Intro. to Network Programming 2022 Fall

Homework 2 – Game 1A2B

General

You will need to implement multiplayer 1A2B game server and your server should be implemented in C/C++

We will provide you docker image, you should ensure your code can run in the given image

We will run **bash build.sh** to compiler your code, and your server executable code pathname should be **build/server**

For student who store data in sqlite3, note that each testcase is run independently (No data at the start of each testcase) in this assignment, so you need to cleanup server's data if you store them in sqldb.

Timeline

Submission Deadline 11/29
Demo 12/1
Late Submission Deadline 12/20
Late Submission Demo 12/22

Submission

Example fornat HW2_109550xxx.zip
To check the zip format is correct and your output of sample testcases, you can run
python3 demo.py -s HW2_109550xxx.zip

Submission that with wrong zip format will start the score from 60

Requirement

Your server should be able to handle at least **10** clients

Client commands are list in the table

Command format	Description	Result	
register <username> <email> <user password>	Register with username, email and password. <username> and <email> must be unique, <password> has no limitation. Fail(1) username or email is used	Success	Regsiter Successfully
		Fail(1)	Username or Email is already used

	Please note that you have to send this request and receive the response with **UDP**		
login <username> <password>	Login account One client can only logged in as one account Fail(1) Username not found Fail(2) You already logged in another account Fail(3) Account is already logged in Fail(4) Password is incorrect	Success	Welcom, <username>
		Fail(1)	Username not found
		Fail(2)	You already logged in as <username>
		Fail(3)	Someone already logged in as <username>
		Fail(4)	Wrong password
logout	Logout account Fail(1) User not logged in Fail(2) User is in game room	Success	Goodbye, <username>
		Fail(1)	You are not logged in
		Fail(2)	You are already in game room <game room id>, please leave game room

create public room <game room id>	Create game room that is public to everyone Fail(1) User not logged in Fail(2) User is in game room already Fail(3) Game room id is exist already Please note that you have to send this request and receive the response with **TCP**	Success	You create public game room <game room id>
		Fail(1)	You are not logged in
		Fail(2)	You are already in game room <game room id>, please leave game room
		Fail(3)	Game room ID is used, choose another one
create private room <game_room_id> <invitation code>	Create game room that is private to everyone **Invitation code should be unsigned 32 bit integer** Fail(1) User not logged in Fail(2) User is in game room already Fail(3) Game room id is exist already Please note that you have to send this request and receive the response with **TCP**	Success	You create private game room <game room id>
		Fail(1)	You are not logged in
		Fail(2)	You are already in game room <game room id>, please leave game room
		Fail(3)	Game room ID is used, choose another one
list rooms	List all the game room status in **ascending** order of game room id You need the handle three properties of the	No game room	List Game Rooms No Rooms

		game room	
	game room 1. Public / Private 2. Game room ID 3. Waiting for player / Started Game Your response should start with **List Game Rooms**, and add a number to each lines Please note that you have to send this request and receive the response with **UDP**	At least one game room	**Example** List Game Rooms 1. (Public) Game Room 1 has started playing 2. (Public) Game Room 2 is open for players 3. (Private) Game room 3 is open for players
list users	List all the user status in **alphabetically** order of Username You need the handle three properties of the user status 1. User name 2. User email 3. Online / Offline Your response should start with **List Users**, and add a number to each lines Please note that you have to send this request and receive the response with **UDP**	No game room	List Users No Users
		At least one game room	**Example** List Users 1. Alice<alice@gmail.com> Online 2. Bob<bob@gmail.com> Online 3. Cindy<cindy@gmail.com> Online 4. David<david@gmail.com> Offline 5. Emmy<emmy@gmail.com> Offline
join room <game room id>	Join **public** game room If you join game room successfully, others in the game room will receive the message from server to notify that you join room Fail(1) You are not logged in Fail(2) You are already in game room	Success	Response to you: **You join game room <game room id>** Response to others that joined game room: **Welcome, <user name> to game!**
		Fail(1)	You are not logged in

		Fail(3) Game room ID is not exist Fail(4) Game room is private Please note that you have to send this request and receive the response with **TCP**	Fail(2)	You are already in game room <game room id>, please leave game room
			Fail(3)	Game room <game room id> is not exist
			Fail(4)	Game room is private, please join game by invitation code
invite <invitee email>		Send invitation to invitee **Invitee email is ensure to be in register list and you should not send invitation to yourself** **There is no testcases that invite yourself** Fail(1) Inviter not logged in Fail(2) Inviter did not join any game room Fail(3) Inviter is not game room manager Fail(4) Invitee not logged in	Success	Response to invitee: You receive invitation from <inviter name>**<**<invitier email>**>** e.g. Alice<alice@gmail.com> Response to inviter: You send invitation to <invitee name>**<**<invitiee email>**>**
			Fail(1)	You are not logged in
			Fail(2)	You did not join any game room
			Fail(3)	You are not game room manager
			Fail(4)	Invitee not logged in
list invitations		List invitations in **ascending** order of GameRoomID Your response should start with **List Invitations**, and add a number to each lines	No invitation	List invitations No invitations
			At least one invitation	List invitations 1. <inviter name>**<**<invitier email>**>** invite you to join game room <game room id>, invitation code is <invitation code> E.g.

			1. Alice<alice@gmail.com> invite you to join game room 101, invitation code is 12345
accept <invitee email> <invitation code>	Join **private** game room If you join game room successfully, others in the game room will receive the message from server to notify that you join room Fail(1) You are not logged in Fail(2) You are already in game room Fail(3) Invitation not exist There are some reasons lead to Fail(3), 1.invitee is not invited 2. invitee is invited but the inviter leave the game room, so the invitation is expired Fail(4) Invitation code is incorrect Fail(5) Game is started **Invitee email is ensure to be in register list** Please note that you have to send this request and receive the response with **TCP**	Success	Response to you: **You join game room <game room id>** Response to others that joined game room: **Welcome, <user name> to game!**
		Fail(1)	You are not logged in
		Fail(2)	You are already in game room <game room id>, please leave game room
		Fail(3)	Invitation not exist
		Fail(4)	Your invitation code is incorrect
		Fail(5)	Game started, you can't join now
leave room	Leave game room **If you are game room manager whom create game room, others are forced to leave game room if you leave** **As soon as someone leaves the game room, the game is forced to end** Sucess(1) If you are	Success(1)	**Response to you**: You leave game room <game room id> **Response to others**: Game room manager leave game room <game room id>, you are forced to leave too
		Success(2)	**Response to you**: You leave game room <game room id>, game

	game room manager		ends
			Response to others: <user name> leave game room <game room id>, game ends
	Success(2) If you are not game room manager and game has started		
	Success(3) If you are not game room manager and game has not started	Success(3)	**Response to you**: You leave game room <game room id>
			Response to others: <user name> leave game room <game room id>
	Fail(1) You are not logged in		
	Fail(2) You are not in game room	Fail(1)	You are not logged in
	Please note that you have to send this request and receive the response with **TCP**	Fail(2)	You did not join any game room
start game <number of rounds> <guess number>	Start a **1A2B** game.	Success	Game start! Current player is <Current player name>
	<number of rounds> is the number of rounds of this game, after all the player guess once, the game is moved to the next round, therefore each player can guess <number of rounds> times.	Fail (1)	You are not logged in
		Fail (2)	You did not join any game room
	<guess number> is the number to guess for this game.	Fail(3)	Please enter 4 digit number with leading zero
	The order of guessing number is the same as the order that player join game room.		
	Fail(1) You are not logged in		
	Fail(2) You did not join any game room		
	Fail(3) You should provide 4 digit number		
	Note that <guess number> is optional, if it is not given, random 4 digit number is generated		
	Note: You have to send this command and get associated message by		

guess <guess number>	Guess number! The rule of guess result is the same as HW1 Fail(1) If your guess number is not 4 digit number **The input is ensure to be number with leading zero and the input size is less than 10** **Non 4 digit number is not count in the guess times (number of rounds) in the game** After each guess, you need to output the **current player** message to all the players **Note:** You have to send this command and get associated message by **TCP**	Success (Not Bingo)	**Example** Alice guess '0301' and got '3A0B'
		Success (No chances)	**Example** Bob guess '3214' and got '2A2B' **Game ends, no one wins**
		Success (Bingo)	**Example** Alice guess '0302' and got Bingo!!! **Alice wins the game, game ends**
		Current player	Game start! Current player is <Current player name>
		Fail(1)	Please enter 4 digit number with leading zero
exit	Close connection. 1. Leave game 2. Logged out **When receiving EOF from client, it is the same as exit** **Note:** You have to send this command by **TCP**		

Scenario

You can found sample testcases and the correct output in *testcases/* and *testcases_correct/*

The output format in pdf might not be correct, **please refer to sample testcases**

Make sure your output is the same as sample testcases output, because all the output format can be found in *testcases_correct/*, **you wont get any credit of that testcase even you just forget to output one comma**

Please make sure the flag `SO_REUSEADDR` is set and your code should have the same result whatever times you run it.

Ask us from Teams if you have any questions!