

Proiect - ChessS (B)

Popa Răzvan-Gabriel, Grupa A6, Anul 2

Facultatea de Informatică Iași

1 Introducere

1.1 Descriere proiect

Proiectul "Chess" propune conceperea aplicație server care pune la dispoziție o tablă de șah și supervizează desfășurarea fiecărei partide de șah, acționând ca un punct central la care clienții din rețea se conectează. Serverul trebuie să determine momentul în care jocul s-a terminat și anunță câștigătorul. Clienții vor avea la dispoziție un meniu din care pot decide când să înceapă un meci sau să iasă din aplicație și tabla de șah.

1.2 Motivație

Motivul pentru care am ales acest proiect este acela că îmi place șahul, joc de aproximativ 4 ani (cel mai mult online dar și cu bunicul meu cateodată), am citit și o carte despre deschideri și strategii în șah, și îmi doresc prin intermediul acestui proiect să învăț cum se programează un joc în rețea.

2 Tehnologii utilizate

Pentru acest proiect am ales protocolul de comunicație TCP, deoarece:

- Acesta permite transferul de date în stream-uri de biți, prevenind pierderea sau coruperea datelor, iar în cazul protocolului UDP există riscul de a se pierde parțial informația;
- TCP realizează verificări pentru erori, în timp ce UDP verifică dacă sunt erori ce apar în pachete cu date, dar le elimină pe cele corupte;
- Prin TCP datele se transmit după acceptarea și formarea conexiunii, fiind un protocol connection-oriented, în timp ce UDP este de tip connection-less

Aplicația fiind bazată pe conectarea clienților la server și transmiterea de informații (dorința de a începe un meci, mutarea unei piese etc.) ce nu se doresc a fi pierdute, protocolul TCP reprezintă cea mai bună alegere. Fiecare client trimite o cerere de conectare la server, acesta acceptă cererea clientului, iar după formarea acestei conexiuni, clientul poate trimite pe socket date, iar serverul să răspundă în urma procesării informației primite de la client (sa porneasă un meci de exemplu).

3 Arhitectura aplicației

3.1 Conceptele implicate

1. Client

Clientul se va conecta la server de la linia de comandă folosind formatul:
./client -adresa IP la care se afla server-ul- -portul server-ului-.

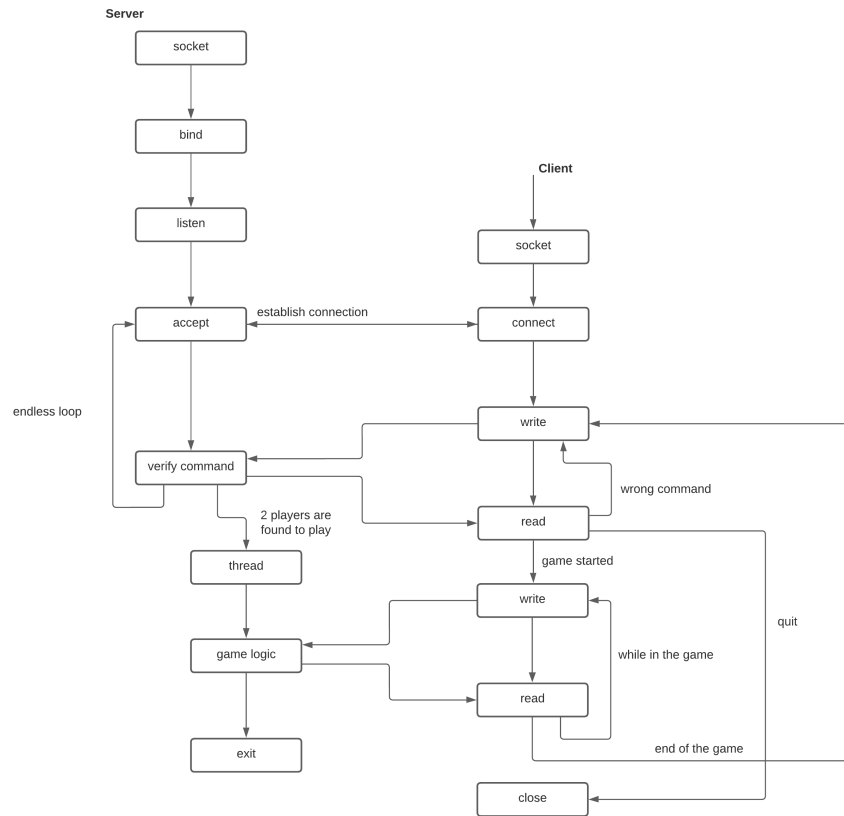
Dupa ce clientul primește confirmarea ca sa conectat la server, îi va apărea în consolă două opțiuni: una de a intra într-un joc si cealaltă de a se deconecta de la server. Odată intrați în meci jucătorii vor putea vedea tabla de sah, piesele lor (litere mare(alb) / litere mici(negre)) și vor putea muta pe rand piesele. Pentru acest lucru, jucătorii trebuie să scrie poziția la care se află piesa inițial și poziția la care vor să o mute.

2. Server

Serverul de tip TCP, ascultă și asteapă conexiuni din partea clienților. Dacă clientul tastează comanda "play" acesta va fi pus într-o listă de asteptare până când un al doilea client dorește să intre într-un joc. Dacă o pereche de jucători este găsită atunci meciul va porni. Iar dacă clientul apasă pe butonul "quit" serverul va termina conexiunea cu acesta.

Meciul se va desfășura ca o partida normală de șah, culoare pieselor jucătorilor vor fi alese aleatoriu, albul începând primul, fiecare jucător având dreptul la câte o mutare pe tură. Dacă mutarea dorită nu este validă (calul este mutat două pătrățele în față, regele se afla în șah și jucătorul mută altă piesă și nu iese din șah etc) atunci piesa mutată își va pastra poziția de după mutare. Jocul se va termina dacă se indeplinesc una din următoarele condiții (un jucator castigă meciul prin șah mat, meciul se termină la egalitate (nu mai sunt suficiente piese pe masă sau un jucător nu mai are mutări posibile și regele lui nu se află în șah) sau unul dintre jucători pierde conexiunea la server)

3.2 Diagrama aplicației



Clienții vor trebui prima oară să se conecteze la server prin intermediul adresei IP și al portului iar dacă cerere lor este acceptată atunci vor putea folosi aplicația. În timpul unui meci, clienții vor trimite informația referitoare la mutarea unei piese prin intermediul tastării acesteia (exemplu: scriem comanda "d2 d3") iar serverul o va verifica dacă este validă și va trimite un răspuns înapoi clientului (dacă mutarea este validă piesa va fi mutată pe poziția dorită și tabla de joc va fi actualizată pentru ambii jucători, în caz contrar piesa va rămâne în poziția inițială).

4 Detalii de implementare

4.1 Secvente de cod

Cautarea jucatorilor care doresc sa intre intr-un meci

```
for (fd = 0; fd <= nfds; fd++) /* parcurgem multimea de descriptori */
{
    if (fd != sd && FD_ISSET(fd, &readfds) && v[fd] == 0) //vedem daca un socket de citire este pregatit
    {
        char mesaj[1000];
        bzero(mesaj, 1000);
        if (read(fd, mesaj, 1000) < 0)
        {
            perror("[server] Eroare la read().\n");
            return errno;
        }

        if (strstr(mesaj, "quit"))
        {
            printf("[server] S-a deconectat clientul cu descriptorul %d.\n", fd);
            fflush(stdout);
            close(fd); /* Inchidem conexiunea cu clientul */
            FD_CLR(fd, &actfds); /* scoatem si din multime */
        }

        if (strstr(mesaj, "play"))
        {
            v[fd] = 1; //jucatorul asteapta sa intre in meci
            write(fd, "Cautare jucator...", 1000);
        }
    }
}

if (cautare_jucator_1(v, nfds) != -1 && cautare_jucator_2(v, nfds) != -1) //daca doi jucatori asteapta sa intre intr-un meci, atunci acesta va incepe
{
    jucatorul_unu = cautare_jucator_1(v, nfds);
    jucatorul_doi = cautare_jucator_2(v, nfds);

    v[jucatorul_unu] = 2; //jucatorii se aflam in meci
    v[jucatorul_doi] = 2;

    pthread_t tid;
    pthread_create(&tid, NULL, Game, (void *)&tid); //logica jocului
}
```

Logica jocului

Jucatorul A

```
while (1)
{
    if (a_round == 1)
    {
        verificare = 0;
        if (a_fd != sd)
        {
            if (Mutarea_facuta(a_fd, b_fd, 'a', s, NewTable) == -1) //Jocul a luat sfarsit! Jucatorul B a castigat!
            {
                v[jucatorul_unu] = 0;
                v[jucatorul_doi] = 0;
                return 0;
            }
            else if (verificare == 1) //mutarea este valida
            {
                codificare_tabla_A(s, NewTable);

                if (write(a_fd, s, 1000) < 0)
                {
                    perror("[server] Eroare la write() catre client.\n");
                    return 0;
                }

                a_round = 0;
                b_round = 1;
                verificare = 0;
            }
            else //mutarea nu este valida si jucatorul trebuie sa introduca una noua
            {
                codificare_tabla_A(s, NewTable); //informatiile despre tabla vor fi trimise catre client printr-un vector char

                if (write(a_fd, s, 1000) < 0)
                {
                    perror("[server] Eroare la write() catre client.\n");
                    return 0;
                }
            }
        }
    }
}
```

Jucatorul B

```
if (b_round == 1)
{
    verificare = 0;
    if (b_fd != sd)
    {
        if (Mutarea_facuta(b_fd, a_fd, 'b', s, NewTable) == -1) //jocul sa terminat! Jucatorul A a castigat!
        {
            v[jucatorul_unu] = 0;
            v[jucatorul_doi] = 0;
            return 0;
        }
        else if (verificare == 1) //mutarea este valida
        {
            codificare_tabla_B(s, NewTable);

            if (write(b_fd, s, 1000) < 0)
            {
                perror("[server] Eroare la write() catre client.\n");
                return 0;
            }
            b_round = 0;
            a_round = 1;
            verificare = 0;
        }
        else //mutarea nu este valida si jucatorul trebuie sa introduca una noua
        {
            codificare_tabla_B(s, NewTable); //informatiile despre tabla vor fi trimise catre client printr-un vector char

            if (write(b_fd, s, 1000) < 0)
            {
                perror("[server] Eroare la write() catre client.\n");
                return 0;
            }
        }
    }
}
```

4.2 Scenarii de utilizare

Aplicația intenționează să ofere utilizatorilor modalitatea de a juca meciuri de șah impotriva altor oameni fără a fi nevoie de o tablă de șah sau de a fi amandoi prezenți în același loc. Odată ce un client intră în aplicație va fi întâmpinat de un meniu din care va putea alege două variante. Când doi jucători se află în același timp în faza de cautare al unui meci, serverul va porni un acest meci iar cei doi vor putea muta simultan piesele dorite. Dacă un jucător tastează cuvântul "surrender" atunci acesta va pierde meciul și va intra din nou în meniu.

5 Concluzii

Aplicația permite clientilor să se conecteze la server și să joace meciuri de șah între aceștia. Clientul poate alege din meniu când să intre într-un meci sau să iasă din aplicație. Odată intrat într-o partidă, clientul poate muta piesele în șah în funcție de regulamentul jocului iar la final serverul va decide un câștigător sau va declara egalitate. Aplicația ar putea fi îmbunătățită prin adăugarea unei opțiuni de a alege cu ce jucător ai dori să ai o partidă ci nu doar să fi bagat într-un meci cu primul jucător disponibil.

6 Bibliografie

References

1. <https://profs.info.uaic.ro/~computernetworks/cursullaboratorul.php>
2. <https://profs.info.uaic.ro/~georgiana.calancea/laboratories.html>
3. <https://www.guru99.com/tcp-vs-udp-understanding-the-difference.html>