Quizgame

Mihaela Potlog

University Alexandru Ioan Cuza of Iasi

Abstract. Quiz-Game requires multithreading and XML-files parsing .

Keywords: Used technologies · Application Architecture Diagram · Optimisation.

1 Introduction

GameQuiz requires knowledge about the ways in which someone can use XML-files for storing data and synchronization of threads;this is why I chose this project,because I want to learn and practice how to do these things.

2 Used Technologies

2.1 Libraries

GTK+ is a multi-platform(including Linux) toolkit for graphical user interfaces.

Libxml2 It's a XML C parser that can be used on a variety of platforms,including Linux.

2.2 Client/Server model

TCP/IP is a reliable protocol and safer than other protocols because it makes the connection("3-way handshake") and only after this, the data is sent.

3 The Application Architecture

The server will make two kinds of threads with a specific task.

The first thread makes sure that the client logs in or if he is not registered,the thread will register him.After this it sends the client's data (socket-pair descriptors) to the thread responsible for the match.

The other one waits to receive form the "register/log-in"-threads,the socket-pair descriptors.When it receives a number of k clients,the game starts.Using the XML-parser to send the questions to the players,making sure that the game goes well(even if some players quit).

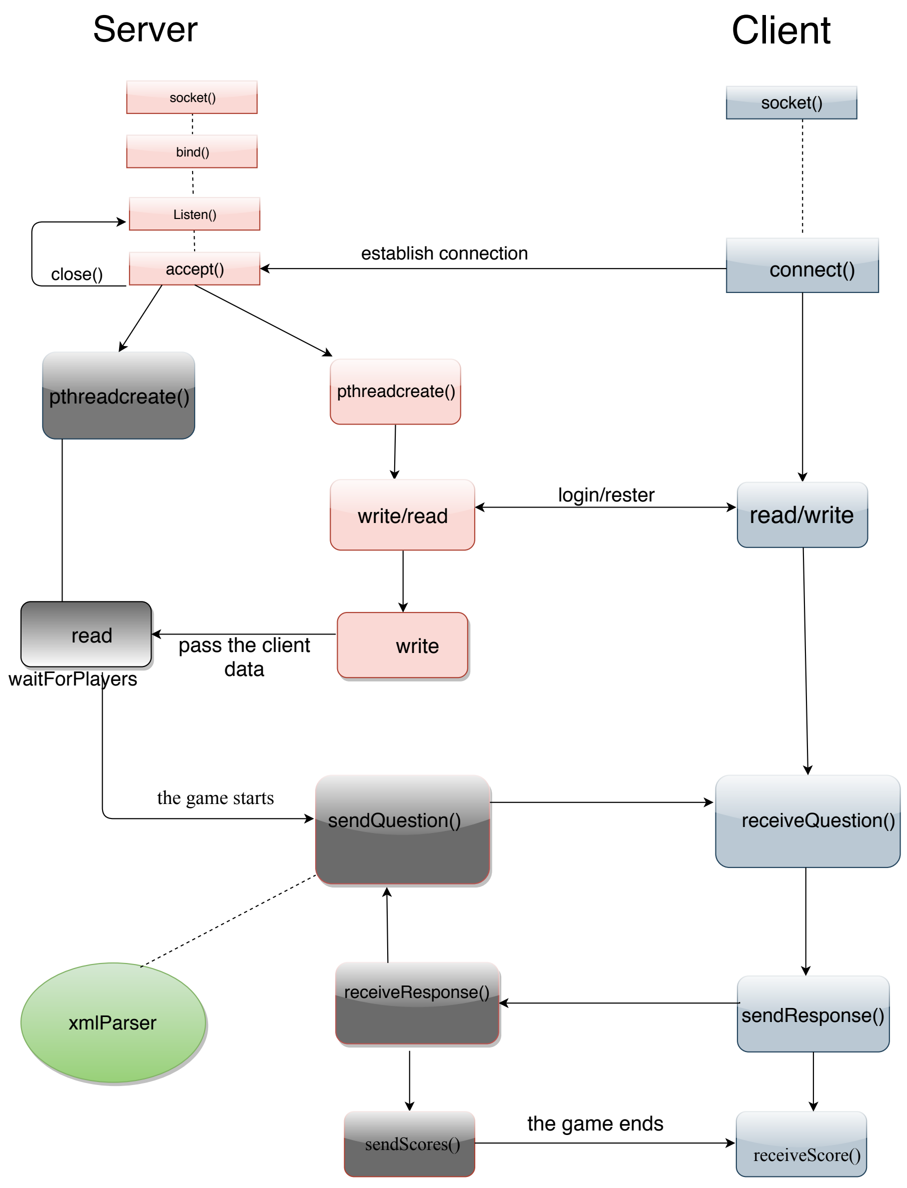

Fig. 1. simplified architecture diagram

4 Implementation details

The client data, could be stored in simple files,for example:username,password,number of won games ,lost games,the score. A new thread is created when it's necessary to begin another game(there is no "waiting room" available).This thread will have to wait for k-1 clients. The game will continue even if a player quits.

```
 while(1){

clients[nrClients++] = accept (sd, (struct sockaddr *) &from,
    &length);
close(clients[nrClients-1]);
//connectAccount-function for log in/registration
error=pthread_create(&threadId,NULL,&connectAccount,clients);
     if(error!=0)
   printf("I can't create thread for login in/register [\%s]",
     strerror(error));

     if(nrClients==1)
          error=pthread_create(&threadId,NULL,&startGame,
     clients);


   if(error!=0)
     printf("I can't create thread for starting the game\n [\%s
     ]",strerror(error));

   if(nrClients==k)
      nrClients=0;
}
```

5 Conclusion

This model of application could be improved by allowing the clients to choose the difficulty of the questions,the offered time or the domain and also they could choose to play with friends by writing their usernames(the server will make a private match).This aspect will create the necessity for other "types" of threads.Another improvement is "learning mode".The client can make a set of questions in order to go through them ,later,when he wants to review what he knows about a specific learned topic .

5.1

A possible solution could involve creating a thread before the while condition.This thread would be responsible for communicating with the "log-in"-threads and passing the client to an already "match"-thread or creating when it is necessary, a new thread for sustaining a match(eliminating the "rainnyday" use-cases mentioned before).

”Rainy-day” cases: If a client takes to long to log in,the ”match”-thread assigned to him,will have to wait ;or there are no new clients to fill the required number of players,the thread will also have to wait.A solution could be communicating with the server,choosing how long it will wait,however in my opinion this is not a good solution,because it makes a slow server.

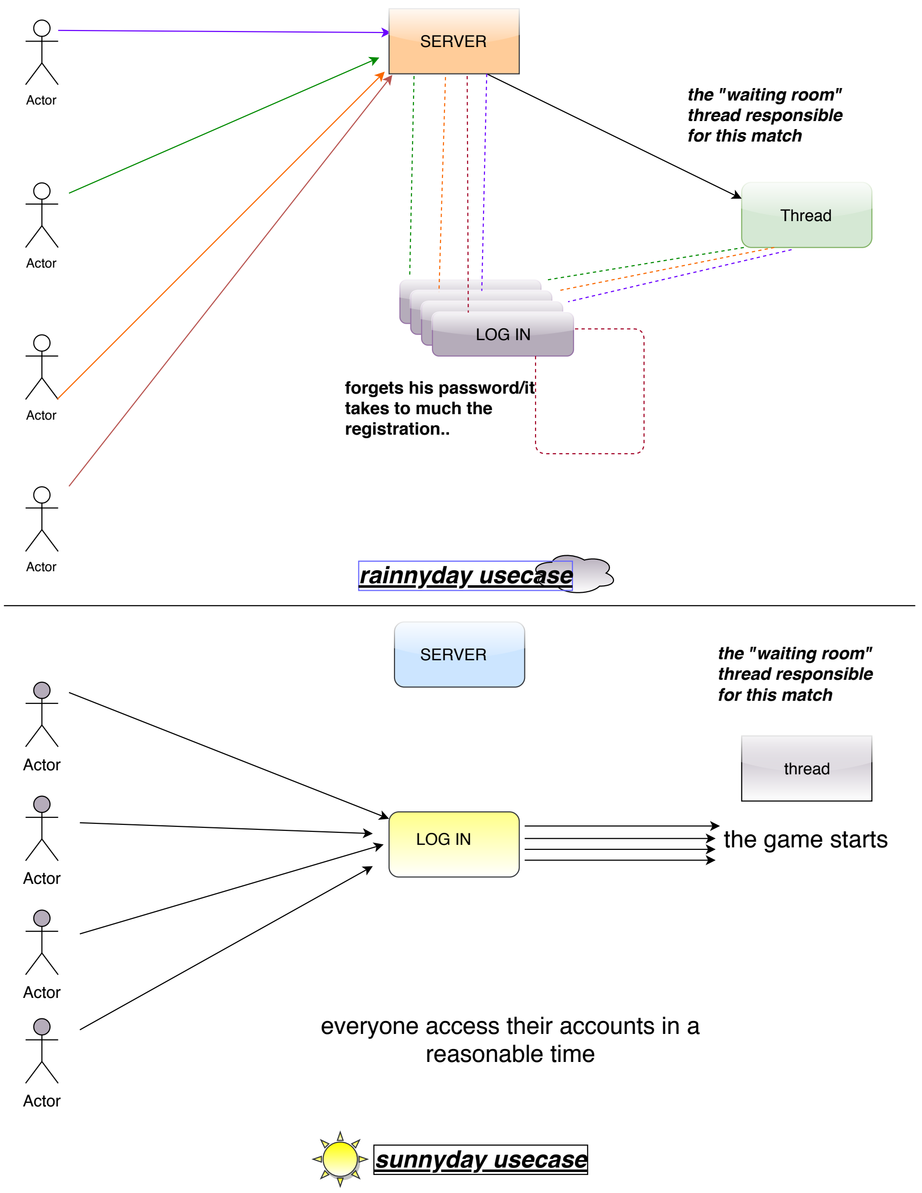
Actor
Actor
Actor
Actor
SERVER
the "waiting room" thread responsible for this match
Thread
LOG IN
forgets his password/it takes to much the registration..
rainnyday usecase
SERVER
the "waiting room" thread responsible for this match
thread
Actor
Actor
Actor
Actor
LOG IN
the game starts
everyone access their accounts in a reasonable time
sunnyday usecase

Fig. 2. Usecase diagram”

References

1. XML parser: http://xmlsoft.org/
2. GTK Homepage, https://www.gtk.org/download/linux.php.
3. XML parser example: http://simplestcodings.blogspot.com/2010/09/simple-xml-parser-in-c-using-libxml.html