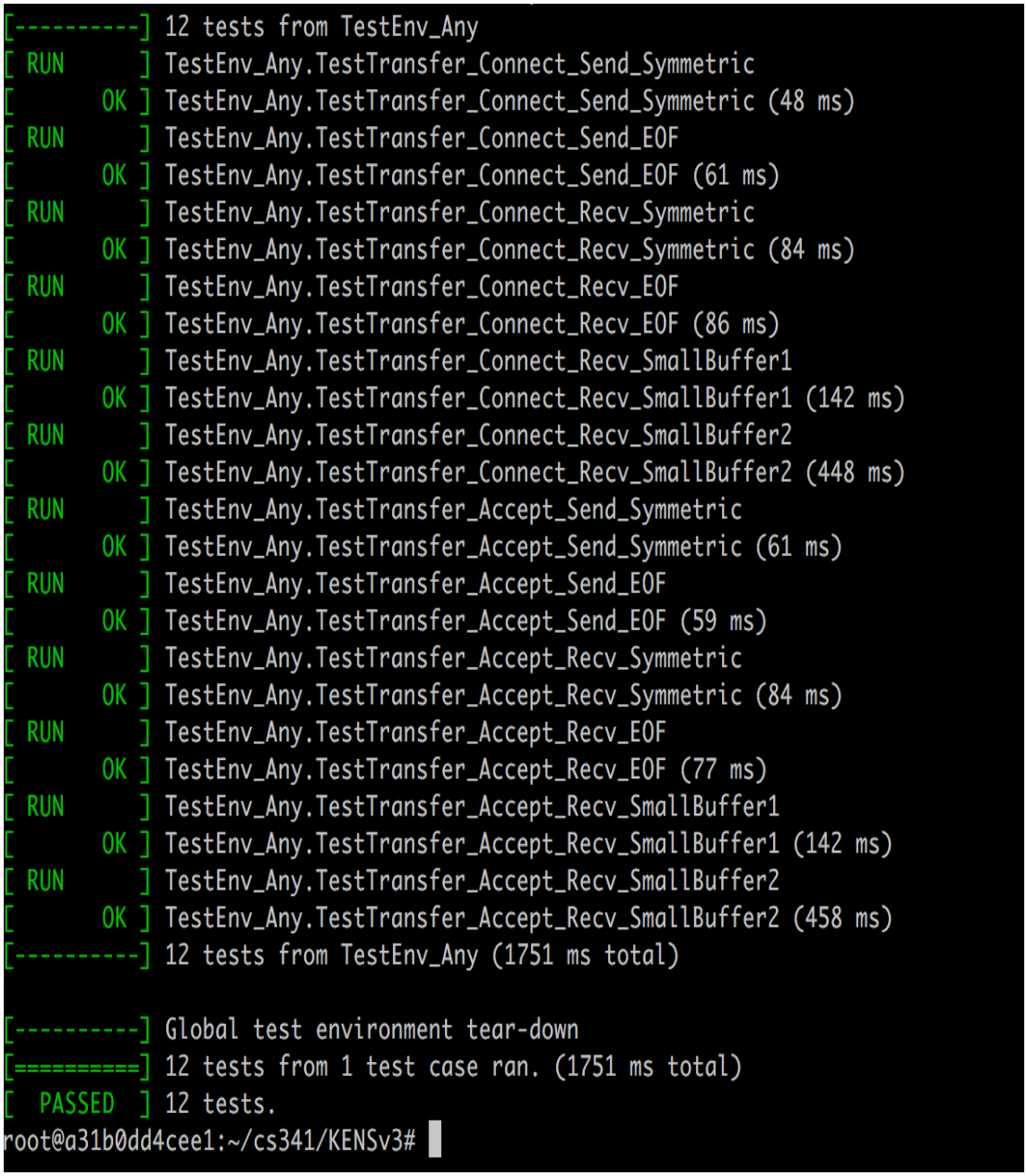
Team30 20140040 Keonil Kim

CS341 Project3-1 Report



127 map<struct PidFd, pair<UUID, pair<void *, size_t>>> read_info_list;
128 map<struct PidFd, deque<uint8_t>> read_buffer_list;
129 map<struct PidFd, pair<size_t, map<int, Packet *>>> internal_buffer_list;
130 map<struct PidFd, map<UUID, deque<pair<int, Packet *>>>> blocked_packet_list;
131 map<struct PidFd, deque<pair<UUID, size_t>>> blocked_uuid_list;

These are added structures for project 3-1

read_info_list save blocked read call

read_buffer_list save read buffer for each connection

internal_buffer_list is maintained with pair of total size and packets mapped to expected ack number

blocked_packet_list saves blocked write packet, which are saved with the expected ack number. All mapped to its syscallUUID.

blocked_uuid_list saves the order of blocked write packet's UUID. The unblocking order is determined with this and blocked_packet_list.

- read
For read function, check if read buffer is empty, and if it is not, read data from read buffer by size of count

- write
First, create packet with the maximum data length of 512 and check internel buffer. If ther is enough space, send Packet immediately. Otherwise, block the packet.

- PacketArrived(when ACK arrived)
Data transfer packets contain ACK flag.

First To handle the ACK which is response of data succesfully received, we first check out the internal buffer. If there is a packet with ack, identical to expected vallue, remove it from buffer. Then check the size of buffer so that if the space is enough, unblock the packet in order.

To handle the ack which is coming with data, read the data and save it to the read_buffer. Check if there is blocked read call, and if exists, unblock it and return. Then, if successfully reads all data, send ACK packet with proper ack number immediately.