**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare Informatică și Microelectronică**

**Departamentul** **Ingineria Software şi Automatică**

**Raport**

**Lucrare de laborator nr.4**

Disciplina: Programarea în rețea

Tema: Aplicație Client-Server TCP

A îndeplinit: st.gr.TI-171 Verhovețchi Daniel

A controlat: Buldumac Oleg

Chișinău 2020

**Scopul lucrării:**

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

**Intrebări de control:**

1. Ce este un protocol orientat pe conexiune ?

Este un protocol care asigură ca un flux de octeți trimiși de o mașină să ajungă fără erori pe orice altă mașină din rețea.

1. Ce tipuri de aplicații beneficiază în general de utilizarea protocolului TCP ?

Aplicații care pot funcționa în regim normal numai dacă toți biții au fost primiți de la o sursă. Exemple de aplicații: World Wide Web, poșta electronică, aplicații pentru transferul de fișiere.

1. Cum TCP garantează că datele vor fi transmise cu succes ?

Sunt utilizate mai multe mecanisme, de exemplu: dacă clientul nu raportează serverului că a primit pachetul cu date, serverul trimite din nou același pachet. Clientul sortează pachetele primite pentru a fi în ordinea corectă. Înainte de a trimite datele, serverul verifică prin metoda numită “3 way handshake” dacă clientul poate primi datele.

1. Diferența dintre blocking si non-blocking sockets

Prin blocking sockent se înțelege că doar un singur fir de execuție poate trata o conexiune prin socket, deoarece firul de execuție devine blocat atunci cînd așteaptă o conexiune, sau cînd așteaptă input de la socket. Acest tip de socket este sincron, pe când non-blocking socket – asincron.

1. Cum are loc procesul TCP Three Way Handshake ?

Să presupunem că sunt două calculatoare: Host A și Host B. Host A trimite un packet cu mesajul SYN (SYNchronize) la Host B. Host B primește acest pachet și trimite înapoi un pachet cu mesajul SYN-ACK (SYNchronize-ACKnowledgement). Host A primește acest mesaj și trimite înapoi mesajul ACK. Conexiunea este stabilită.

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

Socket, bind, listen, accept.

1. Care este rolul metodei bind() ?

Apelul metodei bind() asociază un socket cu o adresă de rețea și un port.

1. Care este rolul metodei accept() ?

Prin apelarea acestei metode se așteaptă, acceptă, o conexiune. Firul de execuție este blocat până când un client se conectează la server.

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

Nu, protocolul TCP are implementat propriul mecanism pentru timeout.

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

Clientul inițializează Three Way Handshake -ul, și tot clientul trimite cererea primul.

1. Care este adresa de loopback IPv4 și care este rolul ei ?

127.0.0.1 – localhost. Este o adresă folosită pentru testare, astfel pachetele trimise la această adresă sunt percepute ca pachete ce vin din rețea.

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

Fiecare aplicație are atașat un port virtual, astfel în dependență de portul specificat în pachet, sistemul de operare redirecționează pachetul la portul respectiv.

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

Nu.

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

Nu.

1. Puteți îmbunătăți performanța aplicației prin dezactivarea algoritmului Nagle ?

Numai în cazuri speciale. În cazul când este nevoie de a transmite imediat datele unidirecțional.

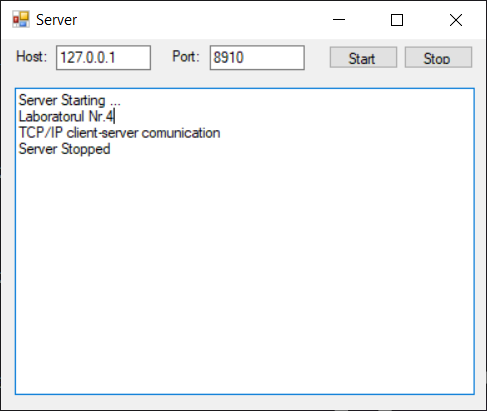
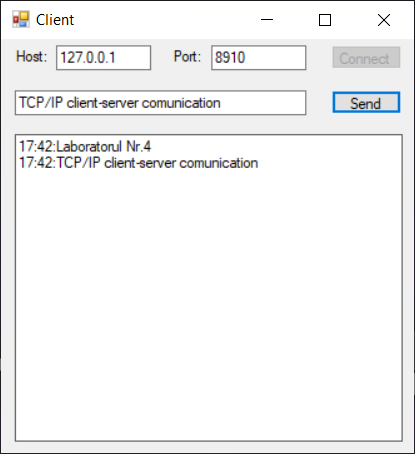
1. Ce instrumente listează socket-urile TCP deschise în sistemele de operare Windows și Linux ?

În Linux este fișierul cu calea /proc/net/tcp, comanda ss. Pentu Windows și Linux – netstat.

1. Tehnici de sincronizare a firelor de execuții

Spinlock, Barriers, Semaphores, Mutex.

**Rezultate obținute:**



**Concluzie:**

In urma efectuării acestei lucrări de laborator s-au acumulat competențe de lucru cu protocolul TCP. De asemenea au fost căpătate cunoștințe în ceea ce privește comunicarea Client/Server și trimiterea pachetelor prin TCP/IP.

În lucrarea dată s-a realizat o aplicație GUI care reprezenta un chat. Clienții se puteau conecta la server și respectiv comunica între ei.

În final, am analizat rezultatele și formulat concluzii.

**Github link:** [**https://github.com/Verhovetchi/PR**](https://github.com/Verhovetchi/PR)