Tema 3 - Trimiterea de mesaje folosind socketi TCP

Data publicarii: 09.04.2012
Data ultimei modificari a enuntului: 09.04.2012
Termen de predare: 24.04.2012, 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 aplicatii client-server folosind socketi TCP pentru schimbul de mesaje si fisiere. 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 1334
```

Un client va primi ca parametru al execuției un nume cu ajutorul căruia se va identifica în sistem și adresa și portul serverului central de descoperire (se consideră existența unui singur astfel de server în sistem). Modul de rulare al unui client este urmatorul:

```
./client nume_client 192.168.1.1 1334
```

Serverul va fi folosit pentru descoperirea clientilor conectati iar trimiterea de mesaje si de fisiere se va realiza direct intre clienti. Atat serverul cat si clientii vor porni un server socket si vor folosi apelul select pentru multiplexarea comunicatiei.

Functionalitate

La pornire serverul va crea un socket si va astepta 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 retine pentru fiecare client conectat adresa IP si portul pe care acesta asculta.

Dupa conectarea la server un client poate primi urmatoarele comenzi de la tastatura:

1) `listclients`
Clientul trimite o cerere serverului care ii va intoarce lista tuturor clientilor conectati. Lista clientilor va fi afisata la consola.

2) `infoclient nume_client`
I se cer serverului si se vor afisa informatii suplimentare despre un client. Serverul va intoarce numele clientului, portul pe care acesta asculta si 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.

4) `sharefile nume_fisier`
Se trimite un mesaj serverului prin care anunta faptul ca se partajeaza un nou fisier. Serverul doar va inregistra numele noului fisier in lista asociata clientului respectiv.

5) `unsharefile nume_fisier`
Se trimite un mesaj serverului continand numele fisierului care trebuie sters din lista fisierelor partajate.

6) `getshare nume_client`
Se trimite un mesaj serverului prin care se cere lista fisierelor partajate de catre un anumit client.

7) `getfile nume_client nume_fisier`
Se transfera un fisier de la un alt client. Acest lucru se face printr-o conexiune directa la clientul sursa. Trimiterea unui fisier trebuie facuta in segmente de maxim 1024B si intre doua bucati transferate sa verificati daca aveti cumva o noua comanda de executat din partea utilizatorului. Este nevoie de acest mecanism pentru ca un client sa nu fie blocat in timpul transferului si sa poata sa primeasca si alte comenzi. Clientul a primi fisierul sub numele *nume_fisier_primit*.

8) `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, porturile si fisierele pe care le partajeaza

2) `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. 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 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.