**Proiect 1 – Programarea Aplicațiilor Distribuite**

**Sincronizare de directoare**

Proiect realizat de:

Matheiu Flavius-Dan, Miuțel Andreea, Micoară Denis, Truican Ana

Proiectul contine un server si mai multi clienti. Serverul ruleaza in background si preia request-uri de la clienti. Clientii, prin aceste request-uri doresc sa aduca pe sistemul local al lor diferite directoare de pe server. Cand se conecteaza un client la server, acesta trimite un mesaj cu numele directorului dorit, iar serverul trimite directorul cu tot cu continutul acestuia(alte directoare, fisiere).

Serverul este concurent, adica accepta mai multe conexiuni cu diferiti client in acelasi timp(poate trata mai multe request-uri in paralel).

**Server(server.c)**

~~-~~ se creeaza socket-ul, se fac setarile pentru receptia datelor

- se creeaza un proces fiu pentru fiecare conexiune cu clientii

- daca totul se executa corect, se afiseaza un mesaj de confirmare a conexiunii cu clientul

Pentru fiecare client care acceseaza serverul:

- se asteapta comanda de la client(trimite\_director), in care se specifica si numele directorului cerut

- daca acea comanda este valida, acesta incepe trimiterea fisierelor spre client, astfel:

→ se verifica daca directorul exista pe server, daca nu se trimite mesajul inexistent\_directory catre client

→ se trimite numele directorului, numele fiecarui fisier/director gasit in interior, iar apoi, daca este fisier obisnuit, se trimite continutul fisierului.

Mesaje trimise de server catre client, atunci cand are loc transferul de date:

→ director <nume> - pt. Directoare(pentru directoare se apeleaza recursiv functia de parcurgere de director)

→ filename <nume> - pt. Fisiere,

→ date\_fisier <nume> - pt. Continut fisiere

→ inexistent\_directory – pt. Director inexistent pe server

- functia trimitere\_X(char \*name,int tip) trimite catre client unul din mesajele de mai sus, in functie de tip, care este un intreg(0,1,2,3).

- functia parcurgere\_dir(char \*dirname) parcurge directorul cerut de client si trimite datele necesare.

- functia sincronizare este functia apelata de fiecare fiu(deci client) si verifica daca comanda de la client este valida. Daca este valida, trimite numele directorului initial, iar apoi apeleaza functia parcurgere\_dir.

- daca totul decurge cum trebuie, se afiseaza la final un mesaj de succes pe server.

**Client(client.c)**

- se creeaza socket-ul, si se conecteaza clientul cu serverul prin socket.

- se afiseaza formatul de comanda posibila pentru lucrul cu serverul (trimite\_director <nume\_dir>)

- clientul trimite mesajul dat de utilizator spre server

- clientul primeste de la server mesajele specificate mai sus si reactioneaza astfel:

--- la comanda director: creeaza un director in directorul curent cu numele dat

--- la comanda filename: creeaza un fisier cu numele dat

--- la comanda date\_fisier: primeste datele si le scrie in fisierul nou creat mai devreme

--- la comanda inexistent\_directory: inchide clientul si socket-ul aferent.

- functia actiuni\_client(int sockfd, char \*sendline) se apeleaza daca comanda data de utilizator pentru server e valida. Aceasta trimite comanda catre server, si intra intr-o bucla in care asteapta mesajele de la server(cele de mai sus). Se mai afiseaza si mesajele venite de la server pentru verificare.

- daca totul decurge cum trebuie, se afiseaza la final un mesaj de succes pe server si inchide clientul.

**Observatii:**

- intre server si client se transmit mesaje de forma celor de mai sus, sau mesaje cu continutul fisierelor care trebuie transmise.

- la majoritatea functiilor folosite se testeaza cazurile de eroare posibile si se afiseaza mesaje de eroare corespunzatoare, urmate de iesirea din program prin exit().

**Mod de utilizare:**

-arhiva trimisa contine acest document de descriere, si un folder cu proiectul, unde serverul si clientul se afla in directoare diferite

-pentru server, se porneste programul executabil fara niciun argument.

-pentru client, se porneste executabilul cu un argument care reprezinta adresa serverului cu care se face conexiunea. Apoi clientul intra in modul de trimis comenzi catre server. Tiparul comenzii: trimite\_director <nume\_director>.

-in directorul cu serverul se afla si 2 directoare de test pentru program.