Big Exercise 1: Lamport Clock

The source code file is named ***big_exercise_1_lamport_clock.py***, the configuration file named ***configuration_file***, output files named like ***output_xxx*** where the ***xxx*** is the id of the client thread who output this file. To the program type the following command in the terminal:

python3 big_exercise_1_lamport_clock.py configuration_file 1

There are two arguments *configuration_file* and *1* in the command which are demanded in the problem description. However the latter is actually not needed in my implementation.

The program mainly consists of 4 functions: *read_params(config_file)*, *send_message(sender_id, receiver_port, lamport_clock)*, *client_thread(own_index, ids, ports)*, *main function*. *read_params(config_file)* reads lines of id and port from the configuration file. *config_file* is the name of the configuration file and it returns two arrays of ids and ports respectively. *send_message(sender_id, receiver_port, lamport_clock)* receives the id of the client who is sending the message, the port of the receiver, the lamport clock of the sending client. *client_thread(own_index, ids, ports)* takes an index which refers to one client's id and port in the array, and the array of id and ports as parameters. *client_thread()* implements the lamport clock algorithm. It receives messages from other clients and also randomly decide to send a message to other clients or simulate a local event. *send_message()* method is called by it. In *main()* function, firstly call *read_params()* to read the information from the configuration file. Then loop the id array and start a new thread to run the *client_thread()* method to simulate an individual client. At the last line of program, let the main thread sleep for seconds to wait the child client threads finish their job.