Raport tehnic

Pîntea Andreea

UAIC, Facultatea de Informatică

1 Introducere

Am ales să fac proiectul QuizzGame, iar acesta reprezintă o competiție intre un număr nelimitat de jucători(clienți). Aplicația vă urma un model de arhitectura client-server, unde serverul coordonează jocul, având acces la o baza de date, iar clienții răspund la întrebări. Jucătorii vor primi puncte daca răspund corect la întrebările date în intervalul de timp precizat, iar la final, cel cu cele mai multe puncte câștiga. Este important de precizat că nu exista o departajare intre jucătorii cu același scor final. Jocul este făcut să continuie chiar daca unul din jucători alege să iasă din joc.

2 Tehnologii Aplicate

Voi utiliza o implementare orientata pe conexiune, folosind un TCP concurent, folosindu-mă de multithreading pentru a lega fiecare client de server. Astfel se realizează concurența serverului, întrucât acesta poate să suporte clienți multipli fără să se blocheze. De asemenea, comunicarea client-server se realizează cu o conexiune de tip socket. Pentru a stoca întrebările și răspunsurile din QuizzGame, precum și utilizatorii logati, voi folosi o bază de date SQLite cu două tabele (Questions și users).

3 Structura Aplicației

Pentru a modela o aplicație care implică un server multithreading care coordonează un joc de întrebări și răspunsuri între clienți, precum și interacțiunea cu o bază de date SQLite, este necesar să definim câteva concepte cheie și să realizăm o diagramă detaliată a proiectului.

Server: Gestionează conexiunile cu clienții și coordonează jocul. Utilizarea thread-urilor pentru a permite gestionarea mai multor clienți simultan. Interacțiunea cu baza de date pentru a înregistră clienții și pentru a obține întrebările și a verifica răspunsurile. Acorda punctaj corespunzător fiecărui jucător și trimite mesaj aferent fiecărui client.

Client: Se conectează la server și primește întrebări de la acesta. Trimite răspunsurile înapoi la server, mai apoi primește mesaj de validare a răspunsului.

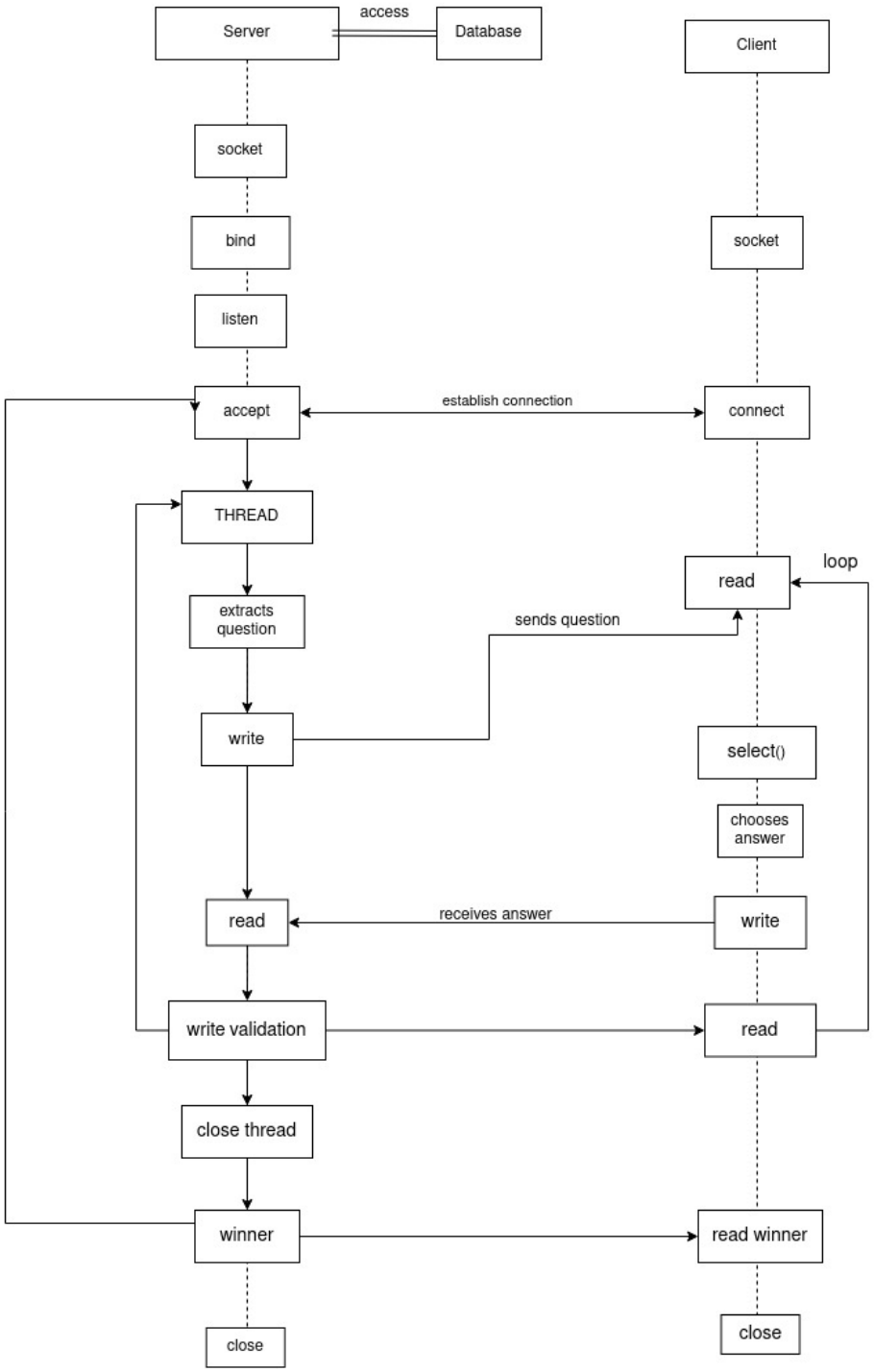

Fig. 1. Diagrama QuizzGame

Bază de Date SQLite: O tabela ce stochează întrebările și răspunsurile, având o coloana cu răspunsul corect. Încă o tabela care tine cont de utilizatorii înregistrați. Aceștia pot fi adaugați sau pot fi regăsiți în tabela.

4 Aspecte de Implementare

O parte semnificativa a programului este cea care surprinde comportamentul threadurilor care leaga clientii. Functiile login(thPlayer) si gameLogic(thPlayer) sunt implementate dupa specificatiile de mai sus.

```
void *treat(void *arg)
{
    struct Thread thPlayer;
    thPlayer = *((struct Thread *)arg);
    printf("[thread- %d] Waiting for players to login
       ...\n", thPlayer.idThread);
    fflush(stdout);
    pthread_detach(pthread_self());
    login(thPlayer);
    gameLogic(thPlayer);
    close(thPlayer.cl);
    return (NULL);
}
```

Functia de login este relevanta in modul in care am verificat daca am un client nou folosind functia newUser, mai apoi dac aera cazul i-am adaugat pe cei noi in tabela de utilizatori din baza de date cu functia addUser. De asemenea am functia verifyPassword care returneaza 1 daca clientul a introdus parola corecta si 0 invers.

```
void login(struct Thread thPlayer)
{
    int connected = 0;
    do
    {
        initPlayerInfo(thPlayer.cl, count_clients); //
           setat username si parola
        connected = newUser(thPlayer.idThread);
        if (connected == 1)
            connected = addUser(thPlayer.idThread);
        else if (connected == 0)
            connected = verifyPassword(thPlayer.
               idThread);
        if (write(thPlayer.cl, &connected, sizeof(int))
            <= 0)
        {
```

```
14              perror("Writing message to client failed.\n
                    ");
15          }
16      } while (connected == 0);
17      if (connected == 1)
18      {
19          printf("User has logged in!\n");
20          return;
21      }
22      fprintf(stderr, "Could not connect client!!\n");
23  }
```

O sectiune inovanta a codului din client este cea unde in momentul in care este introdus un mesaj gresit sintactic, printr-un goto se intoarce la momentul in care asteapta un raspuns de la client.

```
1       loop:
2           memset(buff, 0, MAX_BUFF - 1);
3           fgets(buff, sizeof(buff), stdin);
4           buff[strcspn(buff, "\n")] = '\0';
5           if (write(sd, buff, MAX_BUFF - 1) <= 0)
6                perror("Error");
7           if (strcmp(buff, "exit") == 0)
8           {
9               exited = true;
10              printf("You chose to exit the quizzgame!\n"
                    );
11              flush(stdout);
12              break;
13          }
14          else
15          {
16              if (strcmp(buff, "a") != 0 && strcmp(buff,
                    "b") != 0 && strcmp(buff, "c") != 0 &&
                    strcmp(buff, "d") != 0 && strcmp(buff,
                    "") != 0)
17              {
18                  memset(buff, 0, MAX_BUFF - 1);
19                  if (read(sd, &buff, MAX_BUFF - 1) <= 0)
20                          perror("Error");
21                  printf("%s", buff);
22                  fflush(stdout);
23                  goto loop;
24              }
25              ...
26          }
```

De asemenea este important de notat ca am folosit select() pentru a cronometra fiecare raspuns al clientilor

```
1      FD_ZERO(&active_fds);
2      FD_SET(0, &active_fds);
3      timeout.tv_sec = 10;
4      timeout.tv_usec = 0;
5      int result = select(1, &active_fds, NULL, NULL, &
           timeout);
6      if (result < 0) perror("error");
7      else if(result) {
8          // bloc executat in limita celor 10 secunde din
               timeout
9      } else {
10         // result = 0 => timpul a expirat
11     }
```

5 Concluzii

Ca îmbunătățire, aș adăuga o interfață grafică unde se poate vedea în timp real fiecare jucător care alege un răspuns, mai apoi fiind vizibile și punctele pe care le-a primit pentru răspunsul dat, iar la final să apară câștigătorul alături de scorul său. De asemenea, aș putea să dau bonusuri pentru jucătorii care răspund în timp record, spre exemplu, dacă timpul de răspuns este 10 de secunde, iar jucătorul X răspunde sub 5 secunde, acesta va primi un bonus de 1 punct.

6 Referinte Bibliografice

- laboratorul 7- threads
- cursul 6- multiplexare
- sqlite
- primitiva select()
- mutexuri