

Tema 2 - MyFileTransferProtocol (B)

Tabalae Ioan - Sebastian, Anul 2, Grupa A4

Universitatea Alexandru Ioan-Cuza

January 11, 2022

Abstract

Crearea unei aplicatii client-server TCP ce permite transferul de fisiere intre clienti si server, precum si alte actiuni cu fisierele/directoarele dintr-un sistem local de fisiere. (creare, redenumire, ştergere, mutare etc).

Keywords— TCP, client-server, MyFTP

1 Introducere

Aplicatia permite utilizatorului sa realizeze transfer de fisiere intre client si server (functii de tipul send, receive), sau sa aplice anumite comenzi fisierele/directoarelor locale (creare, redenumire, stergere, mutare, afisarea continutului directorului curent/alt director). Programul poate fi pornit din linia de comanda, iar utilizatorul va trebui intai sa se logeze pentru a avea acces la celelalte functionalitati. Acesta va trimite comenzi serverului, care le va verifica sintactic, apoi le va interpreta, executa si va trimite un raspuns inapoi clientului.

2 Tehnologii utilizate

Aceasta aplicatie se bazeaza pe modelul client-server. Pentru a se conecta la server, clientul trebuie mai intai sa se logheze cu un username si o parola care sunt stocate intr-o baza de date SQLite pe server (ca fiind conturile existente pe acest server). Parola este transmisa in mod securizat, prin criptare cu functia de hashing SHA1. Procesele vor comunica intre ele prin socket-uri, iar protocolul folosit este TCP deoarece asigura o transmitere precisa si sigura a datelor.

TCP este optimizat, mai degraba, pentru livrarea exacta decat livrarea la timp a datelor, si prin urmare, TCP inregistrează uneori, întârzieri relativ mari de timp (de ordinul secundelor), in timpul de asteptare pentru unele mesaje ce sosesc in alta ordine sau pentru retransmisia de mesaje pierdute. Dar, in cazul nostru, transmiterea de fisiere necesita o mai mare atentie, deoarece avem nevoie de trimiterea si primirea datelor in ordine avand siguranta ca acestea au fost trimise/primite in modul in care ne-am astepta, iar astfel TCP asigura toate aceste functionalitati singurul cost fiind viteza.

Se va crea cate un proces-copil din procesul principal pentru fiecare client nou conectat, prin fork(). Clientul trimite cate o comanda serverului (/login, /list files), iar serverul verifica corectitudinea, o proceseaza, si trimite inapoi clientului un mesaj privind statusul comenzii date. Clientii vor fi identificati prin IP si port.

Limbajul folosit este C.

3 Arhitectura aplicatiei

Baza arhitecturii aplicatiei consta in trimiterea de comenzi de la client la server, efectuarea functiilor aferente de catre server si transmiterea de mesaje de la server la client cu rezultatul comenzii date.

Utilizatorul va putea realiza actiuni de transfer de fisiere numai după autentificare (login). Pentru aceasta, va trebui sa introduca comanda /login, iar dupa sa scrie username-ul si parola. Se va verifica username-ul si parola criptata in baza de date, iar daca acesta are drept de acces (exista in whitelist si nu exista in blacklist), va fi logat cu succes si va putea accesa celelalte comenzi. Cand utilizatorul va introduce parola, se va calcula functia hash **SHA1(password, strlen(password), md)** si se va compara cu valoarea stocata in baza de date din server. Acesta este un mod mai sigur de transmitere a parolei deoarece in cazul in care baza de date va deveni publica, decriptarea parolelor este mult mai complexa decat criptarea acestora.

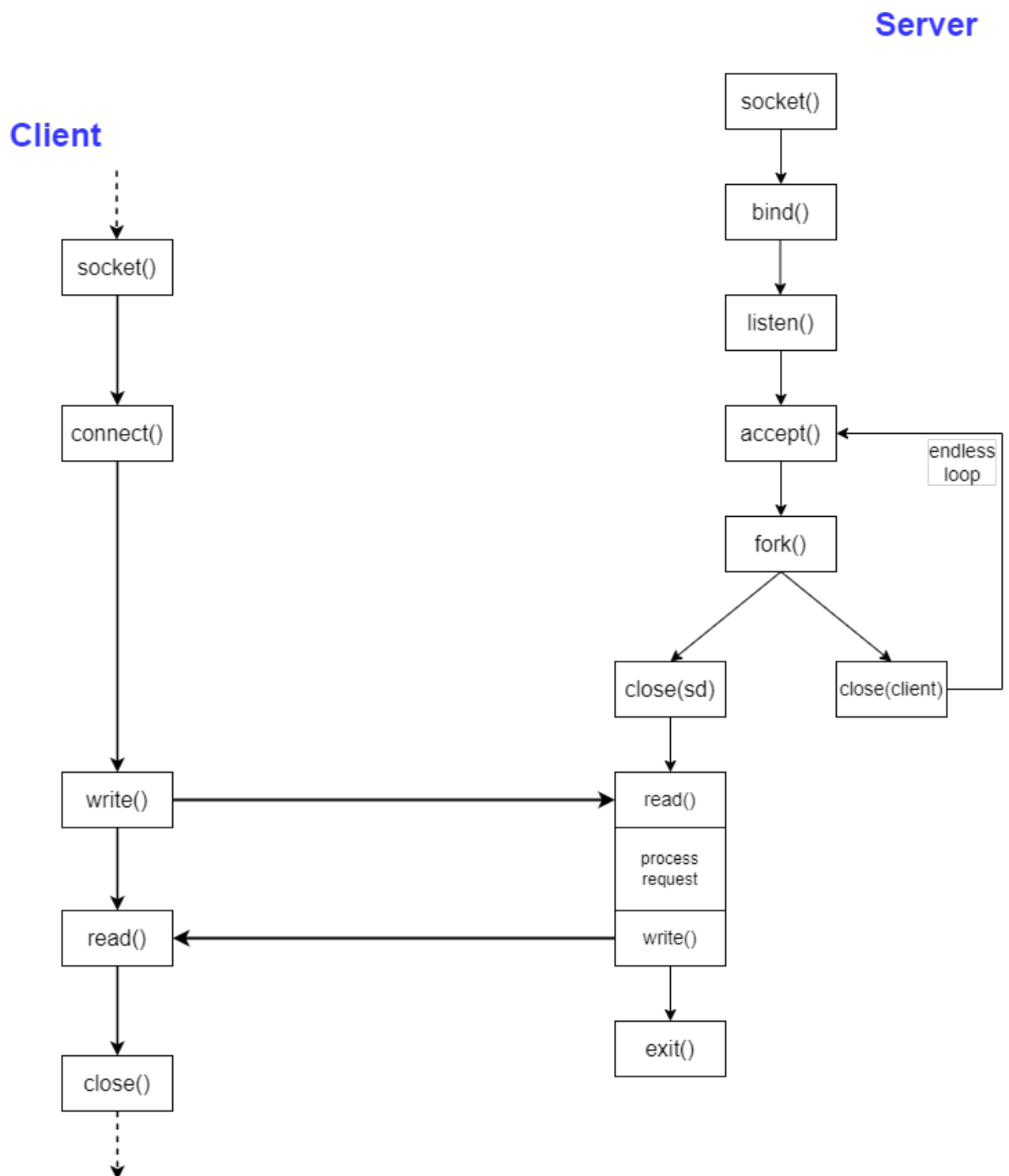


Figure 1: Model client/server TCP

4 Detalii de implementare

```
1 sprintf(sql, "SELECT * FROM users WHERE username = '%s' AND password = '%s';", username,
   h_pass);
2 rc = sqlite3_exec(db, sql, callback, str, &zErrMsg);
3
4 if (rc != SQLITE_OK)
5 {
6     login_status = 0;
7     printf("SQL error: %s\n", zErrMsg);
8     fflush(stdout);
9     sqlite3_free(zErrMsg);
10 }
11 else if (strstr(str, username) && strstr(str, h_pass) && strstr(str, "blacklist = 0"))
12 {
13     sprintf(result, "Bine ai venit %s. Ati fost logat cu succes.\n\n", username);
14     login_status = 1;
15     bzero(str, max);
16 }
17 else if (strstr(str, username) && strstr(str, h_pass) && strstr(str, "blacklist = 1"))
18 {
19     login_status = 0;
20     sprintf(result, "Ne pare rau, esti in blacklist. Pentru a-ti reactiva contul
   contacteaza adminul.\n\n");
21     bzero(str, max);
22 }
23 else
24 {
25     login_status = 0;
26     sprintf(result, "Username sau parola incorecta. Incercati din nou folosind comanda
   login!\n\n");
27     bzero(str, max);
28 }
```

Listing 1: Porcesul de verificare la login

```
1 void write_file(int client)
2 {
3     int n;
4     FILE *f;
5     char *filename = "./SERVER/send";
6     char buffer[1024];
7
8     f = fopen(filename, "w");
9     while (1)
10     {
11         n = read(client, buffer, 1024);
12         if (n <= 0)
13         {
14             break;
15             return;
16         }
17         fprintf(f, "%s", buffer);
18         bzero(buffer, 1024);
19     }
20     return;
21 }
```

Listing 2: Functie de write in fisier

```

1 void send_file(FILE *f, int sd)
2 {
3     int n;
4     char data[1024];
5     bzero(data, 1024);
6
7     while (fgets(data, 1024, f) != NULL)
8     {
9         if (write(sd, data, sizeof(data)) == -1)
10        {
11            perror("Eroare la trimiterea fisierului");
12            errno;
13        }
14        bzero(data, 1024);
15    }
16 }

```

Listing 3: Functie de send file

5 Concluzii

Programul ar putea fi optimizat in primul rand prin folosirea unei interfete grafice care sa usureze modul de folosire al aplicatiei. In al doilea rand, programul ar putea fi imbunatatit prin adaugarea unui stream separat pentru transmiterea fisierelor, dar si prin extinderea efectiva a functiilor programului: ar putea fi adaugata o functie *stat* care sa afiseze diferite detalii despre fisierul care il selectam. Mici modificari usor de realizat ar fi modificarea functiei de criptare in una mai sigura (ex. SHA-256), aplicarea functiei `getpass()` pentru a ascunde cand scriem parola in terminal etc.

In ceea ce priveste tehnologiilor utilizate, un server UDP ar fi mai bun din punctul de vedere al vitezei, fata de TCP. Acesta ar fi mai performant dar nu garanteaza integritatea datelor.

References

- [1] <https://profs.info.uaic.ro/~matei.madalin/retele/>
- [2] <https://profs.info.uaic.ro/~computernetworks/>
- [3] https://en.wikipedia.org/wiki/Transmission_Control_Protocol
- [4] https://en.wikipedia.org/wiki/File_Transfer_Protocol
- [5] <https://www.geeksforgeeks.org/sql-using-c-c-and-sqlite/>
- [6] [https://en.wikipedia.org/wiki/Qt_\(software\)](https://en.wikipedia.org/wiki/Qt_(software))
- [7] <https://sectigostore.com/blog/hashing-vs-encryption-the-big-players-of-the-cyber-security-world/>
- [8] https://en.wikipedia.org/wiki/Hash_table
- [9] <https://app.diagrams.net/>
- [10] https://www.overleaf.com/learn/latex/Learn_LaTeX_in_30_minutes