

CONNECT4

Mandici Rahela

Universitatea Al. I. Cuza, Facultatea de Informatică

1 Introducere

Aplicația creată va simula jocul CONNECT4. Este o aplicație de tipul client/server, unde fiecare client reprezintă un jucător. Serverul se va ocupa de formarea perechilor de jucători, de actualizarea tabelor de joc după fiecare mutare și de actualizarea scorului fiecărei perechi de jucători, după fiecare repriză.

2 Tehnologii utilizate

Ca și protocol de comunicare între client și server, a fost folosit protocolul de transport TCP.

Deoarece fiecare mutare, cât și ordinea în care acestea sunt trimise, determină rezultatul jocului, s-a dorit un protocol de comunicare care să asigure atât primirea datelor în ordinea în care acestea au fost trimise, precum și corectitudinea acestora. De asemenea, pierderea unei mutări ar bloca jocul, deoarece unul dintre clienți ar rămâne blocat în așteptarea mutării partenerului său de joc. De aceea este important ca toate datele trimise atât de server, cât și de client, să ajungă la destinație.

Aceste servicii sunt oferite de protocolul TCP, de aceea a fost ales ca protocol de transport în aplicația CONNECT4.

3 Arhitectura aplicației

3.1 Descrierea diagramei

Aplicația conține un server care așteaptă clienți la portul 2019. Pentru fiecare doi clienți conectați, serverul crează un proces copil care va reprezenta serverul pentru cei doi clienți. Comunicarea dintre cei doi clienți se realizează prin intermediul procesului server fiu, excepție fiind mesajul trimis de către serverul părinte fiecărui client la conectare, și mesajul trimis clientului pentru care nu s-a găsit partener de joc.

3.2 Diagrama aplicației

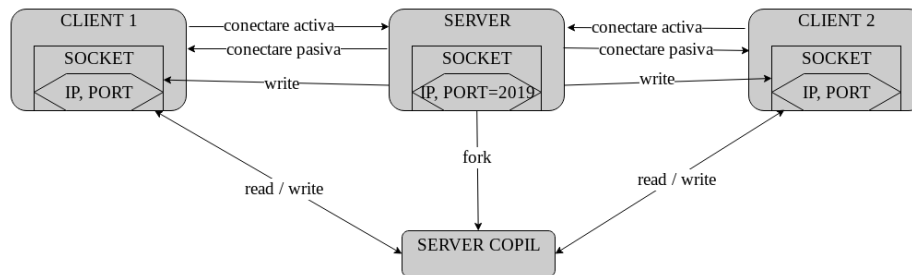


Figura 1. Diagrama proiectului **CONNECT4**

4 Detalii de implementare

Aplicația a fost scrisă în limbajul de programare C, fiind formată din două fișiere .c, unul reprezentând *serverul*, iar celălalt, *clientul*.

Serverul a fost implementat după modelul serverului TCP concurrent.

4.1 Protocol de utilizare

Fiecare client conectat va introduce un username ce va fi comunicat celuilalt jucător.

După ce i se va comunica culoarea sa și numele celuilalt jucător, clientul va aștepta începerea jocului.

După începerea jocului, atunci când va fi rândul lui, jucătorul va introduce fie un număr de la 1 la 7, ce va reprezenta coloana pe care va face el mutarea, fie numărul 0 ce va reprezenta abandonarea jocului.

La finalul unei reprize, jucătorul va fi întrebat dacă dorește continuarea jocului cu încă o repriză.

La finalul fiecărei reprize, jucătorului i se va comunica scorul în urmatorul format scorul său : scorul partenerului de joc.

4.2 Descrierea implementării

Deoarece jocul este proiectat pentru doi jucători, serverul va crea un nou proces (prin apelul `fork()`) în momentul în care s-au conectat doi clienți. Procesul nou creat se va îngriji de comunicarea celor doi clienți, astfel încât jocul dintre aceștia să se desfășoare conform regulilor.

La conectarea fiecărui client, serverul părinte transmite mesajul:

CONNECT4

Alegeți un username:

Dacă în urma conectării unui client, timp de 10 sec nu se mai conectează alt client, serverul părinte trimite jucătorului 1 mesajul:

Nu s-a găsit niciun partener de joc pentru dvs. Invitați un prieten la joc!

și se va aștepta în continuare conectarea celui alt jucător. Pentru implementarea acestui scenariu a fost folosită funcția *select()*.

De preluarea username-urilor se va ocupa procesul fiu (serverul fiu) creat în urma conectării a doi jucători. Serverul fiu va prelua username-urile jucătorilor de care este responsabil el.

După ce fiecare jucător a introdus un username, serverul fiu va trimite fiecăruia dintre ei un mesaj informativ în care le va comunica împotriva cui vor concura, culoare pe care o vor avea, și care dintre ei va începe jocul.

Exemplu pentru primul jucător:

Veți juca împotriva jucătorului: username2

Veți avea culoarea rosu.

Dvs veți deschide jocul.

Succes!

Exemplu pentru al doilea jucător:

Veți juca împotriva jucătorului: username1

Veți avea culoarea galben.

Celălalt jucător va deschide jocul.

Succes!

După acest mesaj, serverul fiu va mai trimite câte o valoare de tip int, prin care clienții vor primi codul culorii ce îi va reprezenta.

În urma acestor mesaje informative, li se va afișa fiecărui jucător tabla de joc, iar fiecare jucător va trimite serverului fiu mutarea sa. Ordinea va fi dată de culoarea jucătorului: întâi va trimite jucătorul de culoare roșie, iar cel de culoare galben va trimite abia după ce a primit mutarea partenerului de joc. Ordinea va fi anunțată printr-un mesaj ce va fi afișat mereu, pentru fiecare jucător, deasupra tablei de joc.

Înainte de a o trimite celui alt jucător, serverul fiu verifica dacă mutarea respectivă este câștigătoare. Dacă este câștigătoare, serverul fiu informează jucătorii că cineva a câștigat, prin codul 91 (jucătorul 1 a câștigat sau a fost remiză după mutarea sa) sau 92 (jucătorul 2 a câștigat sau a fost remiză după mutarea sa). Atunci când clientul 1 primește codul 91 sau 92, el va ști că nu este vorba despre o mutare. Dacă primește codul 91, înseamnă că ultima sa mutare a fost câștigătoare sau e remiză, deci va aștepta un mesaj informativ. Dacă primește codul 92, înseamnă că mutarea jucătorului 2 a fost câștigătoare sau e remiză, deci va aștepta mutarea jucătorului 2 și apoi un mesaj informativ. Similar pentru clientul 2, dacă primește codul 92 înseamnă că el a câștigat sau e remiză, dacă primește codul 91 înseamnă că celălalt jucător a câștigat sau e remiză; pentru fiecare dintre cazuri va avea un comportament similar cu cel prezentat la clientul 1.

Mesajul informativ trimis de serverul fiu atunci când unul dintre jucători câștigă va conține și întrebarea *Continuati cu inca o repriza? (y/n)*. Dacă un jucător

trimite n si celălalt y , atunci jocul se va închide pentru primul jucător, iar celuilalt jucător i se va trimite un mesaj prin care i se transmite că jocul se încheie. Dacă ambii jucători vor trimite y , jocul va continua cu încă o repriză. Dacă un jucător vrea să părăsească jocul înainte de finalul acestuia, va trimite mutarea 0. Atunci când serverul fiu primește de la un jucător codul 0, el va ști că nu este vorba despre o mutare, ci jucătorul respectiv a ieșit din joc. În această situație, serverul fiu trimite celuilalt jucător codul 0, apoi un mesaj de informare. Când un client primește codul 0, el va ști că nu reprezintă o mutare, și va aștepta un mesaj informativ de la serverul fiu, apoi se va deconecta și el.

5 Concluzii

Pentru îmbunătățirea aplicației, s-ar putea adăuga interfață grafică astfel încât jocul să poată fi vizualizat mai bine.

De asemenea, aplicația ar fi îmbunătățită dacă ar fi implementată o modalitate prin care un client care vrea să continue să joace după ce partenerul său de joc a părăsit jocul, să poată fi conectat cu un client nou și jocul să continue.

Bibliografie

<https://profs.info.uaic.ro/~computernetworks/cursullaboratorul.php>
https://en.wikipedia.org/wiki/Connect_Four
<http://web.theurbanpenguin.com/adding-color-to-your-output-from-c/>
<https://stackoverflow.com/questions/17318886/fflush-is-not-working-in-linux/23885008#23885008?newreg=45e52b6e2c7e450dabaa2f5504b79a1c>