**BÀI TẬP HỆ ĐIỀU HÀNH**

**1.**      **Cài đặt, cấu hình, sử dụng hệ điều hành, phần mềm**

**1.1.**  **Cài đặt phần mềm máy ảo**

**1.1.1.**     **Cài đặt phần mềm Hyper-V**

**1.1.2.**     **Cài đặt phần mềm Vmware**

**1.2.**  **Cài đặt hệ điều hành**

**1.2.1.**     **Thiết lập Bios Setup**

**1.2.2.**     **Phân vùng và định dạng ổ đĩa**

**1.2.3.**     **Cài đặt hệ điều hành Windows**

**1.2.4.**     **Cài đặt hệ điều hành mã nguồn mở (Ubuntu, CenOS,…)**

**1.2.5.**     **Cài đặt hệ điều hành Android**

**1.3.**  **Sao lưu và phục hồi máy tính**

**1.3.1.**     **Tạo USB Boot dùng để cứu hộ máy tính (DLC Boot)**

**1.3.2.**     **Sử dụng phần mềm Ghost để sao lưu hệ điều hành**

**1.3.3.**     **Sử dụng phần mềm Ghost để phục hồi hệ điều hành**

**1.4.**  **Cài đặt và sử dụng một số phần mềm thường dùng**

**1.4.1.**     **Microsoft Office**

**1.4.2.**     **Microsoft Visual Studio**

**1.4.3.**     **Microsoft SQL Server**

**1.4.4.**     **Các phần mềm bảo vệ máy tính**

**1.4.5.**     **Các phần mềm tiện ích**

**1.4.6.**     **Các phần mềm phục hồi dữ liệu**

**1.5.**  **Bảo trì và tối ưu hệ điều hành**

**1.5.1.**     **Làm sạch ổ đĩa (Disk Cleanup)**

**1.5.2.**     **Chống phân mảnh ổ đĩa (Disk Defragment)**

**1.5.3.**     **Kiểm tra lỗi ổ đĩa (Check Disk)**

**1.5.4.**     **Thiết lập các chương trình khởi động (Startup)**

**1.5.5.**     **Cấu hình các dịch vụ (Service)**

**1.5.6.**     **Tắt, bật các tính năng (Feature)**

**1.5.7.**     **Cấu hình tường lửa (Windows Firewall)**

**1.5.8.**     **Lập lịch làm việc (Task Scheduler)**

**1.5.9.**     **Quản lý đĩa (Disk Management)**

**1.5.10.Cài đặt và cấu hình dịch vụ Web (Internet Information Service)**

**1.6.**  **Một số thao tác mạng**

**1.6.1.**     **Điều khiển máy tính từ xa thông qua Remote Desktop Connection**

**1.6.2.**     **Chia sẻ thư mục thông qua mạng Lan**

**1.6.3.**     **Cấu hình cho phép kết nối SQL Server từ xa**

**2.**      **Các bài tập về tiến trình, luồng (C#)**

**2.1.**  **Sign**

**2.1.1.**     **Viết chương trình in ra màn hình số tăng dần bắt đầu từ 1, nếu ấn phím Ctrl+C thì thông báo “This program ran for … seconds”, sau đó tắt chương trình (… là số thời gian chương trình đã chạy)**

class bai1

{

 static DateTime start;

 public static void Main1()

 {

 start = DateTime.Now;

 //Add event handler for Ctrl+C command

 Console.CancelKeyPress += new ConsoleCancelEventHandler(Console\_CancelKeyPress);

 int counter = 0;

 while (true)

 {

 Console.WriteLine(++counter);

 System.Threading.Thread.Sleep(500);

 }

 }

 static void Console\_CancelKeyPress(object sender, ConsoleCancelEventArgs e)

 {

 var end = DateTime.Now;

 Console.WriteLine("This program ran for {0:000.000} seconds.", (end - start).TotalMilliseconds / 1000);

 Environment.Exit(0);

 }

}

**2.1.2.**     **Viết chương trình cho phép chạy hoặc đóng ứng dụng với tên ứng dụng được nhập từ bàn phím**

**2.1.3.**     **Viết chương trình hiển thị danh sách các tiến trình đang chạy, nếu chọn tiến trình nào và ấn “Đóng ứng dụng” thì tiến trình sẽ tắt**

**2.2.**  **Pipe**

//Tất cả bài về Pipe đều thêm  thư viện:

using System.IO.Pipes;

**2.2.1.**     **Viết chương trình gửi và nhận thông điệp sử dụng Pipe**

//Sever

class clsPipeServer

{

 public static void Main1()

 {

 using (var s = new NamedPipeServerStream("FooPipe", PipeDirection.InOut))

 {

 s.WaitForConnection();

 // convert the message to byte array

 var data = Encoding.Unicode.GetBytes("Hello! Welcome to FooPipe Client!");

 // send data to clients

 s.Write(data, 0, data.Length);

 var buffer = new byte[1000];

 // read received data into buffer

 s.Read(buffer, 0, 1000);

 Console.WriteLine(Encoding.Unicode.GetString(buffer));

 }

 Console.Read();

 }

}

//Client

class clsPipeClient

{

 public static void Main1()

 {

 using (var stream = new NamedPipeClientStream("FooPipe"))

 {

 stream.Connect();

 var buffer = new byte[1000];

 // read received data into buffer

 stream.Read(buffer, 0, 1000);

 Console.WriteLine(Encoding.Unicode.GetString(buffer));

 var data = Encoding.Unicode.GetBytes("Hello! Welcome to FooPipe Server!");

 // send data to clients

 stream.Write(data, 0, data.Length);

 }

 Console.Read();

 }

}

**2.3.**  **Socket**

//Tất cả bài về socket đều thêm  thư viện:

using System.Net;

using System.Net.Sockets;

using System.IO;

**2.3.1.**     **Viết chương trình nhập vào một số n, tính:**

**2.3.1.1.**          **Tổng 1+3+5+7+...+(2n+1)**

//Sever

class bai1

    {

        public static void Main1()

        {

            try

            {

                IPAddress address = IPAddress.Parse("127.0.0.1");

                TcpListener listener = new TcpListener(address, 9999);

                listener.Start();

                Socket socket = listener.AcceptSocket();

                var stream = new NetworkStream(socket);

                var reader = new StreamReader(stream);

                var writer = new StreamWriter(stream);

                writer.AutoFlush = true;

                while (true)

                {

                    int n = Convert.ToInt32(reader.ReadLine());

                    Console.WriteLine("Ban da nhap n = : " + n);

                    int tong = 0;

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

                    {

                        tong += (2 \* i + 1);

                    }

                    writer.WriteLine(tong);

                }

                socket.Close();

                listener.Stop();

            }

            catch (Exception ex)

            {

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

            }

            Console.Read();

        }

    }

//Client

class bai1

    {

        public static void Main1()

        {

            try

            {

                TcpClient client = new TcpClient();

                client.Connect("127.0.0.1", 9999);

                Stream stream = client.GetStream();

                while (true)

                {

                    var reader = new StreamReader(stream);

                    var writer = new StreamWriter(stream);

                    writer.AutoFlush = true;

                    Console.Write("Nhap n: ");

                    string n = Console.ReadLine();

                    writer.WriteLine(n);

                    Console.WriteLine("Tong la: " + reader.ReadLine());

                }

                stream.Close();

                client.Close();

            }

            catch (Exception ex)

            {

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

            }

            Console.Read();

        }

    }

**2.3.1.2.**          **Tổng 1\*2 + 2\*3+...+n\*(n+1)**

//Sever

class bai1

    {

        public static void Main1()

        {

            try

            {

                IPAddress address = IPAddress.Parse("127.0.0.1");

                TcpListener listener = new TcpListener(address, 9999);

                listener.Start();

                Socket socket = listener.AcceptSocket();

                var stream = new NetworkStream(socket);

                var reader = new StreamReader(stream);

                var writer = new StreamWriter(stream);

                writer.AutoFlush = true;

                while (true)

                {

                    int n = Convert.ToInt32(reader.ReadLine());

                    Console.WriteLine("Ban da nhap n = : " + n);

                    int tong = 0;

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

                    {

                        tong += i\* ( i + 1);

                    }

                    writer.WriteLine(tong);

                }

                socket.Close();

                listener.Stop();

            }

            catch (Exception ex)

            {

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

            }

            Console.Read();

        }

    }

//Client

class bai1

    {

        public static void Main1()

        {

            try

            {

                TcpClient client = new TcpClient();

                client.Connect("127.0.0.1", 9999);

                Stream stream = client.GetStream();

                while (true)

                {

                    var reader = new StreamReader(stream);

                    var writer = new StreamWriter(stream);

                    writer.AutoFlush = true;

                    Console.Write("Nhap n: ");

                    string n = Console.ReadLine();

                    writer.WriteLine(n);

                    Console.WriteLine("Tong la: " + reader.ReadLine());

                }

                stream.Close();

                client.Close();

            }

            catch (Exception ex)

            {

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

            }

            Console.Read();

        }

    }

**2.3.1.3.**          **Tổng 1-2+3-4+...+(2n+1)**

//Sever

class bai1

    {

        public static void Main1()

        {

            try

            {

                IPAddress address = IPAddress.Parse("127.0.0.1");

                TcpListener listener = new TcpListener(address, 9999);

                listener.Start();

                Socket socket = listener.AcceptSocket();

                var stream = new NetworkStream(socket);

                var reader = new StreamReader(stream);

                var writer = new StreamWriter(stream);

                writer.AutoFlush = true;

                while (true)

                {

                    int n = Convert.ToInt32(reader.ReadLine());

                    Console.WriteLine("Ban da nhap n = : " + n);

                    int tong = 0;

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

                    {

                        if(i%2==0)

                        {

                            tong = tong - (2 \* i + 1);

                        }

                        else

                        {

                            tong = tong + (2 \* i + 1);

                        }

                    }

                    writer.WriteLine(tong);

                }

                socket.Close();

                listener.Stop();

            }

            catch (Exception ex)

            {

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

            }

            Console.Read();

        }

    }

//Client

class bai1

    {

        public static void Main1()

        {

            try

            {

                TcpClient client = new TcpClient();

                client.Connect("127.0.0.1", 9999);

                Stream stream = client.GetStream();

                while (true)

                {

                    var reader = new StreamReader(stream);

                    var writer = new StreamWriter(stream);

                    writer.AutoFlush = true;

                    Console.Write("Nhap n: ");

                    string n = Console.ReadLine();

                    writer.WriteLine(n);

                    Console.WriteLine("Tong la: " + reader.ReadLine());

                }

                stream.Close();

                client.Close();

            }

            catch (Exception ex)

            {

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

            }

            Console.Read();

        }

    }

**2.3.1.4.**          **Fibonaxi của một số**

//Sever

class bai1

    {

        public static void Main1()

        {

            try

            {

                IPAddress address = IPAddress.Parse("127.0.0.1");

                TcpListener listener = new TcpListener(address, 9999);

                listener.Start();

                Socket socket = listener.AcceptSocket();

                var stream = new NetworkStream(socket);

                var reader = new StreamReader(stream);

                var writer = new StreamWriter(stream);

                writer.AutoFlush = true;

                while (true)

                {

                    int n = Convert.ToInt32(reader.ReadLine());

                    Console.WriteLine("Ban da nhap n = : " + n);

                    writer.WriteLine(fibonaxi(n));

                }

                socket.Close();

                listener.Stop();

            }

            catch (Exception ex)

            {

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

            }

            Console.Read();

        }

        public static int fibonaxi(int s)

        {

            if (s == 0) return 0;

            else

            {

                if (s == 1 || s == 2 ) return 1;

                else

                    return (fibonaxi(s - 2) + fibonaxi(s - 1));

            }

        }

    }

//Client

class bai1

    {

        public static void Main1()

        {

            try

            {

                TcpClient client = new TcpClient();

                client.Connect("127.0.0.1", 9999);

                Stream stream = client.GetStream();

                while (true)

                {

                    var reader = new StreamReader(stream);

                    var writer = new StreamWriter(stream);

                    writer.AutoFlush = true;

                    Console.Write("Nhap n: ");

                    string n = Console.ReadLine();

                    writer.WriteLine(n);

                    Console.WriteLine("fibonaxi la: " + reader.ReadLine());

                }

                stream.Close();

                client.Close();

            }

            catch (Exception ex)

            {

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

            }

            Console.Read();

        }

    }

**2.3.1.5.**          **Tổng S1=1\*2\*3\*...\*n và S2 = 1+2+3+ ... +n**

//Sever

class bai1

    {

        public static void Main1()

        {

            try

            {

                IPAddress address = IPAddress.Parse("127.0.0.1");

                TcpListener listener = new TcpListener(address, 9999);

                listener.Start();

                Socket socket = listener.AcceptSocket();

                var stream = new NetworkStream(socket);

                var reader = new StreamReader(stream);

                var writer = new StreamWriter(stream);

                writer.AutoFlush = true;

                while (true)

                {

                    int n = Convert.ToInt32(reader.ReadLine());

                    Console.WriteLine("Ban da nhap n = : " + n);

                    int tong1 = 1;

                    int tong2 = 0;

                    for (int i = 1; i < n+1; i++)

                    {

                        tong1 = tong1 \* i;

                        tong2 = tong2 + i;

                    }

                    writer.WriteLine(tong1);

                    writer.WriteLine(tong2);

                }

                socket.Close();

                listener.Stop();

            }

            catch (Exception ex)

            {

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

            }

            Console.Read();

        }

    }

//Client

class bai1

    {

        public static void Main1()

        {

            try

            {

                TcpClient client = new TcpClient();

                client.Connect("127.0.0.1", 9999);

                Stream stream = client.GetStream();

                while (true)

                {

                    var reader = new StreamReader(stream);

                    var writer = new StreamWriter(stream);

                    writer.AutoFlush = true;

                    Console.Write("Nhap n: ");

                    string n = Console.ReadLine();

                    writer.WriteLine(n);

                    Console.WriteLine("Tong S1 la: " + reader.ReadLine());

                    Console.WriteLine("Tong S2 la: " + reader.ReadLine());

                }

                stream.Close();

                client.Close();

            }

            catch (Exception ex)

            {

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

            }

            Console.Read();

        }

    }

**2.3.2.**     **Viết chương trình nhập vào hai số a, b, tính:**

**2.3.2.1.**          **Tổng bình phương a và b**

//Sever

class bai1

    {

        public static void Main1()

        {

            try

            {

                IPAddress address = IPAddress.Parse("127.0.0.1");

                TcpListener listener = new TcpListener(address, 9999);

                listener.Start();

                Socket socket = listener.AcceptSocket();

                var stream = new NetworkStream(socket);

                var reader = new StreamReader(stream);

                var writer = new StreamWriter(stream);

                writer.AutoFlush = true;

                while (true)

                {

                    string str = reader.ReadLine();

                    string[] n = str.Split("|".ToCharArray());

                    int a = Convert.ToInt32(n[0]);

                    int b = Convert.ToInt32(n[1]);

                    Console.WriteLine("Ban da nhap a = " + a + ", b = " + b);

                    int tong = a \* a + b \* b;

                    writer.WriteLine(tong);

                }

                socket.Close();

                listener.Stop();

            }

            catch (Exception ex)

            {

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

            }

            Console.Read();

        }

    }

//Client

class bai1

    {

        public static void Main1()

        {

            try

            {

                TcpClient client = new TcpClient();

                client.Connect("127.0.0.1", 9999);

                Stream stream = client.GetStream();

                while (true)

                {

                    var reader = new StreamReader(stream);

                    var writer = new StreamWriter(stream);

                    writer.AutoFlush = true;

                    Console.Write("Nhap a: ");

                    string a = Console.ReadLine();

                    Console.Write("Nhap b: ");

                    string b = Console.ReadLine();

                    writer.WriteLine(a + "|" + b);

                    Console.WriteLine("Tong la: " + reader.ReadLine());

                }

                stream.Close();

                client.Close();

            }

            catch (Exception ex)

            {

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

            }

            Console.Read();

        }

    }

**2.3.2.2.**          **Tổng lập phương a và b**

//Sever

class bai1

    {

        public static void Main1()

        {

            try

            {

                IPAddress address = IPAddress.Parse("127.0.0.1");

                TcpListener listener = new TcpListener(address, 9999);

                listener.Start();

                Socket socket = listener.AcceptSocket();

                var stream = new NetworkStream(socket);

                var reader = new StreamReader(stream);

                var writer = new StreamWriter(stream);

                writer.AutoFlush = true;

                while (true)

                {

                    string str = reader.ReadLine();

                    string[] n = str.Split("|".ToCharArray());

                    int a = Convert.ToInt32(n[0]);

                    int b = Convert.ToInt32(n[1]);

                    Console.WriteLine("Ban da nhap a = " + a + ", b = " + b);

                    int tong = a \* a \* a + b \* b \* b ;

                    writer.WriteLine(tong);

                }

                socket.Close();

                listener.Stop();

            }

            catch (Exception ex)

            {

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

            }

            Console.Read();

        }

    }

//Client

class bai1

    {

        public static void Main1()

        {

            try

            {

                TcpClient client = new TcpClient();

                client.Connect("127.0.0.1", 9999);

                Stream stream = client.GetStream();

                while (true)

                {

                    var reader = new StreamReader(stream);

                    var writer = new StreamWriter(stream);

                    writer.AutoFlush = true;

                    Console.Write("Nhap a: ");

                    string a = Console.ReadLine();

                    Console.Write("Nhap b: ");

                    string b = Console.ReadLine();

                    writer.WriteLine(a + "|" + b);

                    Console.WriteLine("Tong la: " + reader.ReadLine());

                }

                stream.Close();

                client.Close();

            }

            catch (Exception ex)

            {

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

            }

            Console.Read();

        }

    }

**2.3.2.3.**          **Tính ước số chung lớn nhất a và b**

//Sever

class BAI1

    {

        public static void main1()

        {

            try

            {

                IPAddress address = IPAddress.Parse("127.0.0.1");

                TcpListener listener = new TcpListener(address, 9999);

                listener.Start();

                Socket socket = listener.AcceptSocket();

                var stream = new NetworkStream(socket);

                var reader = new StreamReader(stream);

                var writer = new StreamWriter(stream);

                writer.AutoFlush = true;

                while(true)

                {

                    int a = Convert.ToInt32(reader.ReadLine());

                    Console.WriteLine("ban da nhap a = : " + a);

                    int b = Convert.ToInt32(reader.ReadLine());

                    Console.WriteLine("ban da nhap b = : " + b);

                    int uc, bc;

                    for (uc = a; uc >= 1; uc--)

                    {

                        if (a % uc == 0 && b % uc == 0)

                        {

                            writer.WriteLine(uc);

                            break;

                        }

                    }

                    //for (bc = a; bc <= a \* b; bc++)

                    //{

                    //    if (bc % a == 0 && bc % b == 0)

                    //    {

                    //        writer.WriteLine(bc);

                    //        break;

                    //    }

                    //}

                }

                socket.Close();

                listener.Stop();

            }

            catch(Exception ex)

            {

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

            }

            Console.Read();

        }

    }

//Client

class bai1

    {

        public static void main1()

        {

            try

            {

                TcpClient client = new TcpClient();

                client.Connect("127.0.0.1", 9999);

                Stream stream = client.GetStream();

                while(true)

                {

                    var reader = new StreamReader(stream);

                    var writer = new StreamWriter(stream);

                    writer.AutoFlush = true;

                    Console.Write("nhap a: ");

                    string a = Console.ReadLine();

                    Console.Write("nhap b: ");

                    string b = Console.ReadLine();

                    writer.WriteLine(a);

                    writer.WriteLine(b);

                    Console.WriteLine("uoc chung la: " + reader.ReadLine());

                    //Console.WriteLine("boi chung la: " + reader.ReadLine());

                }

                stream.Close();

                client.Close();

            }

            catch (Exception ex)

            {

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

            }

            Console.Read();

        }

    }

2.3.2.4.          Tính bội số chung nhỏ nhất a và b

//Sever

class BAI1

    {

        public static void main1()

        {

            try

            {

                IPAddress address = IPAddress.Parse("127.0.0.1");

                TcpListener listener = new TcpListener(address, 9999);

                listener.Start();

                Socket socket = listener.AcceptSocket();

                var stream = new NetworkStream(socket);

                var reader = new StreamReader(stream);

                var writer = new StreamWriter(stream);

                writer.AutoFlush = true;

                while(true)

                {

                    int a = Convert.ToInt32(reader.ReadLine());

                    Console.WriteLine("ban da nhap a = : " + a);

                    int b = Convert.ToInt32(reader.ReadLine());

                    Console.WriteLine("ban da nhap b = : " + b);

                    int uc, bc;

//for (uc = a; uc >= 1; uc--)

                    //{

                    //    if (a % uc == 0 && b % uc == 0)

                    //    {

                    //        writer.WriteLine(uc);

                    //        break;

                    //    }

                    //}

                    for (bc = a; bc <= a \* b; bc++)

                    {

                        if (bc % a == 0 && bc % b == 0)

                        {

                            writer.WriteLine(bc);

                            break;

                        }

                    }

                }

                socket.Close();

                listener.Stop();

            }

            catch(Exception ex)

            {

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

            }

            Console.Read();

        }

    }

//Client

class bai1

    {

        public static void main1()

        {

            try

            {

                TcpClient client = new TcpClient();

                client.Connect("127.0.0.1", 9999);

                Stream stream = client.GetStream();

                while(true)

                {

                    var reader = new StreamReader(stream);

                    var writer = new StreamWriter(stream);

                    writer.AutoFlush = true;

                    Console.Write("nhap a: ");

                    string a = Console.ReadLine();

                    Console.Write("nhap b: ");

                    string b = Console.ReadLine();

                    writer.WriteLine(a);

                    writer.WriteLine(b);

                    //Console.WriteLine("uoc chung la: " + reader.ReadLine());

                    Console.WriteLine("boi chung la: " + reader.ReadLine());

                }

                stream.Close();

                client.Close();

            }

            catch (Exception ex)

            {

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

            }

            Console.Read();

        }

    }

2.3.3.     Viết chương trình sử dụng giao thức UDP để nhập một thông điệp và gửi thông điệp đó từ Client tới Server

//Sever

class bai1

    {

        public static void Main1()

        {

            OnStart();

        }

        // Method

        static void OnStart()

        {

            // tạo một đối tượng UdpClient và lắng nghe cổng 2008

            UdpClient udp = new UdpClient(2008);

            // thực hiện listen liên tục

            while (true)

            {

                // xác định điểm Remote IP

                IPEndPoint RemoteIPEndPoint = new IPEndPoint(IPAddress.Any, 0);

                // thu lấy thông tin từ client dạng byte

                Byte[] data = udp.Receive(ref RemoteIPEndPoint);

                // chuyển về string

                string message = Encoding.ASCII.GetString(data);

                // in thông điệp ra

                Console.WriteLine("Address: {0} - Message: {1}", RemoteIPEndPoint.Address, message);

            }

        }

    }

//Client

class bai1

    {

        public static void Main1()

        {

            OnConnect();

        }

        static void OnConnect()

        {

            // thông tin host để kết nối

            string \_host = "127.0.0.1";

            // thông tin về port connect

            int \_port = 2008;

            // tạo một UDP Object

            UdpClient udp = new UdpClient();

            // kết nối tới host

            udp.Connect(\_host, \_port);

            bool isConnect = true;

            while (isConnect)

            {

                // tạo data để gửi đi. Luôn ở dạng Bytes

                Console.Write("Nhap noi dung gui toi Server: ");

                string str = Console.ReadLine();

                if (str == "exit")

                {

                    //isConnect = false;

                    return;

                }

                Byte[] data = Encoding.ASCII.GetBytes(str);

                // gửi data tới host

                udp.Send(data, data.Length);

            }

            Console.ReadLine();

        }

    }

2.3.4.     Viết chương trình sử dụng giao thức TCP/IP để gửi và nhận thông điệp giữa Server và Client dưới dạng byte[]

//Sever

class bai1

    {

        private const int BUFFER\_SIZE = 1024;

        private const int PORT\_NUMBER = 9999;

        static ASCIIEncoding encoding = new ASCIIEncoding();

        public static void Main1()

        {

            try

            {

                IPAddress address = IPAddress.Parse("127.0.0.1");

                TcpListener listener = new TcpListener(address, PORT\_NUMBER);

                // 1. listen

                listener.Start();

                Console.WriteLine("Server started on " + listener.LocalEndpoint);

                Console.WriteLine("Waiting for a connection...");

                Socket socket = listener.AcceptSocket();

                Console.WriteLine("Connection received from " + socket.RemoteEndPoint);

                // 2. receive

                byte[] data = new byte[BUFFER\_SIZE];

                socket.Receive(data);

                string str = encoding.GetString(data);

                Console.WriteLine("Hello: " + str);

                // 3. send

                Console.Write("Enter your Age: ");

                string str1 = Console.ReadLine();

                socket.Send(encoding.GetBytes("Your age: " + str1));

                // 4. close

                socket.Close();

                listener.Stop();

            }

            catch (Exception ex)

            {

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

            }

            Console.Read();

        }

    }

//Client

class bai1

    {

        private const int BUFFER\_SIZE = 1024;

        private const int PORT\_NUMBER = 9999;

        static ASCIIEncoding encoding = new ASCIIEncoding();

        public static void Main1()

        {

            try

            {

                TcpClient client = new TcpClient();

                // 1. connect

                client.Connect("127.0.0.1", PORT\_NUMBER);

                Stream stream = client.GetStream();

                Console.WriteLine("Connected to Y2Server.");

                Console.Write("Enter your name: ");

                string str = Console.ReadLine();

                // 2. send

                byte[] data = encoding.GetBytes(str);

                stream.Write(data, 0, data.Length);

                // 3. receive

                data = new byte[BUFFER\_SIZE];

                stream.Read(data, 0, BUFFER\_SIZE);

                Console.Write(encoding.GetString(data));

                // 4. Close

                stream.Close();

                client.Close();

            }

            catch (Exception ex)

            {

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

            }

            Console.Read();

        }

    }

2.3.5.     Viết chương trình sử dụng giao thức TCP/IP để gửi và nhận thông điệp giữa Server và Client sử dụng StreamReader và StreamWrite

//Sever

class bai1

    {

        private const int PORT\_NUMBER = 9999;

        public static void Main1()

        {

            try

            {

                IPAddress address = IPAddress.Parse("127.0.0.1");

                TcpListener listener = new TcpListener(address, PORT\_NUMBER);

                // 1. listen

                listener.Start();

                Console.WriteLine("Server started on " + listener.LocalEndpoint);

                Console.WriteLine("Waiting for a connection...");

                Socket socket = listener.AcceptSocket();

                Console.WriteLine("Connection received from " + socket.RemoteEndPoint);

                var stream = new NetworkStream(socket);

                var reader = new StreamReader(stream);

                var writer = new StreamWriter(stream);

                writer.AutoFlush = true;

                while (true)

                {

                    // 2. receive

                    string str = reader.ReadLine();

                    if (str.ToUpper() == "EXIT")

                    {

                        writer.WriteLine("bye");

                        break;

                    }

                    // 3. send

                    Console.Write("Enter your Age: ");

                    string str1 = Console.ReadLine();

                    writer.WriteLine("Hello " + str1);

                    Console.WriteLine("Hello: " + str);

                }

                // 4. close

                stream.Close();

                socket.Close();

                listener.Stop();

            }

            catch (Exception ex)

            {

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

            }

            Console.Read();

        }

    }

//Client

class bai1

    {

        private const int PORT\_NUMBER = 9999;

        public static void Main1()

        {

            try

            {

                TcpClient client = new TcpClient();

                // 1. connect

                client.Connect("127.0.0.1", PORT\_NUMBER);

                Stream stream = client.GetStream();

                Console.WriteLine("Connected to Y2Server.");

                while (true)

                {

                    Console.Write("Enter your name: ");

                    string str = Console.ReadLine();

                    var reader = new StreamReader(stream);

                    var writer = new StreamWriter(stream);

                    writer.AutoFlush = true;

                    // 2. send

                    writer.WriteLine(str);

                    // 3. receive

                    str = reader.ReadLine();

                    Console.WriteLine(str);

                    if (str.ToUpper() == "BYE")

                        break;

                }

                // 4. close

                stream.Close();

                client.Close();

            }

            catch (Exception ex)

            {

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

            }

            Console.Read();

        }

    }

2.3.6.     Viết chương trình gửi và nhận File giữa Server và Client

2.3.7.     Viết chương trình chat giữa giữa Server và Client

2.4.  Thread

2.4.1.     Viết chương trình in ra màn hình ký tự x và y đồng thời (sử dụng 1 thread)

class bai1

    {

        public static void Main1()

        {

            Thread t = new Thread(WriteY);

            t.Start();

            //new Thread(WriteY).Start();

            // Đồng thời thực hiện trên main thread.

            //WriteY();

            for (int i = 0; i < 1000; i++) Console.Write("x");

            Console.ReadKey();

        }

        static void WriteY()

        {

            for (int i = 0; i < 1000; i++) Console.Write("y");

        }

    }

2.4.2.     Viết chương trình in ra màn hình chữ “done”, sử dụng chung biến done (sử dụng 1 thread)

class bai1

    {

        static bool done = false;

        public static void Main1()

        {

            new Thread(Go).Start(); // Call Go() on a new thread

            Go(); // Call Go() on the main thread

        }

        static void Go()

        {

            if (!done)

            {

                done = true; Console.WriteLine("Done");

            }

            Console.ReadKey();

        }

    }

2.4.3.     Viết chương trình in ra màn hình ký tự “a”, “b”, “c” đồng thời, trong đó ký tự “c” không được in trước ký tự “b” (Sử dụng 3 Thread)

class bai1

    {

        public static void Main1()

        {

            Thread t1 = new Thread(MethodA);

            Thread t2 = new Thread(MethodB);

            Thread t3 = new Thread(MethodC);

            t1.Start();

            t2.Start();

            t2.Join();

            t3.Start();

            Console.ReadKey();

        }

        static void MethodA()

        {

            for (int i = 0; i < 100; i++) Console.Write("a");

            Thread.Sleep(200);

        }

        static void MethodB()

        {

            for (int i = 0; i < 100; i++) Console.Write("b");

            Thread.Sleep(100);

        }

        static void MethodC()

        {

            for (int i = 0; i < 100; i++) Console.Write("c");

        }

    }

2.4.4.     Viết chương trình hiển thị tên thread đang chạy (main hay worker, sử dụng 1 thread)

class bai1

    {

        public static void Main1()

        {

            Thread.CurrentThread.Name = "main";

            Thread worker = new Thread(Go);

            worker.Name = "worker";

            worker.Start();

            Go();

            Console.ReadKey();

        }

        static void Go()

        {

            for (int i = 0; i < 100; i++)

                Console.WriteLine("Hello from " + Thread.CurrentThread.Name);

            Thread.Sleep(100);

        }

    }

2.4.5.     Viết chương trình có sử dụng thuộc tính IsBackground của Thread để kiểm tra xem sự khác nhau khi sử dụng và không sử dụng thuộc tính này của Thread (sử dụng 1 Thread)

class bai1

    {

        public static void Main1()

        {

            Thread t1 = new Thread(() =>

            {

                Thread.Sleep(1000);

                Console.WriteLine("Thread t1 started");

            });

            t1.IsBackground = true;

            t1.Start();

            Console.WriteLine("Main thread ending...");

            Console.ReadKey();

        }

    }

2.4.6.     Viết chương trình sử dụng ThreadPool để in ra đồng thời dòng “Thread callback:” và “Thread callback: 123” (Sử dụng 2 Thread)

class bai1

    {

        public static void Main1()

        {

            ThreadPool.QueueUserWorkItem(ThreadProc);

            ThreadPool.QueueUserWorkItem(ThreadProc, 123);

            Console.ReadKey();

        }

        static void ThreadProc(object data)

        {

            for (int i = 0; i < 10; i++)

            {

                Console.WriteLine("Thread callback: " + data);

                Thread.Sleep(500);

            }

        }

    }

2.4.7.     Viết chương trình sử dụng lệnh lock trong Thread để đồng bộ hóa dữ liệu trong việc in số từ 1 đến 100 ra màn hình (sử dụng 2 Thread)

class bai1

    {

        public static void Main1()

        {

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

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

            t1.Start();

            t2.Start();

            Console.ReadLine();

        }

        static void IncreaseAmount()

        {

            int amount = 0;

            for (int i = 0; i < 100; i++)

            {

                amount++;

                if (amount > 0)

                {

                    Thread.Sleep(1);

                    Console.Write(amount + "\t");

                }

            }

        }

        static void DecreaseAmount()

        {

            int amount = 0;

            for (int i = 0; i < 100; i++)

            {

                amount--;

            }

        }

    }

2.4.8.     Viết chương trình mô phỏng quá trình Deadlock (sử dụng 2 Thread)

class bai1

    {

        public static void Main1()

        {

            Thread t1 = new Thread(Foo);

            Thread t2 = new Thread(Bar);

            t1.Start();

            t2.Start();

            Console.ReadLine();

        }

       static void Foo()

        {

            object syncObj1 = new object();

            object syncObj2 = new object();

            Console.WriteLine("Inside Foo method");

            lock (syncObj1)

            {

                Console.WriteLine("Foo: lock(syncObj1)");

                Thread.Sleep(100);

                lock (syncObj2)

                {

                    Console.WriteLine("Foo: lock(syncObj2)");

                }

            }

        }

        static void Bar()

        {

            object syncObj1 = new object();

            object syncObj2 = new object();

            Console.WriteLine("Inside Bar method");

            lock (syncObj2)

            {

                Console.WriteLine("Bar: lock(syncObj2)");

                Thread.Sleep(100);

                lock (syncObj1)

                {

                    Console.WriteLine("Bar: lock(syncObj1)");

                }

            }

        }

    }

3.      Lập lịch

3.1.  Cho 4 tiến trình , tất cả các tiến trình đi vào theo thứ tự tại thời điểm 0:

|  |  |  |
| --- | --- | --- |
| **Tiến trình** | **Thời gian CPU** | **Độ ưu tiên** |
| P1 | 6 | 1 |
| P2 | 3 | 2 |
| P3 | 12 | 3 |
| P4 | 4 | 4 |

Vẽ sơ đồ Grantt và tính thời gian chờ trung bình và thời gian xoay vòng trung bình cho các giải thuật định thời:

a.      First Come First Serve (FCFS)

b.      Shortest Remaining Time First (SRTF)

c.      Priority based scheduling (Định thời dựa trên độ ưu tiên)

d.      Round Robin (RR) với quantum = 3

3.2.  Hãy cho biết kết quả điều phối theo các chiến lược (Vẽ biểu đồ Grantt); tính thời gian chờ cho từng tiến trình; thời gian chờ trung bình trong các chiến lược trên với các tiến trình cho ở hình dưới:

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiến trình** | **Thời điểm đến** | **Thời gian CPU** | **Độ ưu tiên** |
| P1 | 0 | 10 | 3 |
| P2 | 1 | 1 | 1 |
| P3 | 2.5 | 2 | 3 |
| P4 | 3 | 1 | 4 |
| P5 | 4.5 | 5 | 2 |

a.      FCFS

b.      SJF

c.      Round Robin với q = 2

d.      Độ ưu tiên độc quyền

e.      Độ ưu tiên không độc quyền

3.3.   Hãy cho biết các kết quả điều phối chiến lược FCFS và SJF (Vẽ biểu đồ Grantt) và thời gian chờ của từng chiến lược với các tiến trình cho ở hình dưới:

|  |  |  |
| --- | --- | --- |
| **Tiến trình** | **Thời điểm đến** | **Thời gian CPU** |
| P1 | 0 | 8 |
| P2 | 0.4 | 4 |
| P3 | 1 | 1 |

3.4.   Điều phối các tiến trình sau theo chiến lược điều phối độ ưu tiên độc quyền (Vẽ biểu đồ Grantt), hãy tính thời gian chờ cho từng tiến trình và thời gian chờ trung bình:

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiến trình** | **Chiều dài CPU burst** | **Thời điểm đến** | **Độ ưu tiên** |
| P1 | 2 | 0 | 2 |
| P2 | 5 | 1 | 3 |
| P3 | 3 | 2 | 1 |
| P4 | 4 | 3 | 0 |