Tema 2 – Reţele de Calculatoare

Timon Roxana Mihaela X3

December 9, 2021

1 Introducere

Acest document are rolul de a prezenta documentaţia corespunzătoare proiectului MyFileTransferProtocol . Proiectul are la bază o aplicaţie client - server şi este realizat folosind limbajul de programare C. De asemenea, proiectul oferă funcţionalitate transferului de fişiere pe maşini aflate la distanţă, clientul putând descărca fişiere aflate pe server sau încarca propriile fişiere pe server. Aplicaţia este prevăzută cu un mecanism de autentificare, astfel încat clienţii care doresc sa execute comenzile menţionate mai sus, vor trebui să-şi creeze un cont de utilizator.

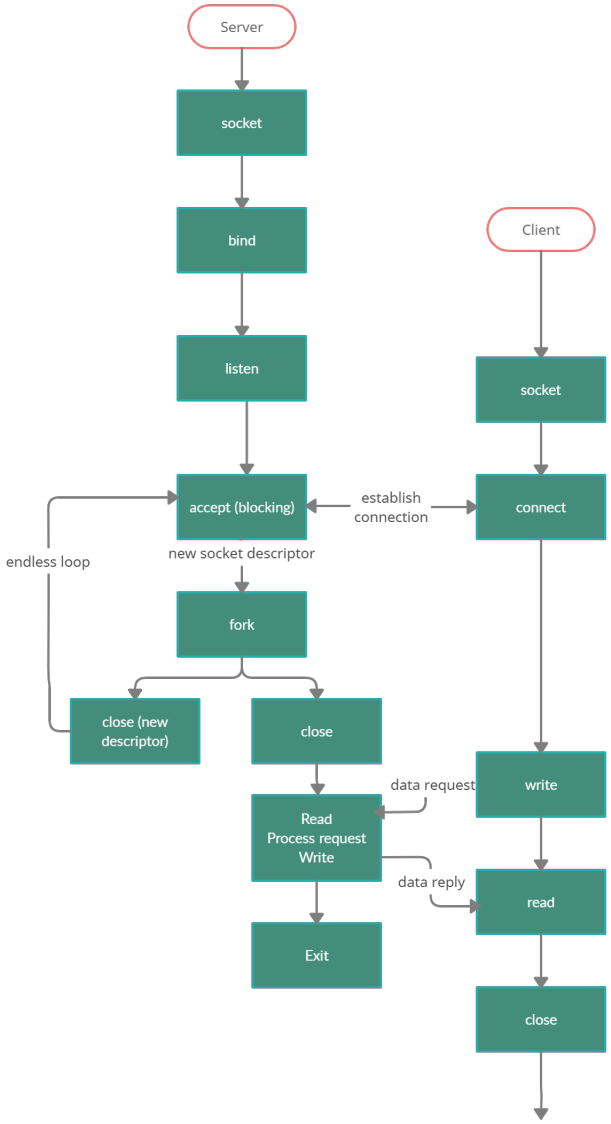
2 Tehnologii utilizate

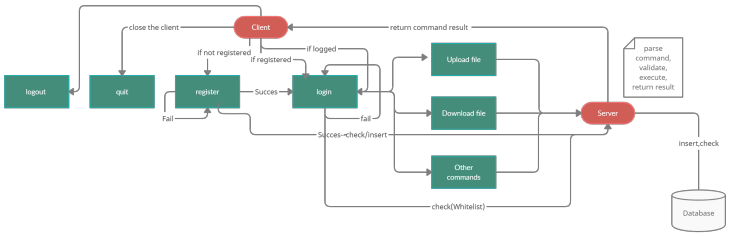
Pentru realizarea acestui proiect am folosit mecanismul de comunicare TCP (Transmission Control Protocol) concurent. Acest protocol de comunicare orientat pe conexiunea permite conectarea mai multor clienţi la server. TCP este un protocol care face parte împreuna cu UDP din nivelul 4 al modelului OSI, mai exact nivelul Transport. O parte din caracteristicile corespunzatoare acestui protocol sunt : se asigură de integritatea datelor transmise, în cazul în care unele pachete cu date se pierd "pe drum" acesta asigura retransmiterea datelor, ordonează pachetele ajunse la destinaţie, stabileşte o conexiune între client şi server. Serverul TCP va crea câte un proces copil prin intermediul instrucţiunii fork(). Serverul va avea rolul de a înregistra date de la client prin intermediul socket-ului corespunzător, de a le procesa ¸ actualiza, şi de a transmite informatii cerute înapoi la cliet. Clientul la rândul sau păstreaza o conexiune cu serverul, recepţioneaza mesaje de la server, le procesează ¸ preia comenzi de la tastatură. Comunicarea dintre server şi client va putea fi observată mai bine prin intermediul digramei de la punctul 3. Pentru stocarea datelor cu privire la utilizatorii aplicaţiei ne vom folosi de o baza de date SQLite, în care vom memora prin intermediul tabelei Users IDuser, username, ip, password. Mecanismul de transmitere securizata a parolei va fi realizat prin metoda de criptare simetrică DES (Data Encryption Standard).Cu ajutorul bazei de date, vom sti exact care utilizatori vor avea accesul de a executa comenzi catre server, astfel încât daca Ip-ul corespunzător clientului apare in baza de date (Whitelisting) acesta

va avea drepturile prevazute de aplicaţie, iar în caz contrar accesul acestuia va fi strict interzis.Scopul includerii în lista albă(WhiteList) este de a proteja computerele și rețelele de aplicații potențial dăunătoare.

3 Arhitectura aplicaţiei

Arhitectura aplicaţiei am realizat-o dupa modelul TCP concurent. Concepte implicate: server, client, utilizator, baze de date.





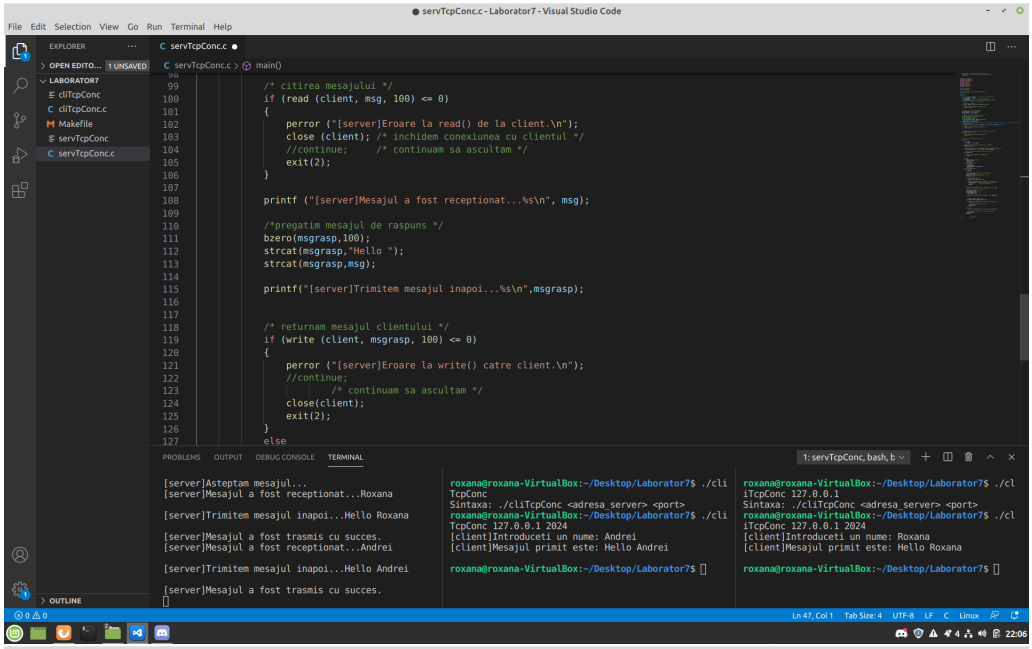
<<Specific Class 1>>
Users

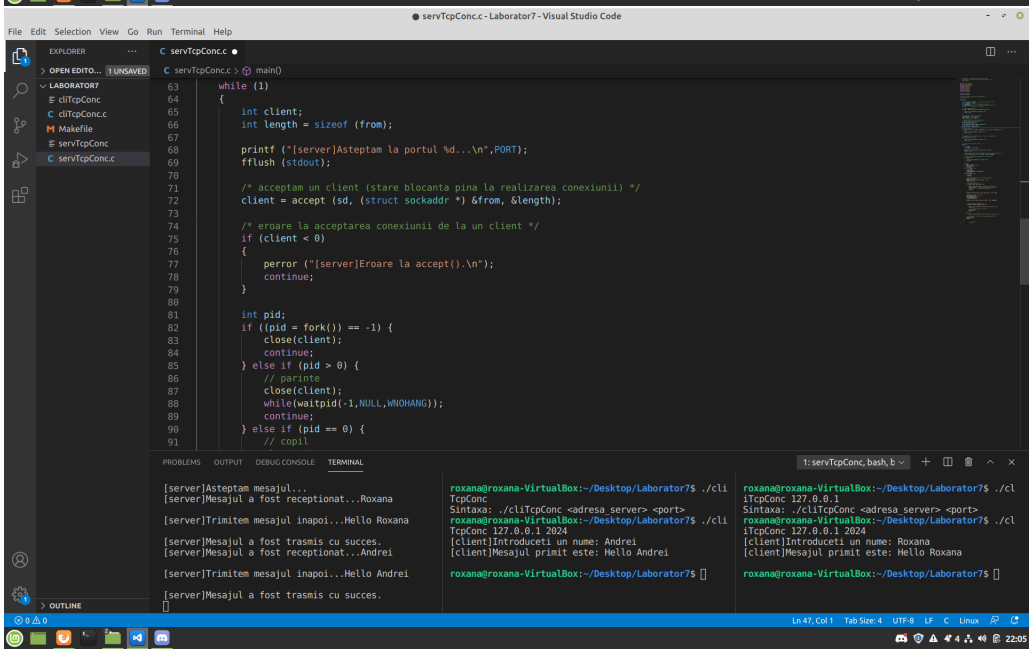
- id: number
+ username: varchar
- password: varchar
- ip: number
- logged : tinyint

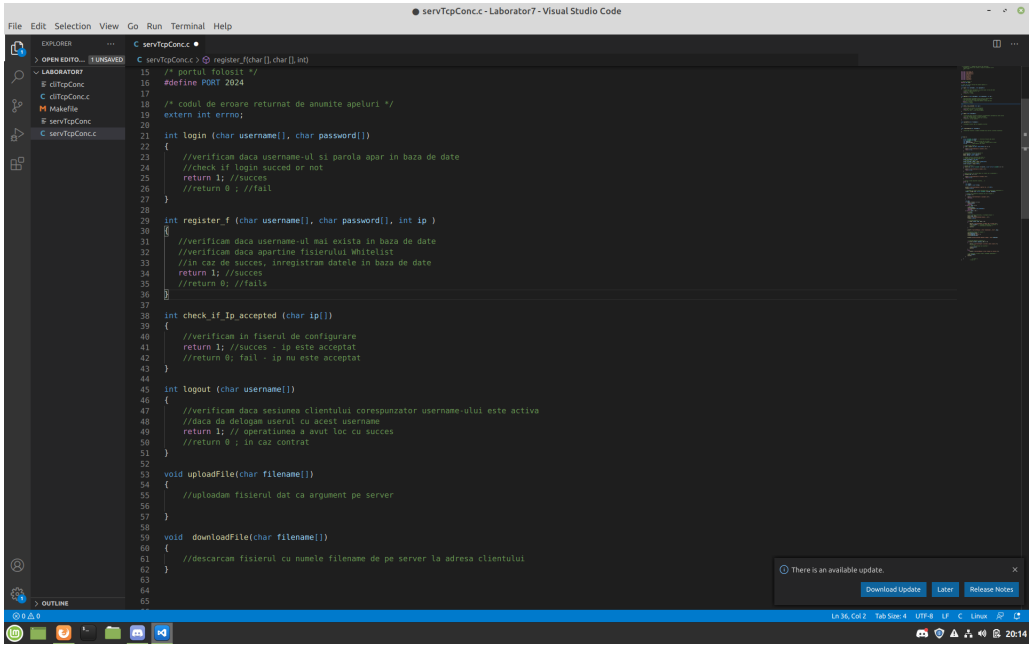
+ AddUser(args): Type
+ GetUser(id): Type
+ CheckIfExistUserWithGivenId(args): Type
+ CheckIfUserIsLogged(id): Type

4 Detalii implementare

În ceea ce priveste implementarea, la nivelul serverului vom folosi mai multe funcţii cu facilităti diferite. In functia main stabilim toate caracteristicile structurii ce indică la adresa serverului,respectiv a clientului, verficăm argumentele primite la linia de comanda si tratam erorile, creăm un socket, si il ataşăm la adresa serverului.Totodată realizam o bucla infinită in care acceptam pe rând cererile de clienţi. La conexiunea cu clientul prin intermediul primitivei accept vom reţine informaţiile despre client şi vom obţine file-descriptorul specific socketului cu care vom comunica cu clientul, vom prelua comanda (mesajul) pe care acesta ni-l transmite şi o vom prelucra. Funţia de înregistrare are rolul de a înregistra un utilizator, iar username-ul, ip si parola criptata vor fi memorate la nivelul bazei de date. Functia de login are rolul de a loga un utilizator,doar daca acesta este deja inregistrat. Odată logat utilizatorul poate executa urmatoarele comenzi : download file, upload file , etc. (comenzi de prelucrare cu fisiere şi directoare). Functiile de ieşire din program si logout vor putea fi accesate oricând şi vor deloga orice client care a cerut acest lucru si are sesiunea activă. Scenarii de utilizare: În cazul în care un utilizator al carui IP nu este inregistrat in Whitelist, încercarea de a-şi face un cont nu va avea succes.


servTcpConc.c - Laborator7 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
OPEN EDITO... 1 UNSAVED
LABORATOR7
cliTcpConc
cliTcpConc.c
Makefile
servTcpConc
servTcpConc.c
servTcpConc.c ●
servTcpConc.c > main()
/* 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 */
exit(2);
}
printf ("[server]Mesajul a fost receptionat...%s\n", msg);
/*pregatim mesajul de raspuns */
bzero(msgrasp,100);
strcat(msgrasp,"Hello ");
strcat(msgrasp,msg);
printf("[server]Trimitem mesajul inapoi...%s\n",msgrasp);
/* returnam mesajul clientului */
if (write (client, msgrasp, 100) <= 0)
{
perror ("[server]Eroare la write() catre client.\n");
//continue;
/* continuam sa ascultam */
close(client);
exit(2);
}
else
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
1: servTcpConc, bash, b
[server]Asteptam mesajul...
[server]Mesajul a fost receptionat...Roxana
[server]Trimitem mesajul inapoi...Hello Roxana
[server]Mesajul a fost trasmis cu succes.
[server]Mesajul a fost receptionat...Andrei
[server]Trimitem mesajul inapoi...Hello Andrei
[server]Mesajul a fost trasmis cu succes.
roxana@roxana-VirtualBox:~/Desktop/Laborator7$ ./cli
TcpConc
Sintaxa: ./cliTcpConc <adresa_server> <port>
roxana@roxana-VirtualBox:~/Desktop/Laborator7$ ./cli
TcpConc 127.0.0.1 2024
[client]Introduceti un nume: Andrei
[client]Mesajul primit este: Hello Andrei
roxana@roxana-VirtualBox:~/Desktop/Laborator7$
roxana@roxana-VirtualBox:~/Desktop/Laborator7$ ./cl
iTcpConc 127.0.0.1
Sintaxa: ./cliTcpConc <adresa_server> <port>
roxana@roxana-VirtualBox:~/Desktop/Laborator7$ ./cl
iTcpConc 127.0.0.1 2024
[client]Introduceti un nume: Roxana
[client]Mesajul primit este: Hello Roxana
roxana@roxana-VirtualBox:~/Desktop/Laborator7$
OUTLINE
Ln 47, Col 1 Tab Size: 4 UTF-8 LF C Linux
22:06


servTcpConc.c - Laborator7 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
OPEN EDITO... 1 UNSAVED
LABORATOR7
cliTcpConc
cliTcpConc.c
Makefile
servTcpConc
servTcpConc.c
servTcpConc.c ●
servTcpConc.c > main()
while (1)
{
int client;
int length = sizeof (from);
printf ("[server]Asteptam la portul %d...\n",PORT);
fflush (stdout);
/* acceptam un client (stare blocanta pina la realizarea conexiunii) */
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) {
// copil
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
1: servTcpConc, bash, b
[server]Asteptam mesajul...
[server]Mesajul a fost receptionat...Roxana
[server]Trimitem mesajul inapoi...Hello Roxana
[server]Mesajul a fost trasmis cu succes.
[server]Mesajul a fost receptionat...Andrei
[server]Trimitem mesajul inapoi...Hello Andrei
[server]Mesajul a fost trasmis cu succes.
roxana@roxana-VirtualBox:~/Desktop/Laborator7$ ./cli
TcpConc
Sintaxa: ./cliTcpConc <adresa_server> <port>
roxana@roxana-VirtualBox:~/Desktop/Laborator7$ ./cli
TcpConc 127.0.0.1 2024
[client]Introduceti un nume: Andrei
[client]Mesajul primit este: Hello Andrei
roxana@roxana-VirtualBox:~/Desktop/Laborator7$
roxana@roxana-VirtualBox:~/Desktop/Laborator7$ ./cl
iTcpConc 127.0.0.1
Sintaxa: ./cliTcpConc <adresa_server> <port>
roxana@roxana-VirtualBox:~/Desktop/Laborator7$ ./cl
iTcpConc 127.0.0.1 2024
[client]Introduceti un nume: Roxana
[client]Mesajul primit este: Hello Roxana
roxana@roxana-VirtualBox:~/Desktop/Laborator7$
OUTLINE
Ln 47, Col 1 Tab Size: 4 UTF-8 LF C Linux
22:05



5 Concluzii

Aplicatia MyFileTransferProtocol ar putea fi îmbunătăţită prin adaugarea de noi comenzi, precum delete, ls, cd, pwd. La nivelul tehnologiei utlizate, consider ca un server UDP ar aduce beneficii in ceea ce priveste viteza de transmitere a datelor, însa in acest mod nu s-ar mai pastra integritatea datelor. Totodata, folosind UDP nu ar trebui sa mai aşteptăm restabilirea conexiunii, în cazul in care una dintre conexiuni pică din cauza unei probleme de mediu. În ceea ce priveste aspectul aplicaţiei, sunt de pă rere ca o interfata grafica (GUI) ar fi mai atractiva.Totodată, folosirea unei metode mai avansate de criptare, ar asigura o mai buna securitate a aplicaţiei.

6 Bibliografie

Site-uri utilizate pentru realizarea proiectului:

https://profs.info.uaic.ro/ computernetworks/

https://www.youtube.com/watch?v=cNdlrbZSkyQt=845s

https://ro.wikipedia.org/wiki/TransmissionControlProtocol

https://profs.info.uaic.ro/ gcalancea/Laboratorul$_7$.*pdf*