Đề 01

Lập 2 chương trình khác nhau trên ngôn ngữ lập trình tùy chọn C++/Java/C# với yêu cầu sau:

Lập chương trình CT1(có hàm main):

1. (2 điểm) Viết 1hàm gồm 1 vòng lặp tạo liên tục 1 số nguyên không âm (số nguyên 4byte),

nếu số được tạo chia hết cho 1024 thì thoát khỏi vòng lặp, trái lại thi ghi số nguyên vào file dulieu.txt (file mở ở mode ghi đè nếu file đã tồn tại),

khi ghi được thì đóng file.

2. (3 điểm) Đặt hàm vào 1 luồng, kích hoạt luồng trong hàm main().

Lập chương trình CT2 (có hàm main):

3. (2 điểm) Viết 1 hàm có 1 vòng lặp vô hạn đọc số nguyên 4 byte ở file dulieu.txt.

Đọc được số nguyên thì đóng file. Hiển thị giá trị đọc được. Nếu số nguyên đọc được chia hết cho 1024 thì thoát vòng lặp.

4. (3 điểm) Đặt hàm vào 1 luồng, kích hoạt luồng trong hàm main().

Chạy cả CT1 và CT2 đồng thời để thấy hiệu ứng trao đổi dữ liệu.

**TẠO 1 FILE** dulieu.txt

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.IO;

using System.Threading;

namespace bai1decuong

{

class Program

{

public static int message = 0;

public static bool changed = false;

public static void Ghi()

{

try

{

do

{

message =Convert.ToInt32( Console.ReadLine());

StreamWriter sw = new StreamWriter("test.txt");

sw.WriteLine(message);

changed = true;

sw.Close();

} while (message%1024!=0);

}

catch(Exception ex)

{

Console.WriteLine("Error:" + ex.Message);

}

finally

{

Console.WriteLine("Executing finally block");

}

}

public static void Doc()

{

int line;

try

{

while (true)

{

//line = sr.ReadLine();

if (changed == true)

{

StreamReader sr = new StreamReader("test.txt");

line =Convert.ToInt32( sr.ReadLine());

if (message % 1024 != 0)

{

Console.WriteLine("Text da nhap: " + line);

changed = false;

}

else

{

break;

}

sr.Close();

}

Thread.Sleep(10);

}

}

catch (Exception ex)

{

Console.WriteLine("Error:" + ex.Message);

}

//Hien thi bien message ra man hinh

}

static void Main(string[] args)

{

Thread t1 = new Thread(new ThreadStart(Ghi));

Thread t2 = new Thread(new ThreadStart(Doc));

t1.Start();

t2.Start();

}

}

}

Đề 02

Lập 1 chương trình trên ngôn ngữ lập trình tùy chọn C++/Java/C# gồm 2 tác vụ (Task1, Task2) hoạt động ở chế độ không đồng bộ như sau:

Task1:

1. (2 điểm) Viết 1 hàm gồm 1 vòng lặp nhập 1 xâu kí tự string từ bàn phím (string là một biến tổng thể).

Loại bỏ các dấu cách ở đầu và cuối xâu string. Nếu xâu string được nhập và xử lý cắt dấu cách là “exit” thì thoát khỏi vòng lặp.

2. (3 điểm) Thiết lập Task1 bằng cách đặt hàm trên vào 1 timer hoặc 1 luồng, kích hoạt timer/luồng trong hàm main().

Task2:

3. (2 điểm) Viết 1 hàm có 1 vòng lặp vô hạn hiển thị biến xâu kí tự string (xem câu 1). Nếu string là “exit” thì thoát vòng lặp. Bổ sung lệnh gọi hàm trong hàm main().

4. (3 điểm) Thiết lập Task2 bằng cách đặt hàm trên vào 1 timer hoặc 1 luồng, kích hoạt timer/luồng trong hàm main(). Dịch và chạy chương trình.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Threading;

namespace de2trongdecuong

{

class Program

{

public static bool ktra = false;

public static string mess = "";

public static void nhap()

{

do

{

Console.WriteLine("hãy nhap 1 ki tu:");

mess = Console.ReadLine();

ktra = true;

} while (mess.ToLower() != "exit");

}

public static void hamin()

{

while (mess.ToLower() != "exit")

{

if (ktra == true)

{

Console.WriteLine("cac ki tu vua nhap vao chuoi");

Console.WriteLine(mess);

ktra = false;

}

}

}

static void Main(string[] args)

{

Thread t1 = new Thread(new ThreadStart(nhap));

Thread t2 = new Thread(new ThreadStart(hamin));

t1.Start();

t2.Start();

}

}

}

Đề 03

Lập 2 chương trình khác nhau trên ngôn ngữ lập trình tùy chọn C++/Java/C# gồm CT1(client), CT2 (server)

sử dụng Socket API, giao thức TCP/IP truyền nhận dữ liệu tại IP="127.0.0.1" (localhost), port=100 và hoạt động như sau:

CT1 (có hàm main):

1. (2 điểm) Viết 1 hàm gồm 1 vòng lặp tạo ra sô nguyên ngẫu nhiên n,

vòng lặp dừng khi số nguyên sinh ra lớn hơn 20000 và chia hết cho 1024.

Viết hàm main() gọi hàm.

2. (3 điểm) Bổ sung lệnh truyền giá trị số nguyên sinh ra cho CT2 trong vòng lặp.

Dịch chương trình CT1.

CT2 (có hàm main):

3. (4 điểm) Viết 1 hàm có 1 vòng lặp vô hạn nhận giá trị biến nguyên được gửi đến từ CT1.

Kiểm tra, nếu CT2 nhận được giá trị nguyên k thì:

Trường hợp 1: Nếu n > 20000 và chia hết cho 1024 thì thoát khỏi vòng lặp.

Trường hợp 2: Trái lại, hiển thị giá trị lớn nhất được CT1 sinh ra tính tại thời điểm đang chạy (giá trị số lớn nhất có thể luôn thay đổi khi CT1 đang chạy).

Viết hàm main(), dịch chương trình CT2. Rồi chạy CT2 và chạy CT1 để xem hiệu ứng nhập mảng ở CT1, hiển thị mảng ở CT2.

4. (1 điểm)

Đặt hàm ở câu 3 vào 1 timer, bổ sung lệnh kích hoạt timer trong hàm main(). Dịch chương trình.

Biên dịch và chạy lần sau cùng:

+ Đóng tất cả CT1, CT2 đang chạy.

+ Chạy chương trình CT2 sau đó chạy CT1 để xem hiệu ứng kết nối, truyền và nhận dữ liệu.

**CT1:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net;

using System.Net.Sockets;

using System.Text;

using System.Threading;

using System.Threading.Tasks;

namespace client

{

internal class Program

{

static void Main(string[] args)

{

try

{

IPEndPoint ip = new IPEndPoint(IPAddress.Parse("127.0.0.1"), 8888);

Socket client = new Socket(SocketType.Stream, ProtocolType.Tcp);

client.Connect(ip);

while (true)

{

Random random = new Random();

int x = random.Next(0, 20481);

Console.WriteLine("Gia tri gui la: " + x);

if(x>20000 && x % 1024 == 0)

{

byte[] gui = ASCIIEncoding.ASCII.GetBytes(x.ToString());

client.Send(gui);

break;

}

else

{

byte[] gui = ASCIIEncoding.ASCII.GetBytes(x.ToString());

client.Send(gui);

}

Thread.Sleep(1000);

}

client.Close();

}

catch { }

}

}

}

**CT2:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net;

using System.Net.Sockets;

using System.Text;

using System.Threading.Tasks;

namespace server

{

internal class Program

{

static void Main(string[] args)

{

try

{

IPEndPoint ip = new IPEndPoint(IPAddress.Parse("127.0.0.1"),8888);

Socket server = new Socket(SocketType.Stream, ProtocolType.Tcp);

server.Bind(ip);

server.Listen(100);

int ktra = 0;

int nhan = 0;

bool ktral = true;

Socket client = server.Accept();

do

{

byte[] bnhan = new byte[255];

client.Receive(bnhan);

nhan = int.Parse(ASCIIEncoding.ASCII.GetString(bnhan).Trim().Replace("\n", ""));

if(nhan > 20000 && nhan % 1024 == 0)

{

ktral = false;

}

else

{

Console.WriteLine("gia tri duoc nhan la: ");

if(nhan >= ktra)

{

Console.WriteLine(nhan);

ktra = nhan;

}

else

{

Console.WriteLine(ktra);

}

}

} while (ktral == true);

server.Close();

}

catch { }

}

}

}

Đề 04

Lập 2 chương trình gồm CT1(client) và CT2 (server), sử dụng giao thức TCPIP cho các hàm SocketAPI với tham số:

socket ip="127.0.0.1", port=2014 cho CT2(server),

+ CT1 có 1 vòng lặp vô hạn nhập 1 mảng số nguyên gồm n phần tử như sau:

Nhập n, nếu n <=0 thì gửi sang CT2 giá trị n bằng socketAPI rồi thoát khỏi vòng lặp.

Nhập tiếp n phần tử của mảng số nguyên, rồi gửi sang CT2 mảng số nguyên bằng socketAPI.

+ CT2. có vòng lặp, chờ kết nối CT1, nhận giá trị n

Nếu n<=0 thì thoát khỏi vòng lặp.

Nhận mảng n phần tử số nguyên. Hiển thị các giá trị nhận được.

Dich CT2 và CT1. Sau đó chạy CT2 rồi chạy CT1.

**CT1:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Net;

using System.Net.Sockets;

namespace client

{

class Program

{

static void Main(string[] args)

{

try

{

IPEndPoint ip = new IPEndPoint(IPAddress.Parse("127.0.0.1"), 8888);

Socket client = new Socket(SocketType.Stream, ProtocolType.Tcp);

client.Connect(ip);

int n;

do

{

Console.WriteLine("nhap so luong phan tu");

n = Convert.ToInt32(Console.ReadLine());

byte[] bguin = ASCIIEncoding.ASCII.GetBytes(n.ToString());

client.Send(bguin);

int[] A = new int[n];

for (int i = 0; i < n; i++)

{

A[i] = Convert.ToInt32(Console.ReadLine());

}

for (int i = 0; i < n; i++)

{

byte[] bgui = ASCIIEncoding.ASCII.GetBytes(A[i].ToString());

client.Send(bgui);

}

} while (n > 0);

client.Close();

}

catch (Exception e)

{

Console.WriteLine(e.Message);

}

Console.ReadKey();

}

}

}

**CT2:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Net;

using System.Net.Sockets;

namespace server

{

class Program

{

static void Main(string[] args)

{

try

{

IPEndPoint ip = new IPEndPoint(IPAddress.Parse("127.0.0.1"), 8888);

Socket server = new Socket(SocketType.Stream, ProtocolType.Tcp);

server.Bind(ip);

Console.WriteLine("cho ket noi");

server.Listen(100);

Socket client = server.Accept();

bool ktra=true;

do

{

byte[] bnhan = new byte[225];

client.Receive(bnhan);

int n = Convert.ToInt32(ASCIIEncoding.ASCII.GetString(bnhan));

int[] A = new int[n];

if (n > 0)

{

for (int i = 0; i < n; i++)

{

byte[] bnhanpt = new byte[225];

client.Receive(bnhanpt);

A[i] = Convert.ToInt32(ASCIIEncoding.ASCII.GetString(bnhanpt));

}

Console.WriteLine("mang da nhan");

for (int i = 0; i < n; i++)

{

Console.Write(A[i] + "\t");

}

}

else

{

ktra = false;

}

} while (ktra == true);

client.Close();

server.Close();

}

catch (Exception e)

{

Console.WriteLine(e.Message);

}

Console.ReadKey();

}

}

}

Đề 05:

Lập trình trên ngôn ngữ tùy chọn với yêu cầu sau:

Viết CT1.

a (2đ). Viết 1 hàm gồm 1 vòng lặp nhập 1 kí tự từ bàn phím, ghi kí tự lên 1 file dulieu.txt. Nếu kí tự là '!' thì thoát khỏi vòng lặp.

b. (4đ) Đặt hàm vào 1 luồng, dịch chương trình CT1.

Viết chương trình 2.

c (2đ). Viết 1 hàm có 1 vòng lặp vô hạn đọc kí tự ở file dulieu.txt. Đọc được kí tự thì đóng file. Hiển thị kí tự. Nếu kí tự đọc được là '!' thì thoát vòng lặp.

d. (4đ) Đặt hàm vào 1 luồng, dịch chương trình CT2.

Chạy CT1 và CT2 để thấy hiệu ứng trao đổi dữ liệu.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.IO;

using System.Threading;

namespace bai5decuong

{

class Program

{

//khai báo 1 biến để nhập kí tự

public static string message = "";

//tao cờ để khi nhập mới ghi ra( có thể không cần)

public static bool changed = false;

public static void Ghi()

{

// khi dùng file cần đến try catch

try

{

do

{

//nhập vào biến message do bài 5 yêu cầu nhập kí tự

message = Console.ReadLine();

// mở file

StreamWriter sw = new StreamWriter("test.txt");

sw.WriteLine(message);

changed = true;

sw.Close();

} while (message.ToLower() != "!");

}

catch

{

}

finally

{

Console.WriteLine("Executing finally block");

}

}

public static void Doc()

{

string line;

try

{

while (true)

{

//line = sr.ReadLine();

if (changed == true)

{

StreamReader sr = new StreamReader("D:\\decuonghethong\\bai5decuong\\test.txt");

line = sr.ReadLine();

Console.WriteLine(line);

if (line.ToLower() != "!")

{

Console.WriteLine("Text da nhap: " + line);

changed = false;

}

else

{

break;

}

sr.Close();

}

Thread.Sleep(10);

}

}

catch (Exception ex)

{

Console.WriteLine("Error:" + ex.Message);

}

//Hien thi bien message ra man hinh

}

static void Main(string[] args)

{

Thread t1 = new Thread(new ThreadStart(Ghi));

Thread t2 = new Thread(new ThreadStart(Doc));

t1.Start();

t2.Start();

}

}

}

Đề 06:

Lập trình trên ngôn ngữ tùy chọn (C/C#/Java) với yêu cầu sau:

Viết chương trình gồm 2 tác vụ hoạt độngkhông đồng bộ như sau:

Task1:

a (2đ). Viết 1 hàm gồm 1 vòng lặp nhập 1 kí tự từ bàn phím, ghi kí tự lên 1 file dulieu.txt. Nếu kí tự là '!' thì thoát khỏi vòng lặp.

b. (4đ) Đặt hàm vào 1 timer cứ 5ms (1ms=1/1000 giây) thực hiện 1 lần.

Task2

c (2đ). Viết 1 hàm có 1 vòng lặp vô hạn đọc kí tự ở file dulieu.txt. Đọc được kí tự thì đóng file. Hiển thị kí tự. Nếu kí tự đọc được là '!' thì thoát vòng lặp.

d. (4đ) Đặt hàm vào 1 timer cứ 10ms thực hiện 1 lần..

Ghi chú: có thể sử dụng timer dựa trên thread để cứ 1 lần chu kỳ lại "nhả" CPU cho hệ điều hành hoặc sử dụng timer được hỗ trợ sẵn bởi môi trường lập trình.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.IO;

using System.Threading;

namespace bai6decuong

{

class Program

{

public static string message = "";

public static bool changed = false;

public static void Ghi()

{

try

{

do

{

message = Console.ReadLine();

StreamWriter sw = new StreamWriter("D:\\decuonghethong\\bai6decuong\\test.txt");

sw.WriteLine(message);

changed = true;

sw.Close();

Thread.Sleep(5);

} while (message.ToLower() != "!");

}

catch

{

}

finally

{

Console.WriteLine("Executing finally block");

}

}

public static void Doc()

{

string line;

try

{

while (true)

{

if (changed == true)

{

StreamReader sr = new StreamReader("D:\\decuonghethong\\bai5decuong\\test.txt");

line = sr.ReadLine();

Console.WriteLine(line);

if (line.ToLower() != "!")

{

Console.WriteLine("Text da nhap: " + line);

changed = false;

}

else

{

break;

}

sr.Close();

}

Thread.Sleep(10);

}

}

catch (Exception ex)

{

Console.WriteLine("Error:" + ex.Message);

}

}

static void Main(string[] args)

{

Thread t1 = new Thread(new ThreadStart(Ghi));

Thread t2 = new Thread(new ThreadStart(Doc));

t1.Start();

t2.Start();

}

}

}

Đề 07:

Lập trình trên ngôn ngữ tùy chọn (C/C#/Java) với yêu cầu sau:

Viết chương trình gồm 2 timer hoạt động như sau:

timer1:

a (2đ). Viết 1 hàm gồm 1 vòng lặp nhập 1 kí tự từ bàn phím, lưu vào 1 biến tổng thể c, hiển thị kí tự. Nếu kí tự là '!' thì thoát khỏi vòng lặp.

b. (4đ) Đặt hàm vào 1 timer cứ 7ms (1ms=1/1000 giây) thực hiện 1 lần.

timer2

c (2đ). Viết 1 hàm có 1 vòng lặp phát ra tiếng beep, nếu biến tổng thể c là '!' thì thoát vòng lặp.

d. (4đ) Đặt hàm vào 1 timer cứ 10ms thực hiện 1 lần..

Chạy chương trình để xem hiệu ứng timer1, timer2.

Ghi chú: có thể sử dụng timer dựa trên thread để cứ 1 lần chu kỳ lại "nhả" CPU cho hệ điều hành hoặc sử dụng timer được hỗ trợ sẵn bởi môi trường lập trình.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.IO;

using System.Threading;

namespace bai7decuong

{

class Program

{

public static bool ktra = false;

public static string mess = "";

public static void nhap()

{

do

{

Console.WriteLine("hãy nhap 1 ki tu:");

mess = Console.ReadLine();

ktra = true;

Thread.Sleep(7);

} while (mess.ToLower() != "!");

}

public static void hamin()

{

while (mess.ToLower() != "!")

{

if (ktra == true)

{

Console.Beep();

ktra = false;

}

Thread.Sleep(10);

}

}

static void Main(string[] args)

{

//câu1,2,3,4

// nhap();

// hamin();

Thread t1 = new Thread(new ThreadStart(nhap));

Thread t2 = new Thread(new ThreadStart(hamin));

t1.Start();

t2.Start();

}

}

}

Đề 08:

Lập trình trên ngôn ngữ tùy chọn (C/C#/Java) với yêu cầu sau:

Viết 2 chương trình gồm CT1(client) , CT2 (server) sử dụng Socket API, giao thức TCPIP truyền nhận dữ liệu tại IP="127.0.0.1", port=100, hoạt động như sau:

CT1:

a(2đ). Viết 1 hàm gồm 1 vòng lặp vô hạn nhập 1 kí tự từ bàn phím, khi nhập phím ESC thì thoát khỏi vòng lặp. Viết hàm main() gọi hàm, dịch và chạy chương trình.

b(2đ).Bổ sung lệnh truyền byte kí tự đến cho CT2 trong vòng lặp.

c(1đ) Đặt hàm vào 1 thread sao cho CT1 không chiếm 100% CPU khi chạy.

Dịch và chạy thử CT1 (chưa có hiệu ứng truyền dữ liệu)

CT2:

d (4đ). Viết 1 hàm gồm 1 vòng lặp vô hạn nhận 1 byte kí tự. Nếu nhận được kí tự thì:

TH1: Nếu kí tự là phím ESC thì thoát khỏi vòng lặp.

TH2: Nếu kí tự không là ESC thì hiển thị kí tự.

Viêt hàm main() gọi hàm.

e(1đ). Đặt hàm vào 1 thread sao cho CT2 không chiếm 100% CPU khi chạy.

Biên dịch lần cuối:

+ Đóng tất cả CT1,CT2 đang chạy.

+ Chạy chương trình CT2 sau đó chạy CT1 để xem hiệu ứng kết nối, truyền và nhận dữ liệu.

+ Chạy Task manager (nếu chạy trên hệ điều hành Windows) để xem CT1 và CT2 khi chạy chiếm bao nhiêu % CPU.

**CT1:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Net;

using System.Net.Sockets;

namespace client

{

class Program

{

static void Main(string[] args)

{

try

{

IPEndPoint ip = new IPEndPoint(IPAddress.Parse("127.0.0.1"), 8888);

Socket client = new Socket(SocketType.Stream, ProtocolType.Tcp);

client.Connect(ip);

while (true)

{

ConsoleKeyInfo keyInfo;

keyInfo = Console.ReadKey(true);

if (keyInfo.Key == ConsoleKey.Escape)

{

Console.WriteLine(keyInfo.KeyChar.ToString());

break;

}

else

{

byte[] bguik = ASCIIEncoding.ASCII.GetBytes(keyInfo.KeyChar.ToString());

client.Send(bguik);

}

}

client.Close();

}

catch (Exception e)

{

Console.WriteLine(e.Message);

}

Console.ReadKey();

}

}

}

**CT2:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Net;

using System.Net.Sockets;

namespace server

{

class Program

{

static void Main(string[] args)

{

try

{

IPEndPoint ip = new IPEndPoint(IPAddress.Parse("127.0.0.1"), 8888);

Socket server = new Socket(SocketType.Stream, ProtocolType.Tcp);

server.Bind(ip);

server.Listen(100);

string nhan ;

bool ktra1 = true;

Socket client = server.Accept();

do

{

byte[] bnhan = new byte[1];

client.Receive(bnhan);

nhan = ASCIIEncoding.ASCII.GetString(bnhan).Trim().Replace("\n","");

if (nhan== "\u0000")

{

ktra1 = false;

}

else

{

Console.WriteLine("gia tri dc nhan la: ");

Console.WriteLine(nhan);

}

} while (ktra1 == true);

client.Close();

server.Close();

}

catch (Exception e)

{

Console.WriteLine(e.Message);

}

Console.ReadKey();

}

}

}

Đề 09:

Lập 2 chương trình trên ngôn ngữ lập trình tùy chọn C++/Java/C#/Python

gồm CT1(client), CT2 (server) sử dụng Socket API, giao thức TCPIP truyền nhận dữ liệu tại IP="127.0.0.1"(localhost), port=100 và hoạt động như sau:

CT1:

1. Viết 1 hàm gồm 1 vòng lặp nhập vào từ bàn phím mảng số thực 64bit (nhập số phần tử n>=2, sau đó nhập từng phần tử mảng), nếu nhập n thỏa n < 2 thì thoát khỏi vòng lặp.

2. Bổ sung lệnh truyền giá trị n và truyền biến mảng đến cho CT2 trong vòng lặp.

CT2:

3. Viết 1 hàm có 1 vòng lặp vô hạn nhận giá trị biến nguyên n được gửi đến từ CT1, Nếu nhận được n thì nếu n < 2 thì thoát khỏi vòng lặp.Trái lại, nhận tiếp mảng n số thực 64bit được gửi đến từ CT1 và

hiển thị 2 giá trị đầu tiên xếp theo thứ tự từ cao xuống thấp của mảng sau khi nhận được (2 giá trị đầu tiên này có thể luôn thay đổi khi CT1 đang chạy).

Dịch và chạy CT2 rồi CT1 để kiểm tra.

**CT1:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Net;

using System.Net.Sockets;

using System.Threading;

namespace client

{

class Program

{

static void Main(string[] args)

{

try

{

IPEndPoint ip = new IPEndPoint(IPAddress.Parse("127.0.0.1"), 8888);

Socket client = new Socket(SocketType.Stream, ProtocolType.Tcp);

client.Connect(ip);

int n;

do

{

Console.WriteLine("nhap so luong phan tu");

n = Convert.ToInt32(Console.ReadLine());

byte[] bguin = ASCIIEncoding.ASCII.GetBytes(n.ToString());

client.Send(bguin);

double[] a = new double[n];

for(int i = 0; i < n; i++)

{

a[i]= Convert.ToDouble(Console.ReadLine());

}

for (int i = 0; i < n; i++)

{

byte[] bgui = ASCIIEncoding.ASCII.GetBytes(a[i].ToString());

client.Send(bgui);

Thread.Sleep(10);

}

} while (n >= 2);

client.Close();

}

catch (Exception e)

{

Console.WriteLine(e.Message);

}

Console.ReadKey();

}

}

}

**CT2:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Net;

using System.Net.Sockets;

namespace server

{

class Program

{

static void Main(string[] args)

{//check lại

try

{

IPEndPoint ip = new IPEndPoint(IPAddress.Parse("127.0.0.1"), 8888);

Socket server = new Socket(SocketType.Stream, ProtocolType.Tcp);

server.Bind(ip);

server.Listen(100);

bool ktra1 = true;

Socket client = server.Accept();

do

{

byte[] bnhan = new byte[225];

client.Receive(bnhan);

int n = Convert.ToInt32(ASCIIEncoding.ASCII.GetString(bnhan));

if (n >= 2)

{

double[] mang = new double[n];

for (int i = 0; i <n; i++)

{

byte[] bnhanpt = new byte[225];

client.Receive(bnhanpt);

double a = Convert.ToDouble(ASCIIEncoding.ASCII.GetString(bnhanpt));

mang[i] = a;

}

Console.WriteLine("mang da nhap");

for (int i = 0; i < n; i++)

{

Console.WriteLine(mang[i]);

}

}

else

{

ktra1= false;

}

} while (ktra1 == true);

server.Close();

}

catch (Exception e)

{

Console.WriteLine(e.Message);

}

Console.ReadKey();

}

}

}

Đề 10: <bai8>

Lập 2 chương trình gồm CT1(client) và CT2 (server),

CT1 có 1 vòng lặp thực hiện gửi file ảnh 'I04.JPG; qua socket ip="127.0.0.1", port=2014

CT2(server), khi nhận đc file thì thoát khỏi vòng lặp

**CT1:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Net;

using System.Net.Sockets;

using System.IO;

namespace client

{

class Program

{

static void Main(string[] args)

{

try

{

IPEndPoint ip = new IPEndPoint(IPAddress.Parse("127.0.0.1"), 8888);

Socket client = new Socket(SocketType.Stream, ProtocolType.Tcp);

client.Connect(ip);

byte[] imageBytes = File.ReadAllBytes("D:\\decuonghethong\\bai10decuong\\client\\I04.jpg");

client.Send(imageBytes);

client.Close();

}

catch (Exception e)

{

Console.WriteLine(e.Message);

}

Console.ReadKey();

}

}

}

**CT2:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Net;

using System.Net.Sockets;

namespace server

{

class Program

{

static void Main(string[] args)

{//check lại

try

{

IPEndPoint ip = new IPEndPoint(IPAddress.Parse("127.0.0.1"), 8888);

Socket server = new Socket(SocketType.Stream, ProtocolType.Tcp);

server.Bind(ip);

server.Listen(100);

Socket client = server.Accept();

byte[] bnhan = new byte[225];

client.Receive(bnhan);

if (bnhan != null)

{

Console.WriteLine("nhan thanh cong");

}

client.Close();

server.Close();

}

catch (Exception e)

{

Console.WriteLine(e.Message);

}

Console.ReadKey();

}

}

}

Đề 12:

Viết chương trình gồm 2 luồng, mỗi luồng có 1 vòng lặp.

+ Luồng 1 thực hiện nhập phím từ bàn phím, nếu người dùng nhấn phím ESCAPE thì thoát khỏi vòng lặp,

trái lại thì nạp kí tự vào 1 hàng đợi (queue).

+ Luồng 2, hiển thị kí tự trong hàng đợi bàn phím theo thứ tự lưu trữ vào hàng đợi.

+ Khi luồng 1 dừng thì luồng 2 cũng dừng.

+ Viết hàm main() khởi tạo và kích hoạt các luồng trên. Dịch và chạy chương trình.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Threading;

namespace bai12decuong

{

class Program

{

public static ConsoleKeyInfo keyInfo;

public static Queue<string> q = new Queue<string>();

public static bool c = false;

public static bool ktradung= false;

public static void nhap()

{

do

{

Console.WriteLine("hay nhap 1 ki tu:");

keyInfo = Console.ReadKey(true);

q.Enqueue(keyInfo.KeyChar.ToString());

c = true;

} while (keyInfo.Key != ConsoleKey.Escape);

}

public static void doc()

{

while (keyInfo.Key != ConsoleKey.Escape)

{

if (c == true)

{

Console.WriteLine("ki tu da nhap vao hang doi");

foreach(var item in q)

{

Console.WriteLine(item);

}

c = false;

}

}

}

static void Main(string[] args)

{

Thread t1 = new Thread(new ThreadStart(nhap));

Thread t2 = new Thread(new ThreadStart(doc));

t1.Start();

t2.Start();

}

}

}

Đề 14:

Lập 1 chương trình trên ngôn ngữ lập trình tùy chọn C++/Java/C# gồm 2 tác vụ (Task1, Task2) hoạt động ở chế độ không đồng bộ như sau:

Task1:

a. Viết 1 hàm gồm 1 vòng lặp sinh số nguyên ngẫu nhiên cho đến khi gặp số chia hết cho 10011 thì dừng lại.

Task2:

b. Viết 1 hàm có 1 vòng lặp chỉ dừng lặp khi vòng lặp của Task1 dừng lại. Vòng lặp này luôn hiển thị số chia hết cho 3 mà Task1 tạo ra tại mọi thời điểm chạy.

Bổ sung lệnh gọi hàm trong hàm main(), dịch và chạy lại chương trình.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading;

using System.Threading.Tasks;

namespace bai14decuong

{

class Program

{

public static int x ;

public static void Ghi()

{

do

{

Random r = new Random();

x = r.Next();

Thread.Sleep(10);

} while (x % 1000 != 0);

}

public static void Doc()

{

while(x % 1000 != 0)

{

if (x % 3 == 0)

{

Console.WriteLine("so ngau nhien " + x);

Thread.Sleep(5);

}

}

}

static void Main(string[] args)

{

Thread t1 = new Thread(new ThreadStart(Ghi));

Thread t2 = new Thread(new ThreadStart(Doc));

t1.Start();

t2.Start();

Console.ReadKey();

}

}

}

Đề 15:

Viết chương trình tạo 2 luồng Thr1 và Thr2, mỗi luồng gồm 1 vòng lặp, sao cho:

+ Trong hàm main() khai báo 2 biến mảng a và b của các số nguyên (mảng không quá 10 phần tử)

+ Thr1 có vùng nhớ riêng là vùng nhớ cho a.

+ Thr2 có vùng nhớ riêng là vùng nhớ cho b.

Khi Thr1 hoạt động thì người dùng nhập số phần tử của mảng a, và Thr2 sinh ra số n ngẫu nhiên và tiếp tục n phần tử của mảng b.

Khi nhập số phần tử không vượt quá 0 thì Thread tương ứng dừng.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Threading;

namespace bai15decuong

{

class Program

{

public static bool c = true;

public static int slpt1, slpt2;

public static Random r = new Random();

public static void task1(object obj)

{

int[] a = (int[])obj;

for(int n = 0; n < a.Length; n++)

{

Console.WriteLine("nhap phan tu thu "+n);

int pt;

var ptmang = int.TryParse(Console.ReadLine(), out pt );

a[n] = pt;

}

Console.WriteLine("mang o task 1 la");

for (int n = 0; n < a.Length; n++)

{

Console.WriteLine(a[n]);

}

c = false;

Console.ReadKey();

}

public static void task2(object obj)

{

int[] a = (int[])obj;

if (c == false)

{

for (int n = 0; n < a.Length; n++)

{

a[n] = r.Next();

}

Console.WriteLine("mang o task 2 la");

for (int n = 0; n < a.Length; n++)

{

Console.WriteLine(a[n]);

}

}

Console.ReadKey();

}

static void Main(string[] args)

{

do

{

Console.WriteLine("nhap so luong phan tu cho mang 1:");

slpt1 = Convert.ToInt32(Console.ReadLine().ToLower().Replace("\n"," "));

} while (slpt1 > 10 );

do

{

slpt2 = r.Next();

} while ( slpt2 > 10);

int[] m1 = new int[slpt1];

int[] m2 = new int[slpt2];

Thread t1 = new Thread(new ParameterizedThreadStart(task1));

Thread t2 = new Thread(new ParameterizedThreadStart(task2));

t1.Start(m1);

t2.Start(m2);

Console.ReadKey();

}

}

}

Đề 18: Lập 1 chương trình trên ngôn ngữ lập trình tùy chọn C++/Java/C# gồm 2 tác vụ (Task1, Task2)

hoạt động ở chế độ không đồng bộ như sau:

Task1:

1. Viết 1 hàm gồm 1 vòng lặp nhập liên tục các số nguyên từ bàn phím, lưu các số

nhập được vào một biến cấu trúc A kiểu bộ đệm vòng, và vòng lặp dùng lại khi

nhập số nguyên < 0.

2. Thiết lập Task1 bằng cách đặt hàm trên vào 1 luồng, kích hoạt luồng trong main().

Task2:

3. Viết 1 hàm vó 1 vòng lặp chỉ dùng lại khi vòng lặp của task1 dừng lại.

Vòng lặp này luôn hiển thị các số nguyên được lưu trong biến A.

Bổ sung lệnh gọi hàm trong hàm main().

4. Thiết lập Task2 bằng cách đặt hàm trên vào 1 luồng, kích hoạt luồng trong main().

Hiển thị trung bình cộng các phần tử của mảng. Dịch và chạy chương trình.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Threading;

namespace bai18decuong

{

class Program

{

public static Queue<int> c = new Queue<int>();

public static bool ktra = false;

public static int mess=0 ;

public static void nhap()

{

while (ktra == false)

{

do

{

Console.WriteLine("hãy nhap 1 so:");

var a = int.TryParse(Console.ReadLine(), out mess);

if (a)

{

c.Enqueue(mess);

}

ktra = true;

Thread.Sleep(10);

} while (mess >= 0);

}

}

public static void hamin()

{

while (mess>=0)

{

if (ktra == true)

{

Console.WriteLine("cac so vua nhap vao chuoi");

foreach (var item in c)

{

Console.WriteLine(item);

}

ktra = false;

}

Thread.Sleep(10);

}

}

static void Main(string[] args)

{

Thread t1 = new Thread(new ThreadStart(nhap));

Thread t2 = new Thread(new ThreadStart(hamin));

t1.Start();

t2.Start();

}

}

}

Đề 513:

Lập 1 chương trình trên ngôn ngữ lập trình tùy chọn C++/Java/C# gồm 2 tác vụ (Task1, Task2) hoạt động ở chế độ không đồng bộ như sau:

Taskl: 1. (2 điểm) Viết 1 hàm gồm 1 vòng lặp sinh số nguyên ngẫu nhiên 32bit cho đến khi gặp số lớn hơn 1042021 thì dừng vòng lặp. Viết hàm main() gọi hàm, dịch và chạy chương trình.

2. (3 điểm) Thiết lập Task1 bằng cách đặt hàm trên vào 1 timer chu kỳ 5 mili giây hoặc 1 luồng, kích hoạt timer/luồng trong hàm main(). Dịch và chạy lại chương trình.

Task2:

3. (2 điểm) Viết 1 hàm có 1 vòng lặp chỉ dừng lặp khi vòng lặp của Task1 dừng lại. Vòng lặp này luôn hiển thị giả trị nhỏ nhất được Task 1 tạo ra tại mọi thời điểm chạy. Bổ sung lệnh gọi hàm trong hàm main(), dịch và chạy lại chương trình.

4. (3 điểm) Thiết lập Task2 bằng cách đặt hàm trên vào 1 timer hoặc 1 luồng, kích hoạt luồng trong hàm main().

Thêm lệnh phát ra tiếng Beẹp khi hiển thị giá trị số lớn nhất trên. Dịch và chạy lại chương trình.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Threading;

using System.IO;

namespace de513

{

class Program

{

public static int a = 0;

// luôn khai báo giá trị của ktra là false( tác dụng là để khi ta nhập giá trị ở luồng 1 thì luồng 2 mới in)

public static bool c = false;

public static void task1()

{

do

{

Random r = new Random();

a = r.Next();

Console.WriteLine(a);

// đặt lại giá trị cho ktra(c) thành true

c = true;

Thread.Sleep(5);

} while (a <= 1042021);

}

public static void task2()

{

int min = 1042022;

while (a <= 1042021)

{

// luôn đặt hàm kiểm tra giá trị của biến ktra xem có bằng true hay ko sau while

if (c == true)

{

if (min > a)

{

min = a;

}

Console.WriteLine("so be nhat la: " + min);

Console.Beep();

}

Thread.Sleep(5);

}

}

static void Main(string[] args)

{

// câu1 câu3

// task1();

// task2();

Thread t1 = new Thread(new ThreadStart(task1));

Thread t2 = new Thread(new ThreadStart(task2));

t1.Start();

t2.Start();

Console.ReadKey();

}

}

}

Đề 516:

Lập 1 chương trình trên ngôn ngữ lập trình tùy chọn C++/Java/C# gồm 2 tác vụ (Task1, Task2) hoạt động ở chế độ không đồng bộ như sau:

Task1:

1. (2 điểm) Viết 1 hàm gồm 1 vòng lặp nhập liên tục các kí tự từ bàn phím, lưu các kí tự nhập vào một biến tổng thể C có cấu trúc dữ liệu kiểu bộ đệm vòng, và vòng lặp dừng lại khi nhập kí tự '#'. Viết hàm main() gọi hàm, dịch và chạy chương trình.

2. (3 điểm) Thiết lập Task1 bằng cách đặt hàm trên vào 1 luồng, kích hoạt luồng trong hàm main(). Dịch và chạy lại chương trình.

Task2:

3. (2 điểm) Viết 1 hàm có 1 vòng lặp chỉ dừng lặp khi vòng lặp của Taski dừng lại. Vòng lặp này luôn hiền thị kí tự theo thứ tự nhập được lưu trong biển tổng thể C của Task1. Bổ sung lệnh gọi hàm trong hàm main(), dịch và chạy lại chương trình.

4. (2 điểm) Thiết lập Task2 bằng cách đặt hàm trên vào 1 luồng, kích hoạt luồng trong hàm main(). Dịch và chạy lại chương trình.

5. (1 điểm) Thay đổi các hàm ở câu 1 và câu 3 (tạo lập Taski và Task2) để vùng nhớ của biển C trở thành vùng nhớ riêng mà các luồng này trẻ tới. Dịch và chạy lại chương trình.

Chú ý: Có thể sử dụng cấu trúc dữ liệu tương đương để thay thế kiểu cấu trúc bộ đệm vòng

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Threading;

namespace de516

{

class Program

{

public static Queue<string> c = new Queue<string>();

public static bool ktra = false;

public static string mess = "";

public static void nhap()

{

do

{

Console.WriteLine("hãy nhap 1 ki tu:");

mess = Console.ReadLine();

ktra = true;

c.Enqueue(mess);

} while (mess.ToLower() != "#");

}

public static void hamin()

{

while (mess.ToLower() != "#")

{

if (ktra == true)

{

Console.WriteLine("cac ki tu vua nhap vao chuoi");

foreach (var item in c)

{

Console.WriteLine(item);

}

ktra = false;

}

}

}

public static void nhap5(object obj)

{

Queue<string> c = (Queue<string>)obj;

do

{

Console.WriteLine("hãy nhap 1 ki tu:");

mess = Console.ReadLine();

ktra = true;

c.Enqueue(mess);

} while (mess.ToLower() != "#");

}

public static void hamin5(object obj)

{

Queue<string> c = (Queue<string>)obj;

while (mess.ToLower() != "#")

{

if (ktra == true)

{

Console.WriteLine("cac ki tu vua nhap vao chuoi");

foreach (var item in c)

{

Console.WriteLine(item);

}

ktra = false;

}

}

}

static void Main(string[] args)

{

Queue<string> c = new Queue<string>();

Thread t1 = new Thread(new ParameterizedThreadStart(nhap5));

Thread t2 = new Thread(new ParameterizedThreadStart(hamin5));

t1.Start(c);

t2.Start(c);

}

}

}