S2 - 2023

Software Developer - Test

***

Coding Language: Preferred C++, or Python if you are more confident

Objectives: The system does not have to be fully optimized. The first goal is to test your skills regarding the position you are applying for:

- networking development
- multi-actors' communication
- multiprocessing/multithreading environment

The second goal is to evaluate your coding style quality as well as your ability to define and manage tests.

Review method:

- Code presentation: You will be invited to present your code and the problem you faced as well as the various choices you had to do;
- Code review;
- Integration test validation.

Mission:

Develop a multiclient application. This application will be composed at least of two types of clients:

- Basic User
- Proxy/Server

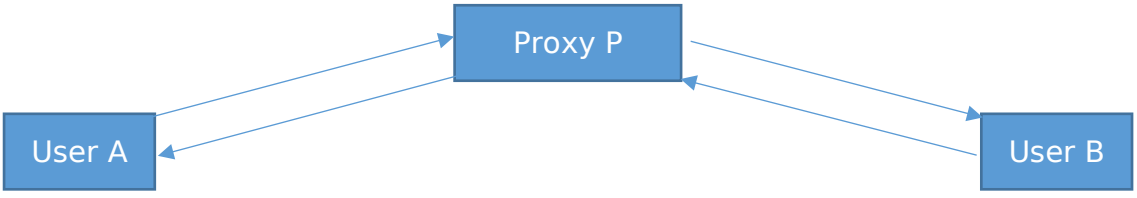
Users and server must communicate through a TCP socket on the same machine.

The Proxy will have to manage the communication between two basic users. The following test cases highlight the application specification.

It will be appreciated that the app is packaged in a binary containing the following options:

- –mode Proxy/User

- --secret <secretvalue>



Test Case 1: Manage absence of correspondent

- P is waiting for incoming connection
- A connects to P, P confirms that the connexion is ok but that there is no other connected user, A is waiting up to 30 secs. After 30secs, A disconnects from P because nobody has connected to the proxy.

Test Case 2: Manage basic communication

- P is waiting for incoming connection
- A connects to P, P confirm that the connexion is ok but that there is no other connected user, A is waiting up to 30 secs.
- B connects to P before the remaining time of 30 secs. P informs B that he will be connected with the connected user A.
- A is informed that it has been connected to B by P.
- A sends a message “hello” to B relayed by P (P prints the message in its terminal before sending the message to B).
- B prints the message in its terminal

Test Case 3: Manage bidirectional communication

- P is waiting for incoming connection
- A connects to P, P confirm that the connexion is ok but that there is no other connected user, A is waiting up to 30 secs.
- B connects to P before the end of the 30 secs countdown. P informs B that he will be connected with the connected user A.
- A is informed that it has been connected to B by P.
- A sends the message “[CMD]ECHOREPLY snowpack” to B relayed by P (P prints the message in its terminal before sending the message to B).
- B prints the message in its terminal and send back to A via P a message “snowpack”. A prints the received message

Test Case 4: Manage bidirectional communication but connect to correspondent based on dedicated secret – Same secrets

- P is waiting for incoming connection
- A connects to P and requests to connect to the guy having secret “123”, P confirm that the connexion is ok but that there is no other connected user, A is waiting up to 30 secs.
- B connects to P before the remaining time of 30 secs and asks to connect to the guy having secret “123”. P informs B that he will be connected with the connected user A.
- A is informed that it has been connected to B by P.
- A sends the message “[CMD]ECHOREPLY snowpack” to B relayed by P (P prints the message in its terminal before sending the message to B).
- B prints the message in its terminal and send back to A via P a message “snowpack”. A prints the received message

Test Case 5: Manage bidirectional communication but connect to correspondent based on dedicated secret – Different secrets

- P is waiting for incoming connection
- A connects to P and requests to connect to the guy having secret “123”, P confirm that the connexion is ok but that there is no other connected user, A is waiting up to 30 secs.
- B connects to P before the remaining time of 30 secs and asks to connect to the guy having secret “456”. P informs B that currently there is no user connected with this secret. B waits up to 30 secs and disconnects. While this period, A has also disconnected