiver::ackno() const {
    // haven't connected
    if (!_flag_SYN)
        return {};
    // connection already exists
    // there was once a bug (_flag_FIN &&
_reassembler.stream_out().input_ended()) ? 1 : 0 must be enclosed by ()!!!
    return wrap(1 + _reassembler.expect_index() + ((_flag_FIN &&
_reassembler.stream_out().input_ended()) ? 1 : 0),
                   _isn);
}
```

2. Implementation Challenges:

- segment_received(const TCPSegment &seg)的逻辑判断容易成团混淆，本实现使用了清晰的逻辑划分方式，并且使用注释在每个逻辑分支处标注
- ackno()处，最开始出现了不易察觉的bug——三元判断需使用括号将整个表达式独立（如前文ackno()实现中注释所述）

3. Remaining Bugs:

本次实验测试样例全部通过，无遗留bug



```
16/26 Test #23: t_strm_reassem_overlapping .......   Passed    0.00 sec
      Start 24: t_strm_reassem_win
17/26 Test #24: t_strm_reassem_win ...............   Passed    0.11 sec
      Start 25: t_strm_reassem_cap
18/26 Test #25: t_strm_reassem_cap ...............   Passed    0.05 sec
      Start 26: t_byte_stream_construction
19/26 Test #26: t_byte_stream_construction .......   Passed    0.00 sec
      Start 27: t_byte_stream_one_write
20/26 Test #27: t_byte_stream_one_write ..........   Passed    0.00 sec
      Start 28: t_byte_stream_two_writes
21/26 Test #28: t_byte_stream_two_writes .........   Passed    0.00 sec
      Start 29: t_byte_stream_capacity
22/26 Test #29: t_byte_stream_capacity ...........   Passed    0.31 sec
      Start 30: t_byte_stream_many_writes
23/26 Test #30: t_byte_stream_many_writes ........   Passed    0.01 sec
      Start 53: t_address_dt
24/26 Test #53: t_address_dt .....................   Passed    0.00 sec
      Start 54: t_parser_dt
25/26 Test #54: t_parser_dt ......................   Passed    0.00 sec
      Start 55: t_socket_dt
26/26 Test #55: t_socket_dt ......................   Passed    0.01 sec

100% tests passed, 0 tests failed out of 26

Total Test time (real) =   0.78 sec
[100%] Built target check_lab2
oslab@oslab-virtual-machine:~/Desktop/lab2-2023autumn-HistoriaY/sponge/build$ make check_lab2
```