1. DeogHun Kim (dkim109),
2. We implemented delay channel by sleeping the thread. So, we first calculated the the time that the messages should supposed to be sent by using the equation that is given on the description and sleep for time that is calculated then sends the message to the coordinate server. However, we did not sleep any of threads that receives messages or sends the message. We have 4 seperate delay thread for the different destinations(since there is 4 servers) and each thread has queue that can save any messages that has the same destination in order to ensure FIFO. Additionally, we have one more thread and queue that handles the dealy of ack messages.
3. i) We simulated linearizability by using queue and lock. In the coordinate server, We first put all the request in the queue and then check whether there is previous operation that is still being processed. How we check is by using the lock. Whenever operation starts its process, we set the lock as True so that any operation that comes after cannot be proceed. This ensures linearizablility because it handles the commands always in the ordrer. With this basic rule in the coordinate server, communication between the coordinate server and basic servers(which can act as client) are following. Whenever user types the command, it sends the msg to coordinate server and this coordinate server broadcast then wait until all the ack comes. If it gets all the ack, sends to the client ack message and then free the lock so that next operation that is stuck in the queue can be processed.

ii) Sequential consistency is quite similar. The difference between linearizability between sequential consistency is that sequential consistency gets the value from the local key-value store instead of broadcasting. In this case, there is case that the client reads the newest version in its own key-value store but maybe old one considering all of the servers. This makes sequential consistency but not linearizable.

iii) For eventual consistency, we don’t have lock that stops the commands being processed. Whenever the commands comes to the coordinate server, sends to each server without blocking of the lock. However, there is still delay and thus each server gets value in different time.