

RAPORT TEHNIC

Offline Messenger (B)

Author:
Stoica Marian

Submitted to:
Lect. Dr. Panu Andrei

December 6, 2022

0.1 Introducere

În acest proiect se va implementa o aplicație client/server care va permite schimbul de mesaje între utilizatori. Aplicația va avea următoarele funcționalități:

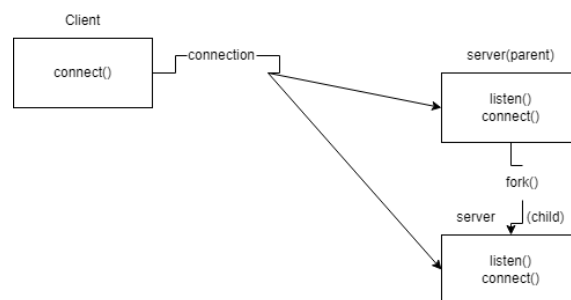
- aplicația va permite transmiterea mesajelor și către utilizatorii offline, acestora din urmă aparându-le mesajele atunci când se vor conecta la server
- utilizatorii vor avea posibilitatea de a trimite un răspuns (reply) în mod specific la anumite mesaje primite
- aplicația va oferi istoricul conversațiilor pentru fiecare utilizator în parte

0.2 Tehnologii utilizate

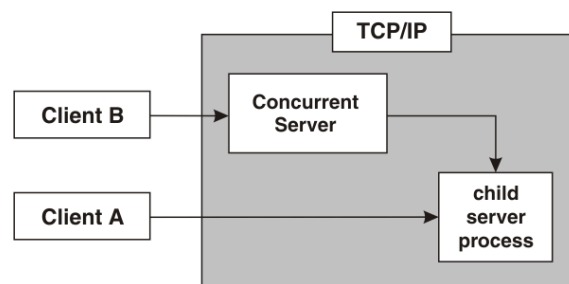
Codul va fi implementat în limbajul C, iar protocolul ales este TCP deoarece oferă siguranță și asigură transmiterea în ordine a datelor. Deoarece serverul nu va fi foarte solicitat, este utilizabilă varianta tradițională de server concurent (un `fork()` per client). Pentru stocarea mesajelor dintre clienți se vor folosi documente XML sau fișiere text.

0.3 Arhitectura aplicației

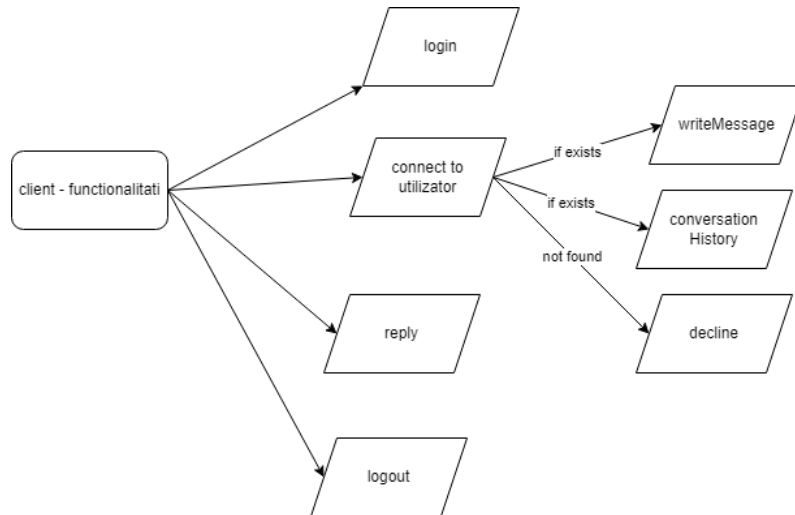
Pentru fiecare client, se va crea un copil care îl va servi:



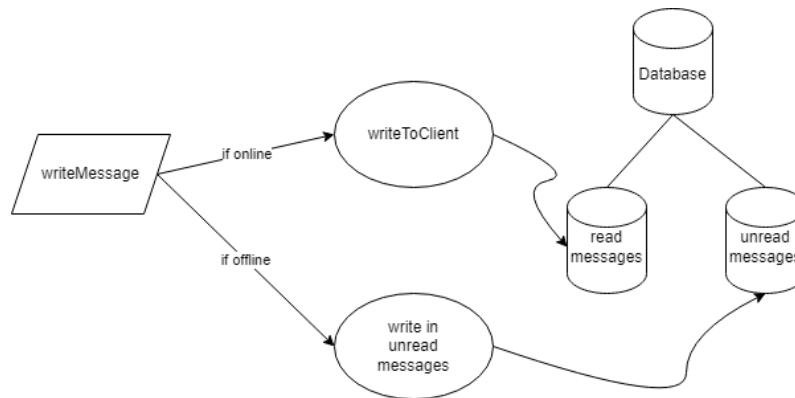
Modelul de server-client TCP:



Arhitectura clientului va fi compusă din următoarele funcționalități:



În particular, funcțiile reply, writeMessage și conversationHistory vor avea acces la baza de date.



În momentul în care un client apelează funcția login se va verifica dacă există mesaje necitite și se vor afișa. Funcția conversationHistory va afișa întâi mesajele citite din baza de date, apoi mesajele necitite aferente clientului conectat cu alt utilizator.

Comunicarea dintre client și server se realizează prin copii creați de server. Clientul va trimite instrucțiunea pe care vrea să o execute, iar interpretarea se va face în server (child). După interpretare, serverul va trimite răspunsul înapoi la client și va aștepta următoarea instrucțiune.

```

/* crearea unui socket */
if ((sd = socket (AF_INET, SOCK_STREAM, 0)) == -1)
{
    perror ("[server]Eroare la socket().\n");
    return errno;
}
/* atasam socketul */
if (bind (sd, (struct sockaddr *) &server, sizeof (struct sockaddr)) == -1)
{
    perror ("[server]Eroare la bind().\n");
    return errno;
}

/* punem serverul sa asculte daca vin clienti sa se conecteze */
if (listen (sd, 1) == -1)
{
    perror ("[server]Eroare la listen().\n");
    return errno;
}
    
```

```

    }
    client = accept (sd, (struct sockaddr *) &from, &length);

    /* eroare la acceptarea conexiunii de la un client */
    if (client < 0)
    {
        perror ("[server]Eroare la accept().\n");
        continue;
    }

    int pid;
    if ((pid = fork()) == -1) {
        close(client);
        continue;
    } else if (pid > 0) {
        // parinte
        close(client);
        while(waitpid(-1, NULL, WNOHANG));
        continue;
    } else if (pid == 0) {
/* citirea mesajului */
        if (read (client, msg, 100) <= 0)
        {
            perror ("[server]Eroare la read() de la client.\n");
            close (client); /* inchidem conexiunea cu clientul */
            continue;      /* continuam sa ascultam */
        }
/* returnam rezultatul clientului */
        if (write (client, rezultat, SIZE_MAX) <= 0)
        {
            perror ("[server]Eroare la write() catre client.\n");
            continue;      /* continuam sa ascultam */
        }
        close (client);
        exit (0);
    }
}

```

0.4 Concluzii

Acest proiect va realiza o aplicatie de mesagerie intre clienti multipli prin folosirea unui protocol TCP concurent si utilizand o baza de date sub forma XML pentru stocarea mesajelor. Aplicatia ar putea fi imbunatatita prin modificarea functiei de login: in loc de afisarea mesajelor necitite imediat ce utilizatorul s-a autentificat, s-ar putea afisa numarul de mesaje necitite si numele utilizatorilor care au trimis respectivele mesaje. In final s-ar putea realiza o interfata grafica asemanatoare cu aplicatia cunoscuta Messenger in care s-ar putea adauga chiar si apeluri vocale intre clienti prin folosirea protocolului UDP.

0.5 Bibliografie

<http://publib.boulder.ibm.com/>
<https://profs.info.uaic.ro/computernetworks/>