**VirtualSoc(A)**

Stefanuca R. Ionel

Facultatea de Informatica, Universitatea Alexandru Ioan Cuza din Iasi

Anul II, grupa A6

[ionel.stefanuca@info.uaic.ro](mailto:ionel.stefanuca@info.uaic.ro)

**Abstract:** Aplicatia permite simularea functionalitatii unei retele sociale. Utilizatorii se vor putea inregistra, posta (postarile fiind publice sau private), vizualiza postarile publice ale altor utilizatori. Sistemul permite si comunicarea private dintre 2 sau mai multe persoane.

**Keywords:** retea sociala, baza de date, client-server, server TCP,socket,fire de executie, mesaje OOB

**1. Introducere**

Acest model client-server permite comunicatea si distribuirea de informatii intre utilizatori. Utilizatorul se va conecta la server cu ajutorul clientului. Isi va putea crea un cont si va putea distribui informatii. Datele vor fi stocate intr-o baza de date. Atat clientul cat si serverul vor fi pornite de la linia de comanda.

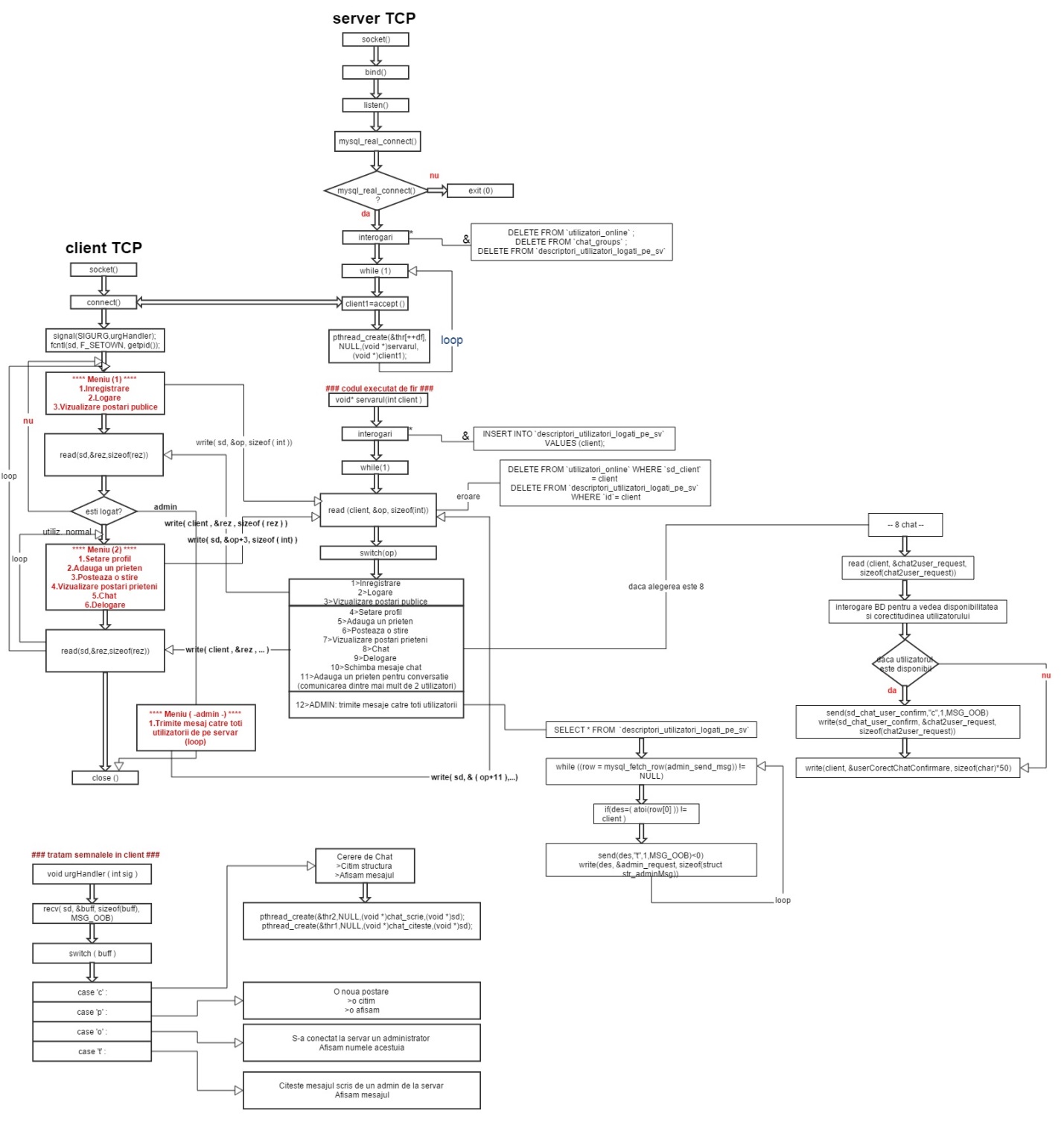
**2. Tehnologii utilizate**

Datele vor fi stocate intr-o baza de date. Aceasta va fi formata din mai multe tabele optimizate. Astfel datele vor fi securizate iar accesul la ele se va face intr-un timp cat mai scurt. Conectarea la baza de date cat si interogarea acesteia se va face cu ajutorul serverului. Utilizatorii vor trimite comenzi catre server, acesta va rezolva cerere dupa care va intoarce rezultatul. Comunicarea se va realiza cu ajutorul unui server TCP, la baza fiind socket-urile. Pentru a elimina timpii de asteptare dintre procesele rulate la un moment dat si utilizatorii din coada de asteptare vom folosi un server concurent creat cu ajutorul firelor de executie. Pentru anumite operatii, servarul va trimite catre clienti mesaje OOB, clientul le va interpreta si va afisa rezultatul. Astfel utilizatorii vor afla cand un prieten a postat o stire, sau un admin a intrat pe server ori daca au primit cerere de chat. Limbajul folosit pentru scrierea aplicatiei va fi C sub Unix. Sistemul de operare unde am compilat aplicatia este Ubuntu14.

**3. Arhitectura aplicatiei**

Aplicatia are la baza comunicatia client-server. Clientul trimite comanda, servarul o proceseaza dupa care intoarce rezultatul. Clientul isi poate crea un cont de utilizator, datele vor fi trimise catre server iar acesta va aplica o interogare corespunzatoare asupra bazei de date pentru a complete un nou tuplu. Utilizatorul poate vedea postarile publice ale altor utilizatori fara sa fie logat. Odata cu logarea, utilizatorul va putea adauga un prieten la lista (apropriat, cunoastinta, etc.), posta o stire dar si comunica cu 2 sau mai multi utilizatori. Toate aceste date vor fi salvate in baza de date, asemenea blogurilor wordpress si vor putea fi accesate de catre utilizatori si administrator.

**3.1 Schema aplicatiei**

[](https://www.gliffy.com/go/html5/6715803?app=1b5094b0-6042-11e2-bcfd-0800200c9a66&accountId=3457095&noc=1417954856146)

Schema realizata cu ajutorul aplicatiei: [Gliffy](https://chrome.google.com/webstore/detail/gliffy-diagrams/bhmicilclplefnflapjmnngmkkkkpfad/related)

Pentru a vedea mai bine schema recomand instalarea aplicatiei pentru a putea deschide fisierul “modelare final.gliffy” sau vizualizarea imaginii “modelare final.jpeg” din fisierul incarcat.

[Link aplicatie](http://www.gliffy.com/)

**4.Detalii de implementare**

**4.1 Comunicarea cu 2 utilizatori – client**

**case** 8**:**

printf**(**"\n\n\x1b[34m--- Modul chat ---\x1b[0m\n\n"**);**

printf**(**" // comentariu Apasa CTRL + z pentru a vedea cine este online. (idee abandonata, comentariile sunt in surse)\n\n"**);**

char userNameChat**[**51**];**

char userNameChatCopie**[**50**];**

char userCorectChatConfirmare**[**50**];**

int lungimee**;**

\_\_confirmareUserChar**:**

**while(**1**)**

**{**

bzero **(**userNameChat**,** 100**);**

printf**(**"Introdu numele celui cu care vrei sa comunici\n"**);**

printf**(**"username: "**);**

fflush**(**stdout**);**

read**(**0**,**userNameChat**,**50**);**

lungimee**=**strlen**(**userNameChat**);**

bzero**(**userNameChatCopie**,sizeof(**userNameChatCopie**));**

**for(**i**=**0**;**i**<**lungimee**-**1**;**i**++)**

userNameChatCopie**[**i**]=**userNameChat**[**i**];** // eliminam '\n'

bzero**(&**chat2user\_request**,sizeof(**chat2user\_request**));**

strcpy**(**chat2user\_request**.**user1**,**login\_request**.**user**);**

strcpy**(**chat2user\_request**.**user2**,**userNameChatCopie**);**

**if(**strcmp**(**userNameChatCopie**,**login\_request**.**user**)==**0**)**

**{**

printf**(**"\n\n>>> User nepermis\n\n"**);**

**goto** \_\_confirmareUserChar**;**

**}**

**if** **(**write **(**sd**,** **&**oop\_switch**,** **sizeof(**int**))** **<=** 0**)**

**{**

perror **(**"Eroare nr: 23\n"**);**

**return** errno**;**

**}**

**if** **(**write **(**sd**,** **&**chat2user\_request**,** **sizeof(**chat2user\_request**))** **<=** 0**)**

**{**

perror **(**"Eroare nr: 24\n"**);**

**return** errno**;**

**}**

bzero**(**userCorectChatConfirmare**,sizeof(**char**)\***50**);**

**if** **(**read **(**sd**,** **&**userCorectChatConfirmare**,** **sizeof(**char**)\***50**)** **<=** 0**)**

**{**

perror **(**"Eroare nr: 25\n"**);**

**return** errno**;**

**}**

**if(**strcmp**(**userCorectChatConfirmare**,**"User gresit"**)==**0**)**

**{**

sch\_op**:**

printf**(**"\n>>> Userul introdus este gresit sau nu este online\nIncerci iar/schimbi optiunea? ir/sc\n"**);**

char schimba\_op**[**40**];**

bzero**(&**schimba\_op**,sizeof(**schimba\_op**));**

read**(**0**,&**schimba\_op**,sizeof(**char**)\***40**);**

**if((**strcmp**(**schimba\_op**,**"ir\n"**)==**0**)||(**strcmp**(**schimba\_op**,**"IR\n"**)==**0**))**

**goto** \_\_confirmareUserChar**;**

**else**

**if((**strcmp**(**schimba\_op**,**"sc\n"**)==**0**)||(**strcmp**(**schimba\_op**,**"SC\n"**)==**0**))**

**goto** meniu2\_label**;**

**else**

**{**printf**(**"\nOptiune gresita"**);**

**goto** sch\_op**;**

**}**

**}**

**else** **if** **(**strcmp**(**userCorectChatConfirmare**,**"User chat ocupat"**)==**0**)**

**{**

printf**(**"\n>>> Userul introdus este deja in chat.\n>>> Trebui sa primiti o invitatie de chat de la el pentru a putea conversa\n"**);**

fflush**(**stdout**);**

**goto** meniu2\_label**;**

**}**

**else**

**{**

printf**(**"\n\n>>> Userul introdus este valid\nI-am trimis cererea de chat\n"**);**

//creem 2 fire de exec

pthread\_create**(&**thr2**,NULL,(**void **\*)**chat\_scrie**,(**void **\*)**sd**);**

pthread\_create**(&**thr1**,NULL,(**void **\*)**chat\_citeste**,(**void **\*)**sd**);**

pthread\_join**(**thr2**,NULL);**

pthread\_join**(**thr1**,NULL);**

**break** **;**

**}**

**}** // end while **break;**

**4.2 Comunicarea cu 2 utilizatori – server**

**case** 8**:**

pthread\_mutex\_lock**(&**mlock8**);**

printf**(**"\n\n\x1b[33m[sock:%d]\x1b[0m=== Chat ===\n"**,**client**);**

char userNameChat**[**50**];**

char userCorectChatConfirmare**[**50**];**

bzero**(**userNameChat**,sizeof(**char**)\***50**);**

bzero**(**userCorectChatConfirmare**,sizeof(**char**)\***50**);**

bzero**(&**chat2user\_request**,sizeof(**chat2user\_request**));**

**if** **(**read **(**client**,** **&**chat2user\_request**,** **sizeof(**chat2user\_request**))** **<=** 0**)**

**{**

perror **(**"Eroare nr: 50\n"**);**

close **(**client**);**

**}**

//SELECT \* FROM `UTILIZATORI` WHERE `username`='stefanuca';

char **\*** interogareChatNumeU1**=**"SELECT \* FROM `UTILIZATORI`"**;**

**if** **(**mysql\_query**(**conn**,** interogareChatNumeU1**))**

fprintf**(**stderr**,** "\nEroare nr: 51\n%s\n"**,** mysql\_error**(**conn**));**

numeU1\_get**=**mysql\_use\_result**(**conn**);**

**while** **((**row **=** mysql\_fetch\_row**(**numeU1\_get**))** **!=** **NULL)**

**{**

**if(**strcmp**(**row**[**0**],**chat2user\_request**.**user1**)==**0**)**

strcpy**(**chat2user\_request**.**nume\_prenume\_u1**,**row**[**2**]);**

**if(**strcmp**(**row**[**0**],**chat2user\_request**.**user2**)==**0**)**

strcpy**(**chat2user\_request**.**nume\_prenume\_u2**,**row**[**2**]);**

**}**

/\* Verificam daca userul nu este deja in chat\*/

//SELECT \* FROM `chat\_groups`

**if** **(**mysql\_query**(**conn**,** "SELECT \* FROM `chat\_groups`"**))**

fprintf**(**stderr**,** "\nEroare nr: 52\n%s\n"**,** mysql\_error**(**conn**));**

chat\_verificare\_u\_group**=**mysql\_use\_result**(**conn**);**

**while** **((**row **=** mysql\_fetch\_row**(**chat\_verificare\_u\_group**))** **!=** **NULL)**

**{**

**if(**strcmp**(**row**[**0**],**chat2user\_request**.**user2**)==**0**)**

**{**

bzero**(**userCorectChatConfirmare**,sizeof(**char**)\***50**);**

strcpy**(**userCorectChatConfirmare**,**"User chat ocupat"**);**

printf**(**"\n\n\x1b[33m[sock:%d]\x1b[0m %s"**,**client**,**userCorectChatConfirmare**);**

**goto** trimite\_rasp\_cerere\_chat**;**

**break;**

**}**

**}**

char **\*** interogareCHat **=** "SELECT \* FROM `utilizatori\_online`"**;**

puts**(**interogareCHat**);**

**if** **(**mysql\_query**(**conn**,** interogareCHat**))**

fprintf**(**stderr**,** "\nEroare nr: 53\n%s\n"**,** mysql\_error**(**conn**));**

postChatVerificare**=**mysql\_use\_result**(**conn**);**

strcpy**(**userCorectChatConfirmare**,**"User gresit"**);**

int sd\_chat\_user\_confirm**;**

bool se\_poate\_insera**=**false**;**

char **\*** interogare\_Chat\_group1**=**"INSERT INTO `proiectRC`.`chat\_groups` VALUES ('"**;**

char txtas**[**16**];**

char **\*** interogare\_Chat\_group2**=**"INSERT INTO `proiectRC`.`chat\_groups` VALUES ('"**;**

**while** **((**row **=** mysql\_fetch\_row**(**postChatVerificare**))** **!=** **NULL)**

**{**

**if(**strcmp**(**chat2user\_request**.**user2**,**row**[**0**])==**0**)**

**{**

bzero**(**userCorectChatConfirmare**,sizeof(**char**)\***50**);**

strcpy**(**userCorectChatConfirmare**,**"User valid"**);**

/\* trimitem cererea de chat catre user 2; acesta va trebui sa confirme\*/

sd\_chat\_user\_confirm **=** atoi **(**row**[**1**]);**

**if(**send**(**sd\_chat\_user\_confirm**,**"c"**,**1**,**MSG\_OOB**)<**0**)**

**{**

puts**(**"Eroare nr: 54"**);**

**}**

/\*trimitem structura pt ca utilizatorul sa stie de la cine este cererea \*/

**if** **(**write **(**sd\_chat\_user\_confirm**,** **&**chat2user\_request**,** **sizeof(**chat2user\_request**))** **<=** 0**)**

**{**

perror **(**"Eroare nr: 55\n"**);**

**}**

//INSERT INTO `chat\_groups` VALUES ('stefanuca',5,NULL) // username,descriptor\_socket,null

interogare\_Chat\_group1**=**concatenare**(**interogare\_Chat\_group1**,**chat2user\_request**.**user1**);**

interogare\_Chat\_group1**=**concatenare**(**interogare\_Chat\_group1**,**"',"**);**

sprintf**(**txtas**,** "%d"**,** client**);** // descriptorul clientului il converim pt a face concatenarea

interogare\_Chat\_group1**=**concatenare**(**interogare\_Chat\_group1**,**txtas**);**

interogare\_Chat\_group1**=**concatenare**(**interogare\_Chat\_group1**,**",NULL)"**);**

interogare\_Chat\_group2**=**concatenare**(**interogare\_Chat\_group2**,**chat2user\_request**.**user2**);**

interogare\_Chat\_group2**=**concatenare**(**interogare\_Chat\_group2**,**"',"**);**

interogare\_Chat\_group2**=**concatenare**(**interogare\_Chat\_group2**,**row**[**1**]);**

interogare\_Chat\_group2**=**concatenare**(**interogare\_Chat\_group2**,**",NULL)"**);**

se\_poate\_insera**=**true**;**

**break;**

**}**

**}**

mysql\_free\_result**(**postChatVerificare**);**

**if(**se\_poate\_insera**)**

**{**

**if** **(**mysql\_query**(**conn**,** interogare\_Chat\_group1**))**

fprintf**(**stderr**,** "\nEroare nr: 56\n%s\n"**,** mysql\_error**(**conn**));**

**if** **(**mysql\_query**(**conn**,** interogare\_Chat\_group2**))**

fprintf**(**stderr**,** "\nEroare nr: 57\n%s\n"**,** mysql\_error**(**conn**));**

printf**(**"Interogari baza de date:\n%s\n%s"**,**interogare\_Chat\_group1**,**interogare\_Chat\_group2**);**

**}**

printf**(**"\n\n\x1b[33m[sock:%d]\x1b[0m %s\n"**,**client**,**userCorectChatConfirmare**);**

fflush**(**stdout**);**

trimite\_rasp\_cerere\_chat**:**

mysql\_free\_result**(**chat\_verificare\_u\_group**);**

//trimiterea raspunsului

**if** **(**write **(**client**,** **&**userCorectChatConfirmare**,** **sizeof(**char**)\***50**)** **<=** 0**)**

**{**

perror **(**"Eroare nr: 58\n"**);**

**return** **&**errno**;**

**}**

pthread\_mutex\_unlock**(&**mlock8**);** **break;**

**4.3 Comunicarea cu mai multi utilizatori: adauga nou user – server**

**case** 11**:**

printf**(**"\n\n\x1b[33m[sock:%d]\x1b[0m=== Adauga prieten pentru conversatie [CHAT] ===\n"**,**client**);**

char getUserName**[**50**];**

char raspuns\_getUserName**[**50**];**

**if** **(**read **(**client**,** **&**chat2user\_request**,** **sizeof(**chat2user\_request**))** **<=** 0**)**

**{**

perror **(**"Eroare nr: 68\n"**);**

**}**

bzero**(&**getUserName**,sizeof(**getUserName**));**

**if(**read**(**client**,&**getUserName**,sizeof(**getUserName**))<**0**)**

**{**

printf**(**"\nEroare nr: 69"**);**

**}**

//SELECT \* FROM `utilizatori\_online`

**if** **(**mysql\_query**(**conn**,** "SELECT \* FROM `utilizatori\_online`"**))**

fprintf**(**stderr**,** "\nEroare nr: 70 \n%s\n"**,** mysql\_error**(**conn**));**

lista\_u\_online2**=**mysql\_use\_result**(**conn**);**

strcpy**(**raspuns\_getUserName**,**"gresit"**);**

bool run\_sql\_comm**=**false**;**

char **\*** interog\_chat**=**"INSERT INTO `proiectRC`.`chat\_groups` VALUES ('"**;**

**while** **((**row **=** mysql\_fetch\_row**(**lista\_u\_online2**))** **!=** **NULL)**

**{**

**if** **(**strcmp**(**row**[**0**],**getUserName**)==**0**)**

**{**

run\_sql\_comm**=**true**;**

bzero**(&**raspuns\_getUserName**,sizeof(**raspuns\_getUserName**));**

strcpy**(**raspuns\_getUserName**,**"corect"**);**

//inseram in baza de date in chat group

interog\_chat**=**concatenare**(**interog\_chat**,**row**[**0**]);**

interog\_chat**=**concatenare**(**interog\_chat**,**"',"**);**

interog\_chat**=**concatenare**(**interog\_chat**,**row**[**1**]);**

interog\_chat**=**concatenare**(**interog\_chat**,**",NULL)"**);**

puts**(**interog\_chat**);**

//trimitem cerere de chat

int desc\_user**;**

desc\_user**=**atoi**(**row**[**1**]);**

**if(**send**(**desc\_user**,**"c"**,**1**,**MSG\_OOB**)<**0**)**

**{**

puts**(**"Eroare nr: 71"**);**

**}**

**if** **(**write **(**desc\_user**,** **&**chat2user\_request**,** **sizeof(**chat2user\_request**))** **<=** 0**)**

**{**

perror **(**"Eroare nr: 72"**);**

**}**

**break;**

**}**

**}**

mysql\_free\_result**(**lista\_u\_online2**);**

**if(**run\_sql\_comm**)**

**{**

**if** **(**mysql\_query**(**conn**,** interog\_chat**))**

fprintf**(**stderr**,** "\nEroare nr: 73\n %s\n"**,** mysql\_error**(**conn**));**

**}**

printf**(**"Raspunsul este %s"**,**raspuns\_getUserName**);**

fflush**(**stdout**);**

**if** **(**write **(**client**,** raspuns\_getUserName**,** **sizeof(**raspuns\_getUserName**))** **<=** 0**)**

**{**

perror **(**"Eroare nr: 74\n"**);**

**}**

**break;**

**4.4 Citirea mesajelor in chat – functie client**

void**\*** chat\_citeste**(**int sd**)** // citim si afisam mesajul, aceasta functie o folosim la comunicarea dintre 2 sau mai multi utilizatori

**{**

**while(**1**)**

**{**

**if** **(**read **(**sd**,** **&**msg\_chat\_request**,** **sizeof(**struct str\_msg\_chat**))** **<=** 0**)**

**{**

perror **(**"\n\nEroare nr:42\nA picat servarul\n\n"**);**

exit**(**0**);**

**}**

**if(**strcmp**(**msg\_chat\_request**.**msg**,**"\n"**)!=**0**)** // nu vrem sa afisam mesajele de tipul '\n' fiindaca utilizatorul poate apasa involuntar '\n'

printf **(**"\n%s"**,**msg\_chat\_request**.**msg**);**

fflush**(**stdout**);**

bzero **(&**msg\_chat\_request**,** **sizeof(**msg\_chat\_request**));**

**}**

**}**

**4.5 Scrierea mesajelor in chat – functie client**

void**\*** chat\_scrie**(**int sd**)**

**{**

**while(**1**)**

**{**

int asdasda**=**10**;**

afara\_meniu\_chat**:**

printf**(**"\x1b[0m\x1b[36m%s:\x1b[0m "**,**login\_request**.**user**);**

fflush**(**stdout**);**

bzero**(**msg**,sizeof(**char**)\***50**);**

**if(**read**(**0**,**msg**,**50**)<**0**){**printf**(**"\nEroare nr: 33"**);}**

**if(**strcmp**(**msg**,**"\n"**)!=**0**)** // nu vrem sa trimitem catre server '\n', un utilizator poate apasa involuntar "ENTER"

**{**

bzero**(&**msg\_copie**,sizeof(**char**)\***49**);**

int lung\_msg**,**i\_lung\_msg**;**

lung\_msg**=**strlen**(**msg**);**

**for(**i\_lung\_msg**=**0**;**i\_lung\_msg**<**lung\_msg**-**1**;**i\_lung\_msg**++)**

msg\_copie**[**i\_lung\_msg**]=**msg**[**i\_lung\_msg**];**

bzero**(&**msg\_chat\_request**,sizeof(**msg\_chat\_request**));**

strcpy**(**msg\_chat\_request**.**userr**,**login\_request**.**user**);**

strcpy**(**msg\_chat\_request**.**msg**,**msg\_copie**);**

bzero**(&**comandaMeniuChat**,sizeof(**char**)\***13**);**

**for(**i\_lung\_msg**=**0**;**i\_lung\_msg**<**13**;**i\_lung\_msg**++)**

comandaMeniuChat**[**i\_lung\_msg**]=**msg**[**i\_lung\_msg**];**

**if((**strcmp**(**msg\_chat\_request**.**msg**,**"/help"**)==**0**)||(**strcmp**(**msg\_chat\_request**.**msg**,**"/h"**)==**0**)){**printf**(**"Meniu chat: \n-adaugaPrieten\n-quit | exit\n"**);** **goto** afara\_meniu\_chat**;}**

**if(**strcmp**(**comandaMeniuChat**,**"adaugaPrieten"**)==**0**)**

**{**

int chatConv **=**11**;**

pthread\_cancel**(**thr1**);** //omoram firul pentru a nu citi mesajul de la server prin functia chat\_citeste, si a trata gresit raspunsul

char raspuns\_getUserName**[**50**];**

bzero**(&**getUserName**,sizeof(**getUserName**));**

**for(**i\_lung\_msg**=**0**;**i\_lung\_msg**<**lung\_msg**-**15**;**i\_lung\_msg**++)**

**if(**getUserName**[**i\_lung\_msg**]!=**'\n'**)**

getUserName**[**i\_lung\_msg**]=**msg**[**i\_lung\_msg**+**14**];**

**if(**strcmp**(**getUserName**,**login\_request**.**user**)!=**0**)**

**{**

**if(**write**(**sd**,&**chatConv**,sizeof(**int**))<**0**){**printf**(**"\nEroare nr: 34"**);}**

strcpy**(**chat2user\_request**.**user1**,**login\_request**.**user**);**

**if** **(**write **(**sd**,** **&**chat2user\_request**,** **sizeof(**chat2user\_request**))** **<=** 0**){**perror **(**"Eroare nr: 35\n"**);** **}**

**if(**write**(**sd**,&**getUserName**,sizeof(**getUserName**))<**0**){**printf**(**"\nEroare nr: 36"**);}**

bzero**(&**raspuns\_getUserName**,sizeof(**raspuns\_getUserName**));**

**if** **(**read **(**sd**,** raspuns\_getUserName**,** **sizeof(**raspuns\_getUserName**))** **<=** 0**){**puts **(**"Eroare nr: 37\n"**);}**

pthread\_create**(&**thr1**,NULL,(**void **\*)**chat\_citeste**,(**void **\*)(**intptr\_t**)**sd**);** //dupa ce am citit mesajul creem din nou threadul omorat mai sus cu 12 linii de cod

**if(**strcmp**(**"gresit"**,**raspuns\_getUserName**)==**0**)**

printf**(**"\nUserul pe care ai incercat sa il adaugi la chat este: %s ori nu este online\n"**,**raspuns\_getUserName**);**

fflush**(**stdout**);**

**}**

**else**

printf**(**"\nNu te poti adauga pe tine la chat\n"**)** **;**

**}**

**else**

**if((**strcmp**(**msg\_chat\_request**.**msg**,**"quit"**)==**0**)||(**strcmp**(**msg\_chat\_request**.**msg**,**"exit"**)==**0**))**

**{**

printf**(**"\n\n\x1b[32mAi iesit din chat\x1b[0m\n"**);**

**if(**write**(**sd**,&**asdasda**,sizeof(**int**))<**0**){**printf**(**"\nEroare nr: 38"**);}**

**if(**write**(**sd**,&**msg\_chat\_request**,sizeof(**struct str\_msg\_chat**))<**0**){**printf**(**"\nEroare nr: 39"**);}**

pthread\_cancel**(**thr2**);**

pthread\_cancel**(**thr1**);**

sleep**(**1**);** // functia pthread\_cancel() este mai lenta decat structura while(1)

//ar aparea un mesaj in plus pe chat daca nu am astepta o sec.

// un mesaj de citire fara "actiune"

**}**

**else**

**{**

**if(**write**(**sd**,&**asdasda**,sizeof(**int**))<**0**){**printf**(**"\nEroare nr: 40"**);}**

int lung**;**

lung**=**strlen**(**msg\_chat\_request**.**msg**);**

**if(**write**(**sd**,&**msg\_chat\_request**,sizeof(**struct str\_msg\_chat**))<**0**){**printf**(**"\nEroare nr: 41"**);}**

**}**

**}**

**}**// end while 1

**}**

**5. Cum folosim aplicatia ?**

Compilam aplicatia cu ajutorul scriptului **serv.sh**, serverul ar trebui sa porneasca automat (daca folosim scriptul). Clientul il pornim in terminal cu ajutorul comenzii **./c 127.0.0.1 2021** (daca serverul este pe localhost) altfel **./c hostserver 2021**

Folosirea meniului(1) este banala , aici ne putem crea un nou cont, vizualiza postarile publice sau loga pe server. Odata cu logarea vom avea acces la meniu(2), care este de asemene foarte usor de folosit. Daca suntem in chat si comunicam cu un prieten pentru a accesa meniul trebuie sa introducem la linia de comanda una dintre optiunile “/h” sau “/help”. Dupa cum sugereaza si acel meniu, pentru a adauga un nou prieten in chat (pentru a face comunicare intre 3 utilizatori) trebuie sa introducem comanda “**adaugaPrieten $username**” unde **username** este nick-ul celui pe care vrem sa il adaugam in chat. Pentru a iesi din chat scriem comanda **exit** sau **quit**.

Recomand vizualizarea pozelor sugestive din folder-ul **poze**.

**6. Concluzii**

Am incercat sa fac serverul concurent cu ajutorul functiei fork(), iar problema a fost la partea de comunicare intre clienti. Clientul cel mai nou putea sa trimita mesaje celorlalti in schimb nu putea sa primeasca fiindca acestea nu ii vedeau descriptorul. Am incercat sa rezolv problema cu un pipe, sa fac un soi de comunicare copil-parinte-server-parinte-copil, aceasta idee a esuat fiindca acel pipe crea o structura blocanta pentru urmatorii clienti care s-ar fi conectat(din cauza acelui pipe serverul ar fi deveni interativ, cu toate ca faceam acel fork() la inceput). Avantajul serverelor concurente de tip fork() este ca pot citi mesale OOB pe care le pot interpreta printr-un switch, astfel pe arhitectura gandita la acest proiect puteam sa apelez o functie care trimitea catre client utilizatorii online.

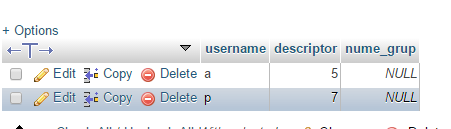
Dezavantajul firelor de executie este ca nu pot citi mesaje OOB. Din aceasta cauza nu mai pot folosi functia pentru a vedea utilizatorii online.

Alta problema este faptul ca mai multi clienti se “bat” pe bucati de cod si modifica variabile de care unii clienti inca au nevoie. Aceasta problema am incercat sa o rezolv prin multiplexare: pthread\_mutex\_lock, pthread\_mutex\_unlock insa obtineam erori de la baza de date. Pentru a scapa si de acestea m-am gandit sa fac fork() pe acele bucati de cod “critice”, iar cu fork() am rezolvat problema.

O alta incercare esuata a fost sa fac un server care sa ofere mai multe servicii la diferite porturi, pe care sa le selectez cu select(). Ideea este buna insa as fi avut problema destul de mari. Un exemplu este ca: nu as fi putut sa fac **o incercare de login de 2 ori** de pe acelasi client fiindca in server as fi facut doar o singura data accept(), iar descriptorul returnat ar fi fost ocupat cu prima incercare de login si a doua incercare ar fi fost blocanta.

Codul programului poate fi optimizat, servarul se poate implementa sub paradigma Remote Procedure Call (RPC) –astfel am rezolva problema dependentei de platform, programul ar fi portabil .

In implementarea propusa, daca in sectiunea **chat** sunt 20 de utilizatori atunci se va realiza comunicarea intre toti 20 si nu se pot forma grupuri de cate 5 (de exemplu), pentru a rezolva aceasta problema propun urmatoarea solutie: In baza de date in tabela chat\_groups in coloana **nume\_grup** in loc de **NULL** se va completa cu numele grupului la care comunica utilizatorul.



**7.Bibliografie**

<http://profs.info.uaic.ro/~busaco/teach/courses/net/net-biblio.html#web>

<http://www.cs.cmu.edu/afs/cs/academic/class/15492-f07/www/pthreads.html>

<http://profs.info.uaic.ro/~eonica/rc/>

<http://profs.info.uaic.ro/~adria/teach/courses/net/cursullaboratorul.php>

<http://www.yolinux.com/TUTORIALS/MySQL-Programming-C-API.html>

Aplicatiile de la care am pornit in creare proiectului sunt urmatoarele:

<http://profs.info.uaic.ro/~eonica/rc/client-tcp.c>

<http://profs.info.uaic.ro/~eonica/rc/server-tcp.c>

Ideea de a utiliza 2 fire de executie pentru comunicarea cu doi sau mai multi utilizatori am preluat-o de aici:

<http://www.techonicals.com/search?updated-max=2012-09-03T03:21:00-07:00&max-results=7&reverse-paginate=true>