Ministère de l'éducation, de la culture et de la recherche
de la République de Moldova
Université technique de Moldavie
Faculté d'Ordinateur, Informatique et Microélectronique
Filière francophone "Informatique"

Compte Rendu

Programmation réseau

Travail pratique nr.4

Thème: Aplicație Client-Server TCP

Effectué par l'étudiant(e) de gr FI-181 : Bonta Alexandr

Vérifié par le professeur : Buldumac Oleg

Chișinau 2021

Proiectul pe github: https://github.com/sasa-bonta/PR

Scopul:

Să se creeze o aplicație Client-Server TCP utilizând Socket API

Pentru nota 9 si 10:

- Să definiți, proiectați și elaborați propriul protocolP

Atenție:

- Fiecare client trebuie procesat de către server într-un fir de execuție aparte
- Nu se admite aplicații simple de genul Echo Client-Server
- Aplicația poate fi consolă sau GUI
- Pentru cei ce doresc 9 și 10, ca exemplu luați protocoalele HTTP, SMTP unde clientul și serverul discută printr-un set de reguli bine definite. De exemplu, pentru o aplicație de tip chat pentru ca clientul să fie identificat de către server acesta trebuie să transmită un mesaj de tipul: HELLO-REQUEST după care serverul va răspunde cu HELLO-APPROVE. Mai detailat accesați ultimele 3 referințe de la sfîrșitul acestui fișier.

Întrebări la apărarea laboratorului:

• Ce este un protocol orientat pe conexiune ?

Mai poate fi numit protocol orientat pe corectitudinea datelor, se utilizeaza in special pentru a asigura integritatea informatiei transportate.

• Ce tipuri de aplicații beneficiază în general de utilizarea protocolului TCP ?

Aplicatiile care au nevoie de integritatea datelor, cum ar fi aplicatiile de transfer a fisierelor, posta electronica, aplicatiile ce necesita logare, sau un flux de date cu caracter personal (numere de carduri bancare, idnp, parole…)

• Cum TCP garantează că datele vor fi transmise cu succes ?

TCP creeaza o conexiune per to per, Transmite un numar de secvente pe care receptorul este capabil sa-l prelucreze la care se asociaza numarul de secventa, daca receptorul primeste numarul de packete definit de secventa, atunci receptorul raspunde cu un mesaj de confirmare (ACK) pentru packele expediate, in caz contrar, in caza ca serverul nu primeste confirmare pentru careva pcket atunci acesta este retransmis cu urmatoarea secventa de pakete.

• Diferența dintre blocking si non-blocking sockets

Blocking asteatpta neaparat raspuns din partea la server, iar operatiunile sunt realizate una dupa alta,

pe cand in nom-blocking clientul poate da start la conexiune iar intretimp sa faca alte operatii ca send(), recv(), daca utilizatorul in timp ce nu s-a realizat connect() apasa butonul cancel, putem chema metoda close().

• Diferența dintre blocking multithreaded și non-blocking single thread socket

Cum are loc procesul TCP Three Way Handshake ?

In prima faza, Clientul (cel care incepe conexiunea) ii va trimite serverului:

1. Un mesaj de sincronizare (**SYN**) sau de incepere a conexiunii
2. Serverul va raspunde cu o confirmare (**SYN-ACK**)
3. Clientul va raspunde si el cu un mesaj de confirmare (**ACK**)

• Cum are loc procesul TCP Three Way Handshake ?

• Numiti cele 4 apeluri de sistem necesare pentru a crea un server TCP

- LISTEN - în cazul unui server se așteaptă o solicitare din partea unui client
- SYN-SENT - se așteaptă din partea nodului pereche trimiterea unui segment TCP cu flagurile de SYN și ACK setate (starea este specifica clienților ce rulează protocolul TCP)
- SYN-RECEIVED - așteaptă din partea nodului pereche a confirmării ca răspuns la confirmarea de conectare trimisa către acesta (stare specifica serverelor cu TCP)
- ESTABLISHED - portul este pregătit pentru a trimite/primi date către/dinspre nodul pereche

• Care este rolul metodei bind() ?

Stabilirea adresei, socket-ul respectiv va primi doar cereri de conectare destinate adresei indicate in bind()

• Care este rolul metodei accept() ?

Apelul functiei accept() are ca efect crearea unui socket de cone-xiune, asociat unui client conectat (prin apelul connect()) la socket-ul de asteptare. Daca nu exista inca nici un client conectat si pentru care sa nu sa creat socket de conexiune, functia accept() asteapta piana la conectarea urmatorului client.

• Ce se întîmplă cînd apelați mai întîi connect() apoi bind() ?

Functia bind()poate apelata doar pentru un socket proaspat creat,caruia nu i s-a atribuit inca o adresa. Astfel bind() va esua.

• Ați avea vreodată nevoie să implementați un timeout într-un client sau server care utilizează TCP?

Desigur, in caz de retineri de lunga durata, v-a trebui sa afisam un mesaj de asteptare sau de esuare.

• Într-o conexiune TCP, clientul sau serverul trimite mai întâi datele ?

Mai intai trimite cereri, primeste raspuns, dupa care trimite datele.

• Care este adresa de loopback IPv4 și care este rolul ei ?

Intervalul de la 127.0.0.0 pana la 127.255.255.255, este utilizata pentru a identifica dispozitivul.

• De unde știe un sistem de operare ce aplicație este responsabilă pentru un pachet primit din rețea ?

Prin intermediul setarilor firewall, Prin folosirea unui firewall avem posibilitatea de a seta exceptii sau de a bloca traficul de date al anumitor aplicatii in functie de caz.

• Datele primite prin recv() au întotdeauna aceeași dimensiune cu datele trimise cu send() ?

Sistemul garanteaza sosirea la destinatie a tuturor octetilor trimisi(sau instiintarea receptorului, printr-un cod de eroare, asupra caaderii cone-xiunii), in ordinea in care au fost trimisi. Nu se pastreaza insa demarcarea intre secventele de octeti trimise in apeluri send() distincte. De exemplu, este posibil ca emitatorul sa trimita, in doua apeluri succesive, sirurile abc si def,iar receptorul sa primeasca, in apeluri recv() succesive, sirurile ab, cde , f

• Este acceptabil să închei executia programului dacă este detectată o eroare de rețea ?

Da este acceptabil in cazul in care este o eroare de hardware, afisand mesajul informativ.

• Puteți îmbunătăți performanța aplicației prin dezactivarea algoritmului Nagle ?

Nagle algoritmul este utilizabil doar cu TCP. Alte protocoale, inclusiv UDP , nu o acceptă.

Aplicațiile TCP care au nevoie de răspuns rapid la rețea, cum ar fi apelurile telefonice prin Internet sau jocurile de tip shooter pentru prima persoană, ar putea să nu funcționeze bine când Nagle este activat. Întârzierile cauzate în timp ce algoritmul necesită un timp suplimentar pentru a asambla mai multe bucăți de date mai mici poate provoca o vizibilitate vizibilă pe ecran sau într-un flux audio digital. Aceste aplicații dezactivează de obicei Nagle.

• Ce instrumente listează socket-urile TCP deschise în sistemele de operare Windows și Linux ?

Netstat, ss

• Tehnici de sincronizare a firelor de execuții

Using locks in the with statement - context manager
Condition objects with producer and consumer
Producer and Consumer with Queue
Semaphore objects & thread pool

Proiectul pe github: https://github.com/sasa-bonta/PR