TRƯỜNG ĐẠI HỌC CẦN THƠ

**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG  
KHOA TRUYỀN THÔNG ĐA PHƯƠNG TIỆN**

-----------------------------

**ĐỀ THI KẾT THÚC HỌC KỲ**

**HỌC KỲ 2, NĂM HỌC 2022 - 2023**

* **Học phần:** Lập Trình Ứng Dụng Mạng – TN414
* **Lớp học phần:** TN41401
* **Thời gian làm bài:** 120 phút
* **Lưu ý:** Sinh viên được phép sử dụng 1 trang tài liệu 2 mặt A4
* **Nội dung đề thi:**

**Câu 1 (4 điểm):** Viết chương trình Simple Chat có cho phép thu hồi tin nhắn

* Sinh viên sử dụng UDPClient để gửi và nhận dữ liệu.
* Thiết kế form như hình 1 với các control sau:

- **txtIPR:** là TextBox để người dùng nhập IP của máy sẽ gửi tin nhắn đi.

- **txtPortR:** là TextBox để người dùng nhập Port của máy sẽ gửi tin nhắn đi.

- **txtPortL:** là TextBox để người dùng nhập Port của máy cục bộ nhận tin nhắn gửi đến.

- **btntConnect:** là nút **Connect,** khi nhấn vào sẽ kết nối với máy nhận tin nhắn theo cấu hình đã thiết lập.

- **lstMsg:** là ListBox hiển thị tin nhắn gửi đi và gửi đến.

- **btnThuHoi:** là nút **Unsent**, khi nhấn vào sẽ gọi hàm unsent()

- **txtMsg:** là TextBox để người dùng nhập nội dung tin nhắn gửi đi.

- **btnGui:** là nút **Send**, khi nhấn vào sẽ thực hiện việc gửi tin nhắn.

\* Hàm **unsent** thực hiện các việc như sau:

- Thay thế tin nhắn **người dùng đang chọn** thành **UNSENT MESSAGE [YOU]**

- Gửi chỉ số tin nhắn sang phía người nhận, tin nhắn có dạng: **UNS: <*chỉ số tin nhắn*>**

- Thay thế nội dung tin nhắn bên phía nhận thành **UNSENT MESSAGE [FRIEND**].

\* **Chú ý:**

- Người dùng chỉ có thể thu hồi tin nhắn do mình gửi.

- Tin nhắn gửi đi hiển thị trong listboxlà **SEN: <nội dung tin nhắn>**

- Tin nhắn nhận được hiển thị trong listboxlà **REC: <nội dung tin nhắn>**

| Hình 1. Giao diện ứng dụng Simple Chat | Các phương thức của lớp Listbox:   * **Listbox.Items**: mảng các phần tử * **Listbox.Items.Add**: thêm phần tử mới * **Listbox.Items[*index*]**=**<*nôi dung mới*>:** thay thế nội dung của phần tử ở vị trí **index** bằng **<nội dung mới>** |
| --- | --- |

**Câu 2 (6 điểm):** Sử dụng các lớp TCP thích hợp, viết ứng dụng quản lý tin nhắn như sau:

* **Mô tả ứng dụng:** 
  + Ứng dụng cho phép người dùng lưu tin nhắn trên server mà không cần đăng ký tài khoản. Để đọc tin nhắn, người dùng chỉ cần cung cấp tiêu đề tin nhắn.
* **Các yêu cầu cài đặt**
  + **Yêu cầu Chương trình Server** (4 điểm):
* Thiết kế form như hình 2 (1 điểm )
  + **txtPort**: là textbox để nhập số hiệu cổng server sẽ nhận dữ liệu đến
  + **btnStart**: là button, khi nhấn vào sẽ thực hiện khởi động server theo thông tin cấu hình đã thiết lập
  + **rtxInfo**: là richtextbox hiển thị các thông tin trạng thái, các lệnh nhận từ client
* Yêu cầu sử dụng MultiThread để phục vụ cùng lúc nhiều Client (1 điểm )
* Cài đặt các lệnh xử lý như sau: (2 điểm)
  + **CRE** **<tiêu đề> <nội dung tin nhắn>:** Tạo tin nhắn mới và máy chủ sẽ lưu tin nhắn vào tập tin có tên là <tiêu đề>.txt.
    - Nếu tập tin đã có, ghi đè nội dung cũ.
    - Nếu lưu thành công, phản hồi là **100. Tin nhan luu thanh cong**
    - Nếu lưu không thành công, phản hồi là **203. Khong the luu tin nhan.**
  + **REA <tiêu đề>:** Đọc tin nhắn lưu trong tập tin **<tiêu đề >.txt**.
    - Nếu tin nhắn tồn tại, phản hồi là nội dung tin nhắn.
    - Nếu tin nhắn không tồn tại, phản hồi là **204. Tin nhan khong ton tai.**
  + **Yêu cầu Chương trình Client** (2 điểm):
* Thiết kế form như hình 3 (1 điểm )
  + **txtServerIP:** là textbox để nhập IP của server nhận dữ liệu
  + **txtServerPort:** là textbox để nhập số hiệu cổng server nhận dữ liệu
  + **btnConnect:** là nút **Connect**, khi nhấn vào sẽ thực hiện kết nối đến server theo thông tin cấu hình đã thiết lập
  + **rtxInfo**: là richtextbox hiển thị các thông tin trạng thái, các phản hồi từ server
  + **txtTitle**: là textbox để nhập tài khoản người dùng
  + **txtMessage**: là textbox để nhập tin nhắn, (thuộ tính **multiline** = **true**)
  + **btnCreate**: là nút khi nhấn vào sẽ gửi lệnh **CRE <tiêu đề> <tin nhắn>** với **tiêu đề** là nội dung của **txtTitle** và **tin nhắn** là nội dung của **txtMessage**
  + **btnRead**: là nút khi nhấn vào sẽ gửi lệnh **REA <tiêu đề>** với **tiêu đề** là nội dung của **txtTitle**
* Cài đặt chức năng của client theo giao diện đã mô tả (1 điểm)

**\* Chú ý:** Các lệnh của client gửi đi đều nhận được phản hồi từ máy chủ và hiển thị bởi **rtxInfo**, nội dung và định dạng các thông báo phản hồi là tùy ý nhưng phải phù hợp với nội dung chương trình.

| Hình 2. Giao diện Server | Hình 3. Giao diện Client |
| --- | --- |

**Duyệt của Trưởng Khoa Giảng viên ra đề**

**Nguyễn Nhị Gia Vinh Phạm Trương Hồng Ngân**

TRƯỜNG ĐẠI HỌC CẦN THƠ

**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG  
KHOA TRUYỀN THÔNG ĐA PHƯƠNG TIỆN**

-----------------------------

**ĐÁP ÁN ĐỀ THI KẾT THÚC HỌC KỲ**

**HỌC KỲ 2, NĂM HỌC 2022 - 2023**

- **Học phần:** Lập Trình Ứng Dụng Mạng – TN414

- **Lớp học phần:** TN41401

**- Nội dung đáp án:**

**Câu 1:**

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

CheckForIllegalCrossThreadCalls = false;

}

private UdpClient udpLocal = null;

private IPEndPoint ipremote;

Thread getThread = null;

private void openinput(bool state)

{

this.btnGui.Enabled = !state;

this.txtIPR.ReadOnly = !state;

this.txtPortL.ReadOnly = !state;

this.txtPortR.ReadOnly = !state;

this.btnKN.Enabled = state;

}

private void NhanDL()

{

try

{

while (true)

{

byte[] data = new byte[1024];

IPEndPoint ipe = new IPEndPoint(IPAddress.Any, int.Parse(txtPortL.Text.Trim()));

data = udpLocal.Receive(ref ipe);

string s = Encoding.UTF8.GetString(data, 0, data.Length);

if(s.StartsWith("UNS:"))

{

string[] tmp=s.Split(' ');

lstMsg.Items[Int32.Parse(tmp[1])] = "UNSENT MESSAGE [FRIEND]";

}

else

lstMsg.Items.Add("REC: " + s);

}

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString());

}

}

private void btnKN\_Click(object sender, EventArgs e)

{

try

{

udpLocal = new UdpClient(int.Parse(txtPortL.Text.Trim()),AddressFamily.InterNetwork);

ipremote = new IPEndPoint(IPAddress.Parse(txtIPR.Text.Trim()), int.Parse(txtPortR.Text.Trim()));

openinput(false);

getThread = new Thread(new ThreadStart(NhanDL));

getThread.IsBackground = true;

getThread.Start();

}

catch (Exception ex)

{

openinput(true);

MessageBox.Show(ex.ToString());

}

}

private void btnGui\_Click(object sender, EventArgs e)

{

try

{

byte[] data = new byte[1024];

data = Encoding.UTF8.GetBytes(txtMsg.Text);

udpLocal.Send(data,data.Length,ipremote);

lstMsg.Items.Add("SEN: " + txtMsg.Text);

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString());

}

}

private void btnUnsent\_Click(object sender, EventArgs e)

{

int index = lstMsg.SelectedIndex;

string line = lstMsg.Items[index].ToString();

if (line.Substring(0, 4) == "SEN:")

{

byte[] data = new byte[1024];

data = Encoding.UTF8.GetBytes("UNS: "+index.ToString());

udpLocal.Send(data, data.Length, ipremote);

lstMsg.Items[index] = "UNSENT MESSAGE [YOU]";

}

}

}

**Câu 2:**

Server:

public partial class Server : Form

{

TcpListener Listener;

int portN;

String rootDir="F:/";

public Server()

{

InitializeComponent();

}

private delegate void dlgAddInfo(string str);

private void AddInfo(string str)

{

if (this.rtxInfo.InvokeRequired)

{ this.Invoke(new dlgAddInfo(AddInfo), str); }

else

{ this.rtxInfo.AppendText(str+"\n\r"); }

}

private void CreJob(String[] request, StreamWriter sw)

{ //acc key pass message

String s = "100. Tin nhan luu thanh cong";

string fileName = rootDir +request[1]+".txt";

try

{

File.WriteAllText(fileName, request[2]);

}

catch(Exception ex)

{

s = "203. Khong the luu tin nhan.";

}

sw.WriteLine(s);

sw.Flush();

}

private void ReaJob(String[] request, StreamWriter sw)

{

string fileName = rootDir + request[1] + ".txt";

String s = "204. Tin nhan khong ton tai.";

if (File.Exists(fileName))

{

String[] lines=File.ReadAllLines(fileName);

s = String.Join(" ", lines);

}

sw.WriteLine(s);

sw.Flush();

}

private void UndefinedCommand(String[] request, StreamWriter sw)

{

String s = "Lỗi. Lệnh không tồn tại:"+ request[0];

byte[] data = new byte[s.Length];

sw.Write(s, 0, s.Length);

sw.Flush();

}

private void ThreadProc(object obj)

{

try {

var client = (TcpClient)obj;

StreamReader sr = new StreamReader(client.GetStream());

StreamWriter sw = new StreamWriter(client.GetStream());

sw.WriteLine("Chao mung ket noi toi SecretBox");

sw.Flush();

while (true)

{

string raw = sr.ReadLine();

string[] request = raw.Split(' ');

AddInfo("Client Command:"+raw);

string command = "";

if (request.Length != 0)

command = request[0];

switch (command.ToUpper().Trim())

{

case "CRE"://tạo tin nhắn

{

CreJob(request, sw);

break;

}

case "REA"://đọc tin tin

{

ReaJob(request, sw);

break;

}

default:

{

UndefinedCommand(request, sw);

break;

}

}

}

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString());

}

}

public void ListenerThread()

{

Listener = null;

try

{

Listener = new TcpListener(IPAddress.Any, portN);

Listener.Start();

while (true)

{

TcpClient client = null;

NetworkStream netstream = null;

if (Listener.Pending())

{

client = Listener.AcceptTcpClient();

netstream = client.GetStream();

AddInfo("Kết nối với Client.");

ThreadPool.QueueUserWorkItem(ThreadProc, client);

// netstream.Close();

// client.Close();

}

}

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString());

}

}

private void btnStart\_Click(object sender, EventArgs e)

{

Thread thdListener = new Thread(new ThreadStart(ListenerThread));

portN = int.Parse(txtPort.Text);

thdListener.Start();

thdListener.IsBackground=true;

AddInfo("Server đã khởi động");

}

}

Client:

public partial class Client : Form

{

IPEndPoint iep;

TcpClient client;

StreamReader sr;

StreamWriter sw;

public Client()

{

try

{

InitializeComponent();

}

catch (Exception ex)

{

MessageBox.Show(ex.ToString());

}

}

private void AddInfo(String cnt)

{

rtxInfo.AppendText(cnt + "\n\r");

}

private void btnConnect\_Click(object sender, EventArgs e)

{

iep = new IPEndPoint(IPAddress.Parse(txtServerIP.Text), int.Parse(txtServerPort.Text));

client = new TcpClient();

client.Connect(iep);

sr = new StreamReader(client.GetStream());

sw = new StreamWriter(client.GetStream());

AddInfo(sr.ReadLine());

}

private void btnGet\_Click(object sender, EventArgs e)

{

sw.WriteLine("REA " + txtTitle.Text);

sw.Flush();

AddInfo(sr.ReadLine());

}

private void btnSend\_Click(object sender, EventArgs e)

{

sw.WriteLine("CRE " + txtTitle.Text + " "+txtMessage.Text);

sw.Flush();

AddInfo(sr.ReadLine());

}

}

**Duyệt của Trưởng Khoa Giảng viên ra đáp án đề thi**

**Nguyễn Nhị Gia Vinh Phạm Trương Hồng Ngân**