

PA4 Q&A 2

(1) how are we supposed to know when all the Mappers are done on the Server side? Because we know the max number of Mappers/connections, but as long as the count is less than that, it seems like we don't know for sure if there could be more connections coming? OR - is it OK for the Server just stay open until manually killed?

☐ The server is not able to know when all Mappers are done with their job. That is why the server stays open and listen next client connections until it is manually killed.

(2) What should the client do when it receives a RSP_NOK code?

☐ Nothing. Just print out the response in a format stated in the section 4.3.

(3) Is it ok if update_status_table uses a one dimensional array of structs instead of a 2D-array?

☐ Yes. The data types and shapes for the requests and responses should conform with the guidelines. However, you can use any data types and shapes for the internal data structures (azList and update_status table).

(4) If a mapper id tries to check in but it is already in the update status table, is this an error that should produce a RSP_NOK response?

☐ If there is an existing entry for the mapper ID && the check in/out flag is 0 (checked out), the server should produce RSP_OK and change the flag to 1 (checked in). However, of there is an existing entry for the mapper ID && the check in/out flag is 1 (checked in), the server should produce RSP_NOK and change nothing in the table.

(5) Does the server have a thread per process or per request?

☐ The server has a thread per process for mapper clients, whereas the server has a thread per request for master clients (This is extra credit).

(6) Does the server need to enforce a deterministic order of requests or does it only need to make sure that the check in/out is correct for the specific request that is being sent?

☐ This is correct - it only needs to make sure that the check in/out is correct for the specific request that is being sent.

(7) If a mapper id tries to check out when it already checked out, is this a RSP_NOK?

☐ Yes. RSP_NOK should be sent to the client.

(8) For the letter frequencies, does this correspond to first letter in word like PA2?

☐ Yes. Same with PA2.

(9) Are we just counting the first letter of the first word again like we were in PA2 or are we counting the first letter of every word like PA3?

☐ Counting the first letter of the first word like PA2.

(10) In the Assumptions section, the maximum number of mapper clients per a master client is 32. Each mapper client (mapper process) initiates a TCP connection to the server, which means the maximum number of connections to the server is 32. Then, why it still says "Maximum number of concurrent connections at the server side is 50"?

□ A client program can initiate up to 32 TCP connections by 32 mapper clients. The server can handle 50 TCP connections concurrently. That is, you can use the value (50) when calling `listen()` function at the server side. Since we will not test your server program by running multiple clients programs in parallel, you can assume that the maximum number of TCP connections (32) established between the server and clients does not exceed the maximum number of concurrent connections (50).

(11) The prints in the Appendix show that a process prints all of its commands and responses in a chunk before that of the other process prints. There should be some interweaving of prints between processes because the processes are executing in parallel.

□ The logs can be interweaved. As long as your mappers are running in parallel, the printing order of logs are not considered for grading purposes.

(12) How to test if the server handles error cases if I don't attempt the extra credit?

□ You can test the error cases by sending different sets of requests from your mapper client. For example, first create one mapper client, and send two CHECKIN requests to the server. You will get one RSP_OK for the first checkin request, and will get RSP_NOK for the next one.