# Lunch01

## Retele de calculatoare - Proiect

Danu Teodor

Universitatea "Alexandru Ioan Cuza" Facultatea de Informatica
danu.teo20@gmail.com

**Abstract.** O aplicatie care implementeaza un TCP server concurent

## 1 Introducere

Acest proiect este un server concurent in care clientii trimit aleatoriu un numar catre server, unde se pot conecta N clienti [1]

Clientii trimit un numar ales aleatoriu, care reprezinta un fel de mancare, intre 1 si 5 catre server din T in T secunde. Serverul va contoriza fiecare numar si va trimite catre clientii un mesaj, in functie daca au nimerit sau nu numarul cu cele mai multe cereri.

Serverul trimite mesajul "Masa e servita" daca clientul a cerut numarul preferat, atunci clientul va raspunde cu "Satul!" si isi va incheia executia. Pentru cei care nu au cerut numarul prefarat le va trimite mesajul "Indisponibil!". Acestia vor incerca din nou, trimitand serverului din nou acelasi numar.

Daca clientul a fost refuzat de 3 ori acesta va afisa "Schimb cantina! Aici mor de foame!" si isi v-a incheia executia

## 2 Tehnologii utilizate

Protocolul TCP este foarte fiabil si sigur. Totusi, are cateva dezajantaje precum: este mai lent decat UDP, foloseste mai multe resurse si nu suporta broadcasting. Protocolul TCP foloseste 3 way handshake prin care se asigura ca nu se pierd date intre expeditor si destinatar, de aici vin si aceste dezavantaje.

Am folosit protocoliul TCP pentru ca se specifica in enuntul proiectului si pentru ceea ce trebuie sa faca aplicatie nu se pot pierde pacheta de date intre client si server.

Mai folosesc un fisesr unde se scriu numerele trimise de clienti ca apoi, serverul sa poata sa vada care e numarul cu cele mai multe cereri

## 3 Arhitectura aplicatiei

Clientulv-a geneara un numar intre 1 si 5 pe care il trimite serverului.

Serverul primeste in T secunde cate un numar de la fiecare client conectat, pe care ii scrie intr-un fisier text, ca mai apoi sa foloseasca un vector de frecventa

de la 1 la 5 ca sa verifice care e cel mai cerut numar. Dupa ce verifica care e cel mai cerut numar il compara cu numarul pe care l-a primit de la client, iar daca clientul a trimis numaru preferat atunci o sa primeasca de la server mesajul "Masa e servita!", daca nu o sa primeasca "Indisponibil!".

Daca measjul primit e "Masa e servita!" atunci clientul afiseaza "Satul!" si isi incheie execcutia. Daca primeste "Indisponibil!" atunci va incerca din nou cu un nou numar ales tot random intre 1 si 5.

Daca clientul a incercat de 3 ori si nu a nimerit numarul preferat atunci acesta va afisa "Schimb cantina! Aici mor de foame!" si isi v-a incehia executia.

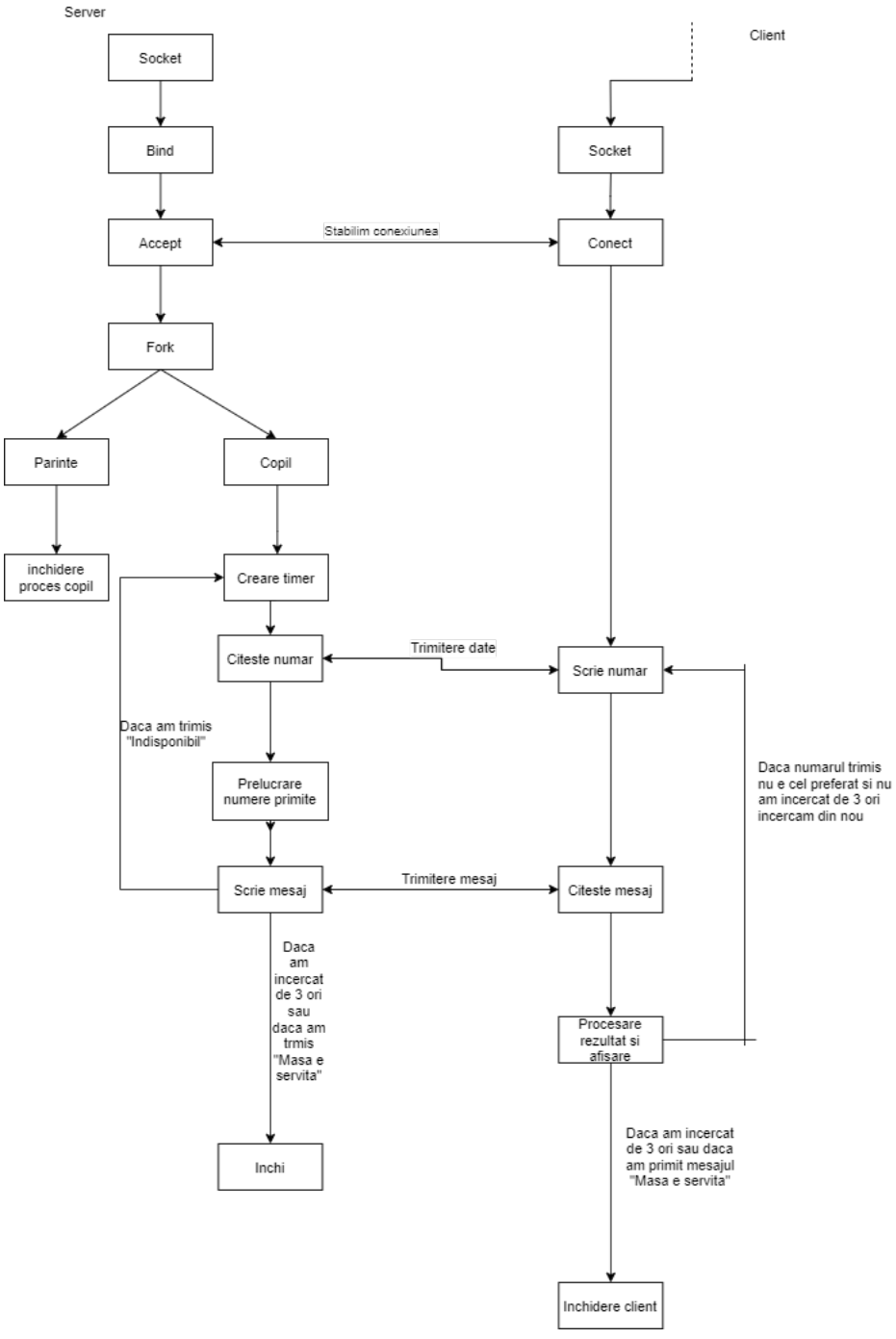


**Fig. 1.** Application flowchart

## 4 Detalii de implementare

Urmatoarele secvente de cod vor fi din server.c

Aici primim numarul de la client ca si un char si apoi tratam daca ne da eraoare la read si apoi punem numarul in fisier

```
char primim;
                if (read(client, &primim, sizeof(char)) <= 0) {
                    perror("[server]Eroare la read() de la client.\n");
                    close(client);
                    continue;
                }
                printf("am primit felul %c\n", primim);
fflush(stdout);

                FILE *f = fopen("feluri.txt", "a");
                fprintf(f,"%c" ,primim);
fflush(f);
                fclose(f);
```

Acest algortm o sa calculeze care e cel mai cerut numar si se va folosi de un delay ca sa poata sa trimita toti clientii un numar

```
time_t delay1; time(&delay1);
                time_t delay2; time(&delay2);
                while(delay2-delay1<0.5) time(&delay2);
                f = fopen("feluri.txt", "r");

                int depuse=0, fel, dorit, dorit_freq;

                    int freq[6] = {0};
                    dorit = -1; dorit_freq = -1;

                    fseek(f,0,SEEK_SET);

                    char chr[size]={0};
                    fread(chr,1,size,f);

                    for(int i=0;i<size;i++) {
                        char c=chr[i];
                        if(c<'1'||c>'5')break;
                        fel=c-'0';
                        freq[fel]++;
                        if(dorit_freq < freq[fel]) {
                            dorit_freq = freq[fel];
                            dorit = fel;
```

```
                    }
                    depuse++;
                }
```

Verificam daca numarul trimis de client e la fel cu cel mai cerut numar. Daca e atunci ii trimitem "Masa e servita" altfel ii trimitem "Indisponibil"

```
if(primim == (char)(dorit+(int)'0')) {
                    strcpy(msgrasp,"Masa e servita");
                    incercari=4;
                }
                else {
                    printf("ghinion\n");fflush(stdout);
                    incercari++;
                    if(incercari<3) strcpy(msgrasp,"Indisponibil");
                    else strcpy(msgrasp,"rip");
                }
```

Urmatoarele secvente de cod vor fi din client.c

Generam numarul pe care vrem sa il trimitem serverului si apoi il transformam in char si il trimitem.

```
msg=rand()%5+1;
        printf("clientul cere felul %d\n",msg);fflush(stdout);

        char trimitem = msg+(int)'0';
        write(sd,&trimitem,sizeof(char));

        char raspuns[30]={0};
```

Analizam in client mesajul pe care l-am primit de la server si in fuctie de acest mesaj hotaram ce facem mai departe.

```
if(strcmp(raspuns, "Masa e servita") == 0){
             printf("Satul!");
             fflush(stdout);
             close(sd);
             return 0;
         }
         else if(strcmp(raspuns, "Indisponibil") == 0){
             incercari++;
         }
         else if(strcmp(raspuns, "rip") == 0) {
             printf("Schimb cantina! Aici mor de foame!\n");
             fflush(stdout);
             close(sd);
             return 0;
         }
```

## 5 Concluzii

Lunch01 este o aplicatie in care serverul primeste un numar intre 1 si 5 de la clienti si vede care e cel mai cerut si apoi ii anunta pe clienti daca au trimis sau nu numarul preferat

## References


1. Lista proiecte
   `https://profs.info.uaic.ro/~computernetworks/ProiecteNet2019.php`
2. Cursuri
   `https://profs.info.uaic.ro/~computernetworks/cursullaboratorul.php`