Ministerul Educaţiei, Culturii şi Cercetării Republicii Moldova

Universitatea Tehnică a Moldovei

Facultatea Calculatoare Informatică și Microelectronica

Departamentul Ingineria Software şi Automatică

**Referat**

**Disciplina: Programarea în rețea**

**Laboratorul 5**

Tema: Aplicație Client-Server UDP

A elaborat:

Pavlenco Ecaterina, st. gr. TI-171

A verificat:

Buldumac Oleg

Chișinău 2020

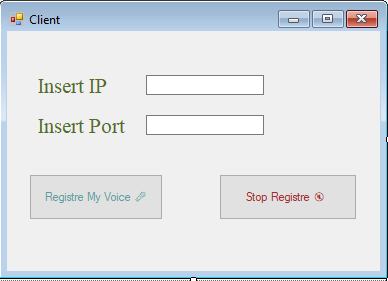
Să se creeze o aplicație Client-Server UDP utilizând Sockets API

În laboratorul dat s-a creat aplicație clinet server , care urmăresc conexiune UDP , anume s-a implimentat crearea aplicației de tip VoIP , adică voice on IP , și anume Clientul Transmite date în cazul nostru sunete către server . Pentru elaborarea acesteea s-a utilizat diverse clase biblioteci legate de protocolul UDP cât și de sunete .

De sigur pentru o implimentare mai reușită avem nevoie de 2 dispozitive diverse pentru implimentarea aceste aplicații atâta timp cât aici noi înregistrăm sunetul el deja este transmis către partea server și îl redă astfel se primește răsunare . Referitor la program vocea adică sunetul primit noi îl convertim în biți care la rândul său este transformat de către un Thread pentru al îl reda înapoi în formă de sunet.

Forms pentru părțile client și server:





Acest laborator lucrează astfel : Se înserează Ip și Porturile pe ambele părți pentru a fi creată conexiunea, după se urmărește pe server vocea pe acelaș port cu care este conectat . După dacă apăsăm la înregistrare Voce de pe partea client se înregistrează și se transmite către server cu o scurtă întârziere , deja apăsând la stoparea vocii aceasta încetează de a fi transmisă.

Link:

<https://github.com/katy198/PR-laboratorul5>

Video:



Întrebări

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

Aplicații care pot tolera unele pierderi de date, dar necesită o întârziere mică sau deloc

Aplicații cu tranzacții simple de cerere și răspuns

Comunicări unidirecționale în care fiabilitatea nu este necesară sau poate fi gestionată de aplicație

• De ce protocolul UDP nu garantează că datele vor fi transmise cu succes ?

UDP nu retransmite pachetele cu date pierdute.

• Diferența dintre blocking si non-blocking sockets

Socketurile non-blocking au efect ca la accept(), cînd apelăm accept și nu există deja client care să conecteze acesta returneaza Operation Would Block.Pe când la blocking este valabil precum ar fi la write() și/sau connect(), când le executăm conexiunea se blochează până când operațiunea este completă.

• În protocolul TCP există Three Way Handshake, de ce în UDP nu există ?

Scopul Three Way este de stabilirea conexiunii între părți ce nu este prevăzut de protocolul UDP (este un protocol orientat fără conexiuni).

• Numiti cele 2 apeluri de sistem necesare pentru a crea un server UDP

bind()

recyform()

• Care este rolul metodei bind() ?

Are rolul de a asocia socketul la adresa locala, adică adresa IP, portul, și familia de adrese.

• Care este rolul metodelor sendto() și recvfrom() ?

Sendto() trimite date către un socket la adresa specificată.

Recvfrom() este folosit pentru apelul de sistem, poate fi utilizat pentru a primi date pe un socket indiferent dacă este sau nu conectat.

• Care este dimensiunea antetului unui pachet UDP în octeți ?

4 câmpuri fiecare cu lungimea a câte 2 octeți

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

Clientul mai întâi transmite date .

• Care este adresa de loopback IPv6 și care este rolul ei ?

Pentru loopback IPv6, adresa este 0000: 0000: 0000: 0000: 0000: 0000: 0000: 0001/128

• Datele primite prin recvfrom() au întotdeauna aceeași dimensiune cu datele trimise cu sendto() ?

NU . Protocolul UDP nu răspunde de pierderea datelor , de aceea acestea nu se retransmit ca la protocolul TCP.

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

Nu , ca la protocolul TCP exită un șir de probleme de rețea ce pot fi înlăturare ,fără închiderea aplicației.

• De ce nu este folosit algoritmul Nagle în protocolul UDP ?

Algoritmul Nagle este folosit la TCP pentru îmbunătățirea și asigurarea tranmiterii datelor , cee ace la UDP nu e posibil deoarece aici se utilizează date directe din timp real ce trebuie transmise , iar Nagle duce la încetinirea programului.

• Același program poate folosi UDP și TCP ?

TCP și UDP folosesc conexiuni diferite de aceea nu e posibil ca acestea să folosească aceeași aplicație

• Diferența dintre aplicații UDP Unicast, Broadcast, și Multicast

Unicast – un client un server

Broadcast – un client mai multe servere

Multicast – mai mulți client mai multe servere

• Ce face mai ușor multiplexarea cu UDP decît cu TCP ?

Nu se face conexiunea directă client-server.

• În protocolul UDP este un antet „Total length”, cum se calculează și care este rolul lui ?

Datele sunt 16 octeți, lungimea antetului UDP este de 8 octeți, deci raportul este : 16/8+16 = 2/3

Total length = 001C16 = 28 bytes