Task 4: Game Server

1 Task

Write a TCP game server called `linear` which takes three positional arguments: `port_number`, `num_players` (values between 2 and 5, inclusive), `board_size` (values between `num_players` and `5*num_players`, inclusive). The server should handle a multiplayer game with the following rules:

- board is 1D linear - players can move only in left or right direction
- at the beginning, the position of all players is random but unique
- a player can either scan the board or move
- scanning the board means getting full board information, i.e. positions of all players. Without scanning, the player moves blindly and doesn't know positions of others
- a player can move by 1 or 2 steps in right or left direction
- if a player A moves into the field occupied by player B, player B loses the game (if two players move "simultaneously" whoever is first is decided by who claims the semaphore first)
- at the beginning everyone receives positions of all players
- moving outside the board results in a loss
- the last player on the board wins the game

Technical details:

- each board tile is protected by a separate unnamed semaphore
- **you must use POSIX semaphores**
- whenever board is printed the format should be single space (for empty tiles) or player number for each tile separated by `|`, e.g. `|1| | |2|3| | |`.

More details are provided in Stages description. You can use e.g. `netcat` as the client - don't implement it by yourself.

1.1 Lab part

- Stage 1 - **2 points** - server listens on `localhost:<port_number>` and accepts client connections, responds to each with the following message: `"you are player#<PLAYER_NUMBER>. Please wait..."` and closes the connection. You can use any unique `PLAYER_NUMBER`, e.g. autoincrement each new client (1, 2, 3, ...)
- Stage 2 - **3 points** - when `num_players` connect to the server it spawns `num_players` threads, each responsible for handling single player. Threads send a message to their corresponding clients: `"The game has started."`, receive messages from players and print them to the stdout. Messages may be one of the following: -2, -1, 0, 1, 2. Other messages must be ignored.
- Stage 3 - **2 point** - When the game starts, positions of players are randomized (uniquely) and threads initially send all positions to all players (once).

1.2 Home part

- Stage 4 - **3 points** - Player messages handling: positive values (1 or 2) move player to the right by 1 or 2 fields, negative values (-1 or -2) move player to the left. `0` scans the board (i.e. the server returns all positions of all players). Moving beyond borders of the board results in losing the game - send `"you lost: <REASON>"` (where `<REASON>` is always `"you stepped out of the board"`) message to the player who lost and close the connection. No other win/lose logic at this stage.
- Stage 5 - **3 points** - If player moves into a field occupied by another player, the other player loses the game (with `<REASON>` being `"PLAYER#<N> stepped on you"`. Last one standing is a winner (receives a message `"You have won!"`). After the game is over, the server disconnects the winner and waits for new `num_players`
- Stage 6 - **1 point** - Server shuts down gracefuly on SIGINT. The server refuses connections to new players when the game is currently going on.

2 UPLOAD

Please upload your solution to: `/home2/samba/sobotkap/unix/` You have to upload each stage immediately after finishing it. You upload to a network share via `ssh.mini.pw.edu.pl` server:

```
scp user.etapX.tar.bz2 user@ssh.mini.pw.edu.pl:/home2/samba/sobotkap/unix/
```

Please name your stages files according to the schema: `LOGIN.etapN.tar.bz2(.gz)`

All programs must build using single `make` (with no arguments) command

3 THE STATEMENT

By decree 27/2020 of University Rector you must add the following statement to the uploads:

I declare that this piece of work which is the basis for recognition of achieving learning outcomes in the OPS course was completed on my own. [First and last name] [Student record book number (Student ID number)]

Please add it as comment at the beginning of each source file you upload. Replace square brackets with your data.