

Tema 3 - Trimiterea de mesaje folosind socketi TCP

Data publicarii: 18.04.2013

Data ultimei modificari a enuntului: 03.05.2013

Termen de predare: 08.05.2013, ora 23:55

Responsabil: Catalin Leordeanu (catalin.leordeanu@cs.pub.ro)

Obiective

Scopul temei este realizarea unui sistem de partajare a fișierelor în rețea. Obiectivele temei sunt:

- Înțelegerea mecanismelor de dezvoltare a aplicațiilor folosind sockets.
- Dezvoltarea unei aplicații practice de tip client-server ce folosește sockets.

Descrierea temei

Se cere dezvoltarea unei aplicații client-server folosind socketi TCP pentru schimbul de mesaje și fișiere. În cadrul sistemului se consideră existența a două tipuri de entități: clienți și un server central ce ajută la descoperirea clienților.

La pornire serverul va primi ca parametru un port pe care va asculta cereri de conexiune. Modul de apelare al serverului este:

```
./server <port_server>
```

Un client va primi ca parametru al execuției un nume cu ajutorul căruia se va identifica în sistem, un port pe care va aștepta conexiuni din partea altor clienți, adresa și portul serverului central de descoperire (se consideră existența unui singur astfel de server în sistem). Modul de rulare al unui client este următorul:

```
./client <nume_client> <port_client> <ip_server> <port_server>
```

Serverul va fi folosit pentru descoperirea clienților conectați iar trimiterea de mesaje și de fișiere se va realiza direct între clienți. Atât serverul cât și clienții vor porni un server socket și vor folosi apelul select pentru multiplexarea comunicății.

Funcționalitate

La pornire serverul va crea un socket și va aștepta cereri de conexiune pe portul specificat. Un client se conectează la serverul central și trimite datele de identificare (numele primit la pornire, portul pe care asculta eventuale cereri de conexiune). Serverul poate răspunde cu accept sau reject, în funcție de numele clientului (dacă deja există un client în sistem cu respectivul nume deoarece nu sunt admise duplicate de nume). Serverul va reține pentru fiecare client conectat adresa IP, numele și portul pe care acesta asculta.

După conectarea la server un client poate primi următoarele comenzi de la tastatură:

1) listclients

Clientul trimite o cerere serverului care îi va întoarce lista tuturor clienților conectați. Lista clienților va fi afișată la consolă.

2) infoclient <nume_client>

Î se cer serverului și se vor afișa informații suplimentare despre un client. Serverul va întoarce numele clientului, portul pe care acesta asculta și timpul scurs de la conectarea acestuia la server.

3) message <nume_client> <mesaj>

Se trimite un mesaj unui alt client. Aceasta se face prin realizarea unei conexiuni directe la clientul care

trebuie sa primeasca mesajul. Clientul destinatie va afisa mesajul primit si numele clientului de la care l-a primit. Afisarea mesajului primit va fi sub forma:

[Timestamp][Sursa]Mesaj

4) broadcast <mesaj>

Functioneaza asemanator cu comanda message, dar este trimis un mesaj catre toti clientii inregistrati in server in acel moment.

5) sendfile <client_destinatie> <nume_fisier>

Se transfera un fisier catre un alt client. Acest lucru se face printr-o conexiune directa catre clientul destinatie. Trimiterea unui fisier trebuie facuta in segmente de maxim 1024B si intre doua bucati transferate trebuie sa verificati daca aveti cumva o noua comanda de executat din partea utilizatorului (folositi select pentru acest lucru) . Este nevoie de acest mecanism pentru ca un client sa nu fie blocat in timpul transferului unui fisier de dimensiune mare. Clientul destinatie va primi fisierul sub numele nume_fisier_primit.

6) history

Se afiseaza la consola toate mesajele si numele fisierelor primite e catre client de la pornirea sa. Ordinea de afisare este aceeaasi cu ordinea in care s-au dat comenzile.

7) quit

Clientul trimite un mesaj serverului prin care anunta ca va parasi sistemul, inchide toate conexiunile si iese.

Serverul poate primi de la tastatura doar urmatoarele comenzi:

1) status

Afiseaza lista clientilor conectati, adresele lor ip si porturile pe care acestea asculta. ~~fișierele pe care le partajeaza~~

2) kick <nume_client>

Elimina un client din sistem. Ii trimite un mesaj clientului ca sa il anunte ca trebuie sa inchida si il elimina din lista de clienti activi.

3) quit

Se inchide serverul. Cand clientii detecteaza ca a fost inchisa conexiunea cu serverul vor iesi si ei.

Serverul si clientii pot apela comenzile in orice ordine. Serverul central trebuie să mențină intern un socket pe care ascultă cereri de noi conexiuni din partea clienților. Un client odată conectat și înregistrat poate menține conexiunea respectivă și o poate refolosi pentru schimbul de mesaje. Pentru primirea de comenzi de la tastatura in paralel cu mesajele se poate adauga file descriptor-ul pentru stdin in select. Se vor trata cazurile de eroare in comunicatie, de exemplu daca se incearca trimiterea unui mesaj catre un client care nu exista.

Cerinte privind realizarea temei

Tema (client și server) va fi realizată folosind sockets stream (peste TCP) în C sau C++. Apelurile de sistem și mecanismele necesare pentru realizarea temei sunt descrise pe larg în suportul de curs și în cadrul laboratorului de socketi TCP. Formatele de mesaje și protocolul de comunicație folosit în implementarea aplicației sunt la latitudinea voastra dar trebuie să fie descrise în fișierul Readme (cu justificare asupra alegerii). Este necesara si uploadarea unui makefile in arhiva temei. Pentru multiplexarea comunicației folosiți apelul select (studiat în cadrul laboratorului). Nu aveți voie să folosiți crearea de procese sau fire de execuție. Rezumați-vă la folosirea apelului select.