

Lab03: multi-user server

Grading Key for Advanced Socket exercise

Purpose of the exercise: Write a chat system using sockets. Server side should accept multiple clients, client side should connect to the server. Both sides must avoid blocking.

To read p. 374 to 389 in "רשתות מחשבים"

Verification	Expected result	Score if not achieved
Basic functions		
Connection of 2 clients at the same time	The server receives the 2 clients	Grade 0
Both clients execute NAME and then GET_NAMES, their names appear in the list of clients they receive. The names of all clients are one word only, in English, case sensitive, without special characters.	Their names appear on the list of clients they receive.	Minus 30
A client sends a message to another client, only one word without any space.	The other client receives the word, including the correct name of the client who sent it.	Minus 30
Client disconnects with EXIT.	The server goes on running properly and does not crash or hang.	Minus 30
A client sends a message that is not one of the defined messages.	The server does not crash, and sends an error message	Minus 20
Application standardization		
A client sends a one-word message to another client.	The other client receives the message.	Minus 10
A client sends a message to himself.	The message is received, or the client receives a message that it is not possible to send a message to yourself (think about the case and take care of it).	Minus 10
A client picks the same name as another client.	A message indicating that it is impossible, is displayed.	Minus 20
When the client disconnects, its name is deleted from the client list (the other client executes GET_NAMES).	The list of names is updated upon logout.	Minus 20

A client sends a message to a client that does not exist.	The client receives a message that the other client does not exist.	Minus 20
A client sends a message that is not one of the defined messages.	The client receives a message that this is not a defined message.	Minus 10
Sending a valid command but with wrong parameters (for example adding words after GET_NAMES).	The programmer can choose whether to execute the command and ignore the additional words or ignore the command. What is important is that the server does not crash.	Minus 10
A client crash doesn't crash the server or other clients.	Close the client without EXIT and check that there is no effect on the server and the other clients.	Minus 10
Deadlocking		
Blocking test. The following cases are not good: A client does not receive a message from another client until he taps some character. A client who has started writing a message, stops receiving messages until he finishes sending.	No blocking. Messages appear directly at the client who should receive them.	Minus 20
Code		
Efficiency	The protocol between the server and the client uses a length field	Minus 10
Correct programming	Structure of the program: separation into functions, constants, documentation	Minus 1 to 5 according to the assessment of the examiner
Refactoring and OO	The server and client make use of the protocol. There are no cases where a function is defined, and it is written again in the code. As above, constants: there is no way that if defined as constant, a number is written in the code.	Minus 20

Good Luck!