Tema 2-MyFileTransferProtocol (B)

Iordan Cosmina 2B1

1 Introducere

My file transfer protocol este o aplicaţie de tip client/server ce permite transferul de fişiere între clienţi şi server.

Serverul acestei aplicaţii pune la dispoziţie clienţilor diferite comenzi cum ar fi întregistrarea,autentificarea şi diverse operaţii cu directoare şi fişiere(ex: deschiderea unui director, afişarea conţinutului unui director, transferul unui fişier). De asemenea acest server dispune de un mecanism de autorizare(ex: clienţii care sunt în "blacklist" nu se vor putea autentifica) şi un mecanism de ascundere a parolei la autentificare.

Un client nu v-a putea efectua nicio operaţie dacă acesta este restricţionat(se afla în "blacklist"),dacă nu este înregistrat sau dacă nu s-a autentificat. După autentificare clientul se va putea folosii de toate operaţiile puse la dispoziţie de server,iar la sfarşit acesta îşi va putea încheia sesiunea printr-o comandă de deconectare.

Această aplicaţie este destinată utilizatorilor ce vor să transmită fişiere.

2 Tehnologiile utilizate

Pentru această aplicație voi folosi un protocol de tip TCP,deoarece este un protocol sigur orientat pe conexiune care permite ca un flux de octeți trimiși de pe o mașină să ajungă fără erori pe orice altă mașină din inter-rețea.

Modul de operare al protocolului TCP implică existența a trei faze. În prima fază, conexiunea trebuie stabilită , urmând un proces de confirmare pe baza mai multor pași. Imediat ce conexiunea a fost realizată, urmează transferul datelor. Odată ce transferul datelor s-a încheiat, conexiunea trebuie terminată în ideea de a închide calea virtuală și de a elibera resursele hardware/software implicate în proces.

O aplicație de tip FTP folosește 2 conexiuni TCP pentru a transfera date între server și clienți.

- Conexiunea control,inițiată prin portul 21 ce permite trimiterea informațiilor de control (ex: parolă,username).
- Conexiunea de date,inițiată prin portul 20 ce permite transferul de date.

Aspecte pozitive privind utilizarea TCP:

- Este orientat conexiune.
- Oferă siguranță și asigură transmiterea în ordine a datelor.
- Oferă mecanisme de control al fluxului și control al congestiei.

– Este o arhitectură scalabilă, client-server. Acest lucru permite adăugarea rețelelor fără a perturba serviciile actuale.

Serverul TCP va crea câte un proces copil,folosind fork, pentru fiecare client pentru a putea servi mai mulți clienți simultan. Acesta va avea rolul de a prelua datele de la client,de a le procesa și,ulterior, de a trimite rezultatul către client.

Am ales acest protocol pentru ca este un serviciu de încredere care garantează livrarea unui flux de date trimis de la o gazdă la alta fără duplicarea sau pierderea de date.

3 Arhitectura aplicatiei

Pentru acest program se va instala o librărie ce are suport pentru protocolul FTP,diagramele de mai jos arată cum v-a funcționa codul,clientul va trimite comenzile către server,iar serverul va compila funcțiile în funcție de comenzile primite de la client,în diagrama server client este evidențiat acest aspect.

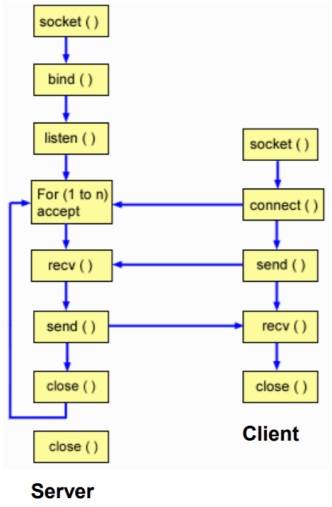

Fig. 1: Modelul TCP client/server

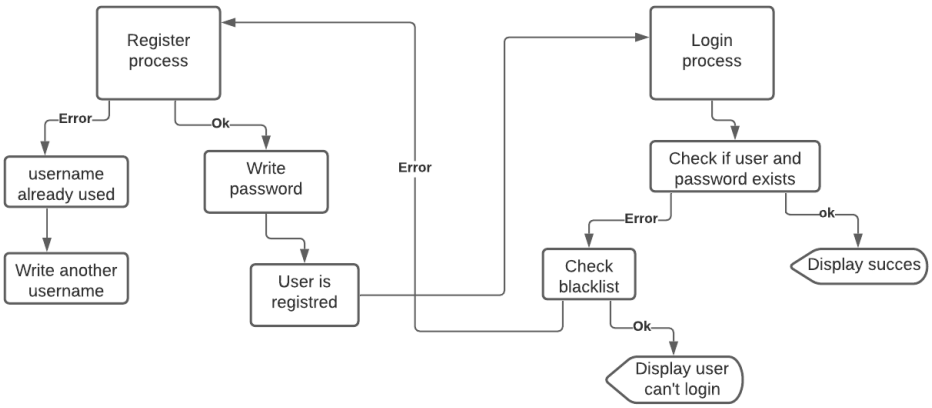

Fig. 2: Diagrama login/register

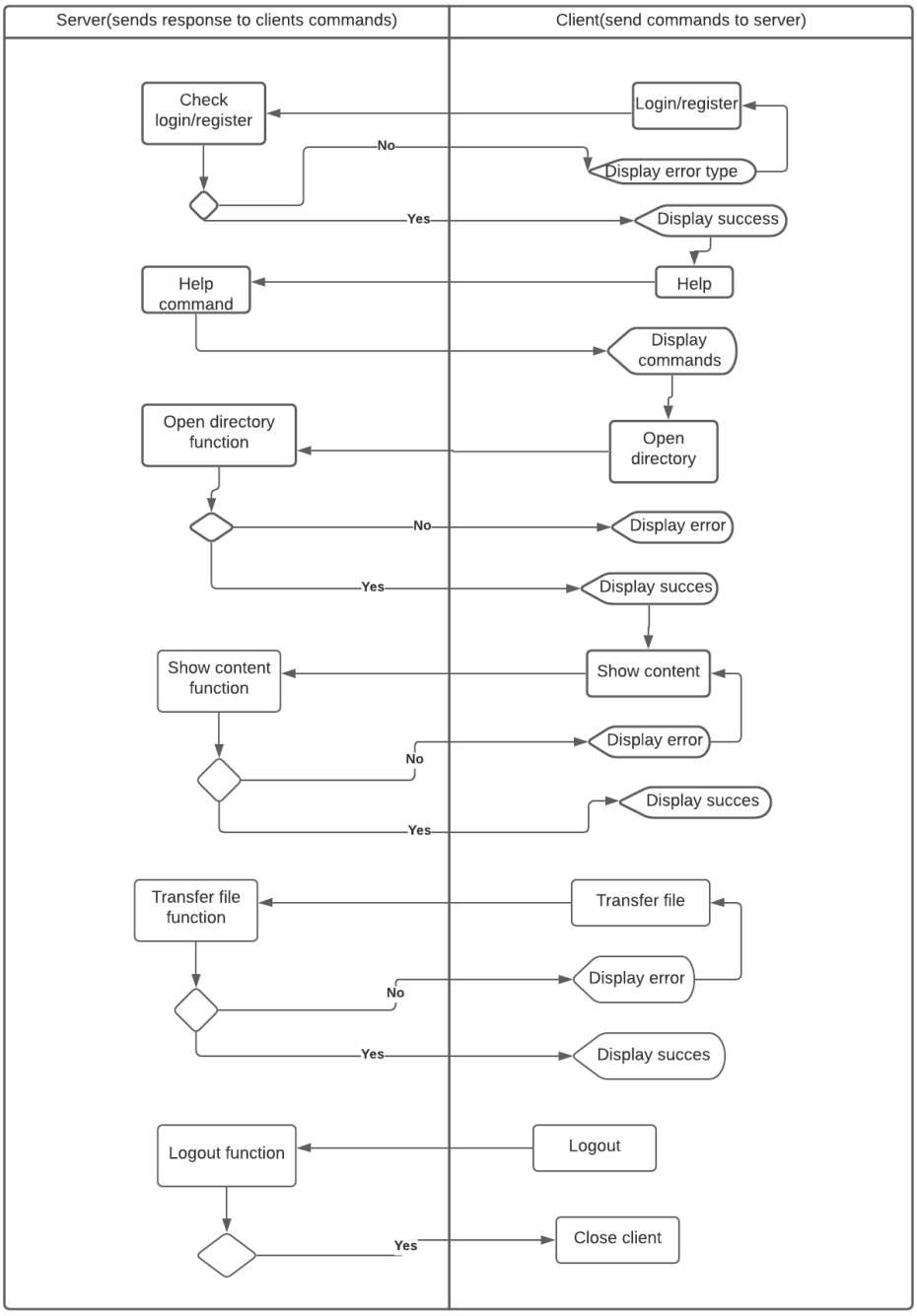

Fig. 3: Diagrama server/client

4 Detalii de implementare

Pentru această aplicație voi folosi un protocol TCP concurent,astfel serverul va putea primi și executa comenzile mai multor clienți în același timp,rolul său fiind de a se fixa (bind) pe un port specific (port folosit de client pentru a localiza serverul) și de a asculta pentru noi conexiuni.

Acest server va primi de la client comenzi precum: register, login, help, logout, open directory, show directory content, transfer file.

Comanda ***register*** va primi ca parametrii un username și o parolă(aceasta este "codată" în momentul scrierii,doar clientul va știi care este) și le va stoca într-un fișier cu conturile clienților,în același timp va verifica dacă există un alt client cu același username,dacă există clientul ce dorește înregistrarea va trebui să introducă un nou username până când se va găsi un username unic(nu mai există alt client cu acest username).

Comanda ***login*** primește ca parametrii un user și o parolă pentru acestea verifică dacă există în fișierul cu conturi sau dacă sunt în fișierul cu conturi restricționate(blacklist). Dacă perechea *username:parolă* se află în fișierul cu conturi atunci clientul este logat și folosi celelalte comenzi ale serverului,dacă această pereche se află în fișierul cu conturi resticționate atunci clientul nu se va putea loga,conexiunea lui va fi întreruptă. De asemenea,la fel ca în cazul comenzii *register* parola va fi "codată" astfel încât doar clientul va știi care este parola. Pentru implementarea acestei funcții voi folosi librăria *conio.h*(este diferită față de librăria *conio.h* disponibilă pentru Windows. Din această librărie mă folosesc de funcția *getch()* pentru a obține codul ASCII al caracterului introdus de la tastatură fără că acesta să apară pe consola.

Comanda ***help*** va afișa lista de comenzi disponibile pentru clienți.

Comanda ***open directory*** va deschide directorul indicat de client prin numele acestuia sau prin calea acestuia. Dacă primește de la client numele directorului această comandă va căuta recursiv prin toate directoarele serverului și va returna succes când va găsi directorul.

Comanda ***show directory content*** va afișa pe ecranul clientului fișierele disponibile din directorul curent,sau în directorul a cărui cale acesta o trimite către server. Această funcție se comportă precum comanda "*ls*" introdusă într-un terminal, listează toate directoarele și fișierele ce se află în directorul specificat.

Comanda ***transfer file*** va permite serverului să trimită către client fișiere. Pentru transferul de date se folosește portul 20.Transferul se realizează utilizând modul de transfer activ (binar sau text).

Comanda ***logout*** va deconecta clientul de la server și va închide procesul acestuia.

Un client nu va putea transmite către server comenzile pentru lucrul cu directoare dacă este restricționat sau dacă nu este autentificat. Deci comenzile ***register*** și ***login*** sunt disponibile clienților imediat, iar restul comezilor vor fi disponibile îndată ce aceștia se conectează la server.

5 Concluzii

Această aplicație este destinată persoanelor ce doresc să transmită fișiere. Pentru îmbunătățire se mai pot adăuga diferite comenzi pentru operații cu directoare cum ar fi: ștergerea unui fișier/director, crearea unui fișier/director, afișarea mărimii unui director/fișier, redenumire, afișarea atributelor unui director/fișier, copierea conținutului unui fișier.

6 Bibliografie

Site-uri utilizate:

- https: //profs.info.uaic.ro/ computernetworks/cursullaboratorul.php
- https://ro.wikipedia.org/wiki/Transmission_Control_Protocol
- https://github.com/zoelabbb/conio.h
- https://en.wikipedia.org/wiki/File_Transfer_Protocol#Data_transfer_modes
- https://www.competentedigitale.ro/internet/internet_ftp.php
- https://en.wikipedia.org/wiki/List_of_FTP_commands
- https://tools.ietf.org/html/rfc959
- https://mbpfaus.net/ pfau/ftplib/