

XÂY DỰNG ỨNG DỤNG PHÂN TÁN

Chương 0: Giới thiệu môn học

Nguyễn Duy Phúc

duyphucit@live.com

Vĩnh Long, 08/2013

Tổ chức môn học

- Thời gian học:
120h = 20 tuần x 6h
- Từ **26/08/2013** đến **11/01/2014**
- Kiểm tra thường xuyên: **5**
- Thi cuối kỳ: **Thực hành**
- Điều kiện dự thi: dự giảng **$\geq 80\%$** , trung bình kiểm tra thường xuyên **≥ 5.0**

Nội dung môn học

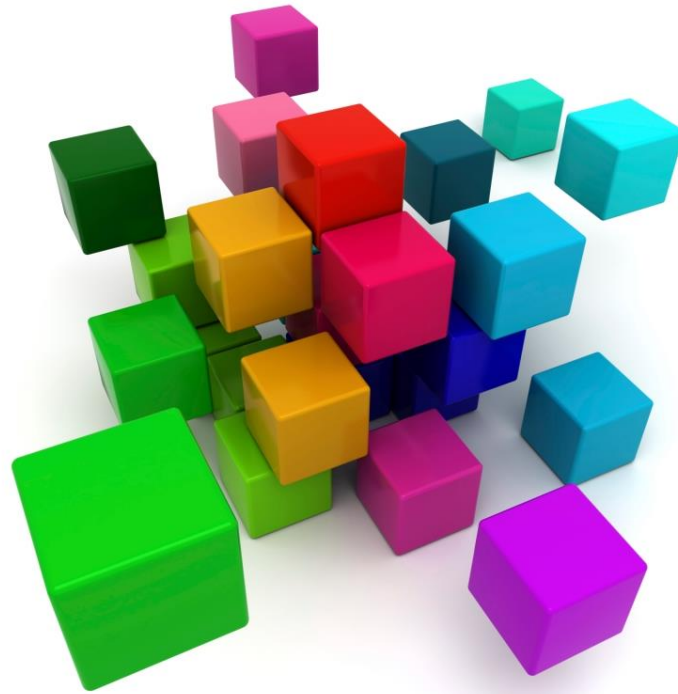
- ❑ Chương 1: Tìm hiểu Internet và lập trình mạng
- ❑ Chương 2: Nhập xuất trong môi trường .NET
- ❑ Chương 3: Sockets
- ❑ Chương 4: HTTP
- ❑ Chương 5: SMTP và POP3
- ❑ Chương 6: FTP
- ❑ Chương 7: Web Services
- ❑ Chương 8: Tối ưu chương trình

Tài liệu tham khảo

- Slides bài giảng môn học
- Fiach Reid: **Network Programming in .NET With C# and Visual Basic .NET** – Elsevier Digital Press, 2004
- Vũ Nguyên & Hoàng Đức Hải: **Lập trình mạng trên Windows** – NXB Thống Kê, 2004
- Keyword: distributed application programming, network programming

Thông tin liên lạc

- Nguyễn Duy Phúc
- Khoa Tin học – Kế toán, Trường Cao đẳng Sư phạm Kỹ thuật Vĩnh Long
- Email: duyphucit@live.com,
duyphucspkt@gmail.com
- Website môn học: sdrv.ms/ZANGIV
→ [Bai_giang](#) → [Distributed Application Development](#)



XÂY DỰNG ỨNG DỤNG PHÂN TÁN

Chương 1: Tìm hiểu Internet và lập trình mạng

Nguyễn Duy Phúc

duyphucit@live.com

Vĩnh Long, 08/2013

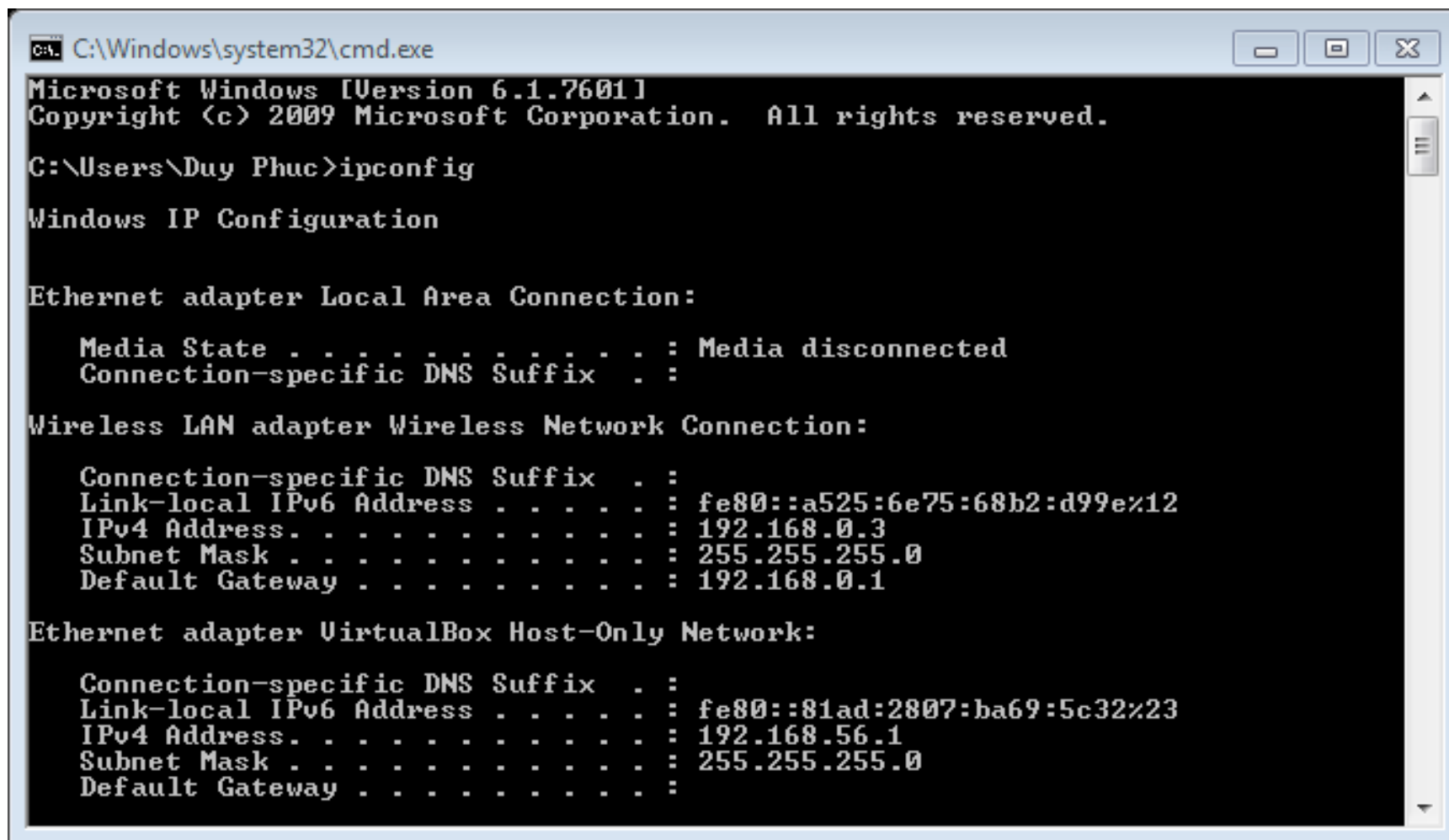
Ứng dụng phân tán (Distributed Application)

- Là ứng dụng mà quá trình xử lý được chia ra cho nhiều máy tính trên mạng thực hiện
- Thông thường một ứng dụng phân tán gồm có 3 tầng (layer):
 - Trình bày (Presentation)
 - Nghiệp vụ (Business Logic)
 - Dữ liệu (Data Store)
- *Khóa học chủ yếu tập trung vào lập trình mạng (network programming) trên môi trường .NET*

Địa chỉ IP

- Mỗi một thiết bị (host) khi kết nối trực tiếp vào mạng Internet đều có một địa chỉ IP (Internet Protocol) duy nhất để phân biệt
- Địa chỉ IP có kích thước 32 bit, thường được biểu diễn dưới dạng 4 byte phân cách bởi dấu chấm (.)
- Ví dụ: 192.168.1.2
- Trong môi trường Windows, có thể dùng lệnh ***ipconfig*** để xem thông tin địa chỉ IP của máy

Địa chỉ IP (2)



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Duy Phuc>ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . :

Wireless LAN adapter Wireless Network Connection:

    Connection-specific DNS Suffix  . :
    Link-local IPv6 Address . . . . . : fe80::a525:6e75:68b2:d99e%12
    IPv4 Address. . . . . : 192.168.0.3
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.0.1

Ethernet adapter VirtualBox Host-Only Network:

    Connection-specific DNS Suffix  . :
    Link-local IPv6 Address . . . . . : fe80::81ad:2807:ba69:5c32%23
    IPv4 Address. . . . . : 192.168.56.1
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . :
```

Kết quả thực hiện lệnh **ipconfig** trên một máy tính chạy Windows 7

Địa chỉ IP (3)

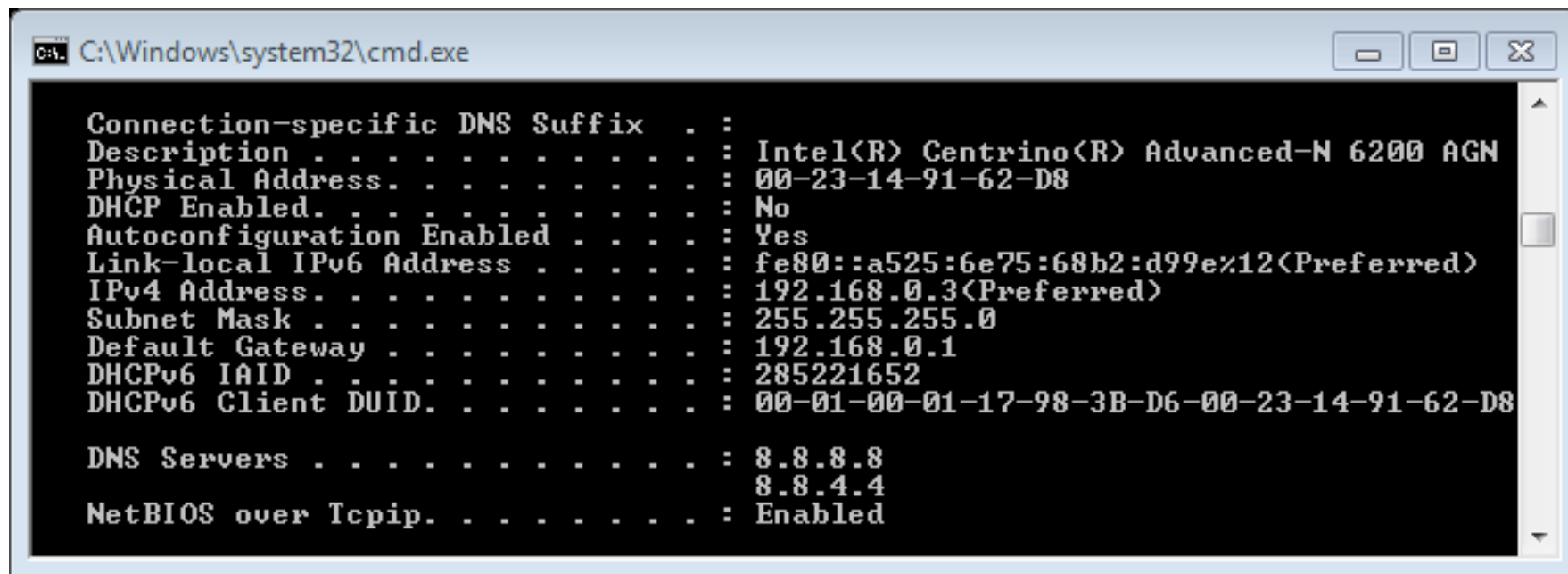
- 32 bit của địa chỉ IP phân làm 2 phần: phần đầu là địa chỉ mạng, phần sau là địa chỉ của host



- 2 host muốn giao tiếp trực tiếp với nhau thì phải cùng địa chỉ mạng
- IP của một host có thể tự thiết lập hoặc được cấp phát từ máy chủ DHCP (Dynamic Host Configuration Protocol)

Địa chỉ IP (4)

- Có thể biết được Network ID và Host ID qua thông tin Subnet Mask được cấp kèm địa chỉ IP
- VD: Chi tiết địa chỉ IP khi dùng lệnh ***ipconfig /all***



```
C:\Windows\system32\cmd.exe

Connection-specific DNS Suffix . : 
Description . . . . . : Intel(R) Centrino(R) Advanced-N 6200 AGN
Physical Address. . . . . : 00-23-14-91-62-D8
DHCP Enabled. . . . . : No
Autoconfiguration Enabled . . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::a525:6e75:68b2:d99e%12(Preferred)
IPv4 Address. . . . . : 192.168.0.3(Preferred)
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.0.1
DHCPv6 IAID . . . . . : 285221652
DHCPv6 Client DUID. . . . . : 00-01-00-01-17-98-3B-D6-00-23-14-91-62-D8

DNS Servers . . . . . : 8.8.8.8
                       8.8.4.4
NetBIOS over Tcpi. . . . . : Enabled
```

Mô hình mạng

□ Mô hình OSI

Tầng	Chức năng	Đơn vị dữ liệu
7. Application	Xử lý của ứng dụng mạng	Data
6. Presentation	Biểu diễn, mã hóa, giải mã, chuyển đổi dữ liệu	Data
5. Session	Quản lý phiên giao tiếp giữa các ứng dụng	Data
4. Transport	Đảm bảo tin cậy, kiểm soát lưu lượng	Segment
3. Network	Tìm đường, định địa chỉ logic	Packet/Datagram
2. Datalink	Định địa chỉ vật lý	Frame
1. Physical	Truyền tải tín hiệu	Bit

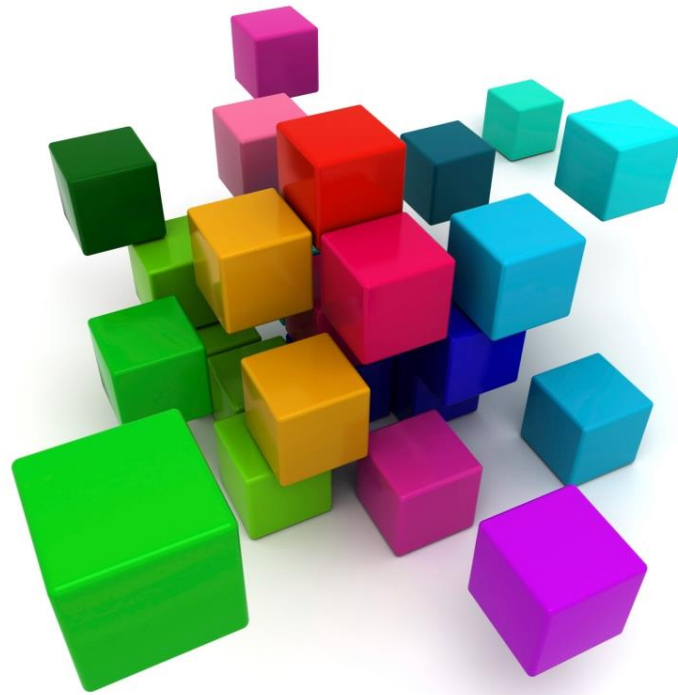
Cổng (Port)

- Dùng để phân biệt dữ liệu mạng sẽ được ứng dụng nào tiếp nhận
- Một vài port thông dụng:
 - 20 & 21: File Transfer Protocol (FTP)
 - 23: Telnet remote login service
 - 25: Simple Mail Transfer Protocol (SMTP)
 - 53: Domain Name System (DNS) service
 - 80: Hypertext Transfer Protocol (HTTP)
 - 110: Post Office Protocol (POP3)
 - 143: Internet Message Access Protocol (IMAP)

.NET Framework



- Được phát triển bởi Microsoft, là tập các thành phần chạy trên Windows cung cấp môi trường để xây dựng và thực thi các ứng dụng
- Ưu điểm: đơn giản, dễ dàng, bảo mật, chạy nhanh trên Windows
- Ngôn ngữ lập trình trên .NET: VB.NET, C#, C++, J#
- IDE (Integrated Development Environment): thường dùng là Visual Studio



XÂY DỰNG ỨNG DỤNG PHÂN TÁN

Chương 2: Nhập xuất trong môi trường .NET

Nguyễn Duy Phúc

duyphucit@live.com

Vĩnh Long, 09/2013

Luồng (Stream)

- Cung cấp một giao tiếp lập trình tương tự nhau cho các thiết bị I/O (đĩa, mạng, máy in,...)
- Stream có 2 cách sử dụng:
 - Đồng bộ (Synchronously): thread gọi stream sẽ tạm dừng đến khi thao tác I/O thực hiện xong
 - Bất đồng bộ (Asynchronously): thread gọi stream sẽ tiếp tục thực hiện, khi thao tác I/O hoàn tất thì một phương thức (method) sẽ được gọi để xác nhận
- Các Stream được khai báo trong namespace **System.IO**

Lớp FileStream

Phương thức / Thuộc tính	Mô tả
Khởi tạo (Constructor)	Tạo mới đối tượng FileStream . Dạng: FileStream(<path>, <mode>) <mode> có thể là: FileMode.Append, FileMode.Create, FileMode.CreateNew, FileMode.Open, FileMode.OpenOrCreate, FileMode.Truncate
Length	Trả về kích thước tập tin
Position	Cho biết hoặc thiết lập vị trí hiện hành của con trỏ tập tin
Write	Ghi dữ liệu vào tập tin. Dạng: Write(<buffer>, <offset>, <count>)
Read	Đọc dữ liệu từ tập tin. Dạng: Read(<buffer>, <offset>, <count>)
BeginRead	Đọc dữ liệu theo kiểu bất đồng bộ (asynchronous)
BeginWrite	Ghi dữ liệu theo kiểu bất đồng bộ
Close	Đóng tập tin sau khi truy xuất xong

Lớp StreamReader

Phương thức / Thuộc tính	Mô tả
Khởi tạo (Constructor)	Tạo mới đối tượng StreamReader để đọc dữ liệu. Dạng: <code>StreamReader(<stream>)</code>
<code>Read()</code>	Đọc một ký tự từ stream
<code>Read(buffer, index, count)</code>	Đọc count ký tự bắt đầu từ vị trí index của stream lưu vào mảng buffer . Trả về số ký tự thực sự đọc được
<code>ReadLine()</code>	Đọc một dòng văn bản trong stream, trả về một String
<code>ReadToEnd()</code>	Đọc từ vị trí hiện tại đến hết stream, trả về một String
<code>EndOfStream</code>	Cho biết con trỏ đã đến cuối luồng dữ liệu đọc hay chưa
<code>Close()</code>	Đóng luồng dữ liệu đọc

Lớp StreamWriter

Phương thức / Thuộc tính	Mô tả
Khởi tạo (Constructor)	Tạo mới đối tượng StreamWriter để dọc dữ liệu. Dạng: StreamWriter(<stream>)
Write(exp)	Ghi giá trị của biểu thức exp vào stream
Write(buffer, index, count)	Ghi count byte bắt đầu từ vị trí index của mảng buffer vào stream
WriteLine(exp)	Ghi giá trị của biểu thức exp vào stream, xong đưa con trỏ xuống dòng mới trong stream
WriteLine(buffer, index, count)	Ghi count byte bắt đầu từ vị trí index của mảng buffer vào stream, xong đưa con trỏ xuống dòng mới trong stream
Flush()	Cập nhật luồng dữ liệu vào stream, xóa bộ đệm của StreamWriter
Close()	Đóng luồng dữ liệu ghi

Lớp BinaryReader

Phương thức / Thuộc tính	Mô tả
Khởi tạo (Constructor)	Tạo mới đối tượng BinaryReader để đọc dữ liệu. Dạng: <code>BinaryReader(<stream>)</code>
<code>Read(buffer, index, count)</code>	Đọc count byte bắt đầu từ vị trí index của stream lưu vào mảng buffer . Trả về số byte thực sự đọc được
<code>ReadChar</code> , <code>ReadBoolean</code> , <code>ReadDecimal</code> , <code>ReadDouble</code> , <code>ReadInt16</code> , <code>ReadInt32</code> , <code>ReadInt64</code> , <code>ReadUInt16</code> , <code>ReadUInt32</code> , <code>ReadUInt64</code>	Đọc một giá trị tương ứng với kiểu chỉ định từ stream, di chuyển con trỏ đến vị trí kế tiếp tương ứng với kích thước của kiểu dữ liệu đọc
<code>ReadBytes(count)</code> <code>ReadChars(count)</code>	Đọc count byte/char kể từ vị trí hiện tại trong stream, trả về một mảng kiểu Byte/Char
<code>EndOfStream</code>	Cho biết con trỏ đã đến cuối luồng dữ liệu đọc hay chưa
<code>Close</code>	Đóng luồng dữ liệu đọc

Lớp BinaryWriter

Phương thức / Thuộc tính	Mô tả
Khởi tạo (Constructor)	Tạo mới đối tượng BinaryWriter để ghi dữ liệu. Dạng: <code>BinaryWriter(<stream>)</code>
Write(exp)	Ghi giá trị của biểu thức exp vào stream
Write(buffer, index, count)	Ghi count byte bắt đầu từ vị trí index của buffer vào stream
Seek(offset, System.IO.SeekOrigin)	Di chuyển con trỏ đến vị trí offset tính từ vị trí SeekOrigin trong stream. SeekOrigin có thể là: Begin, Current, End
Flush()	Cho biết con trỏ đã đến cuối luồng dữ liệu đọc hay chưa
Close	Đóng luồng dữ liệu đọc

Ví dụ

Đọc file

```
Dim buffer as Byte()
```

```
Dim fs as FileStream
```

```
Fs = New FileStream(<filename>, FileMode.Open)
```

```
Redim buffer(fs.Length)
```

```
Fs.Read(buffer, 0, fs.Length)
```

```
Fs.Close()
```

Ví dụ (2)

Ghi file

Dim buffer as Byte()

Dim fs as FileStream

Fs = New FileStream(<filename>, FileMode.Create)

'Chuẩn bị nội dung cần ghi đưa vào buffer

Fs.Write(buffer, 0, buffer.Length)

Fs.Close()

Ví dụ (3)

Đọc file text theo từng dòng

```
Dim line as String
```

```
Dim fs as FileStream = New FileStream(<filename>,  
Filemode.Open)
```

```
Dim sr as StreamReader = New StreamReader(fs)
```

```
While not sr.EndOfStream
```

```
    line = sr.ReadLine()
```

```
End While
```

```
Fs.Close()
```


Ví dụ (4)

Ghi file text theo từng dòng

```
Dim line as String
```

```
Dim fs as FileStream = New FileStream(<filename>,  
Filemode.Create)
```

```
Dim sw as StreamWriter = New StreamWriter(fs)
```

```
Line="Hello World"
```

```
sw.WriteLine(line)
```

```
sw.Flush()
```

```
Fs.Close()
```

Ví dụ (5)

Đọc file dạng nhị phân

```
Dim fs as FileStream = New FileStream(<filename>,  
Filemode.Open)
```

```
Dim br as BinaryReader = New BinaryReader(fs)
```

*‘Đọc file sử dụng phương thức Read của BinaryReader
‘như: Read, ReadByte, ReadChar, ...*

```
Fs.Close()
```

Ví dụ (6)

Ghi file dạng nhị phân

```
Dim fs as FileStream = New FileStream(<filename>,  
Filemode.Create)
```

```
Dim bw as BinaryWriter = New BinaryWriter(fs)
```

'Ghi file sử dụng phương thức Write của BinaryWriter

'Vd: bw.Write(Cbyte(65))

```
Fs.Close()
```

Bài tập

1) Viết chương trình tạo form gồm 1 textbox, 2 nút nhấn là Open và Save

- Nút Open: cho người dùng chọn một tập tin trên máy sau đó nội dung tập tin sẽ được hiển thị trong textbox
- Nút Save: người dùng sẽ chọn và đặt tên cho tập tin, nội dung người dùng soạn thảo trong textbox sẽ được lưu vào tập tin đã chọn

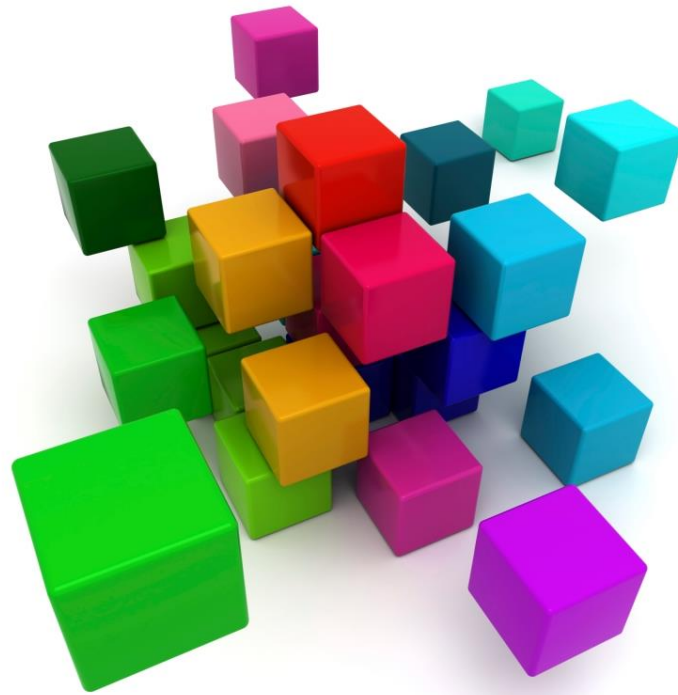
2) Viết chương trình soạn thảo văn bản đơn giản tương tự như notepad

Bài tập (2)

3) Viết chương trình tạo form có một số tùy chỉnh. Thông tin này sẽ được lưu vào một file text như là cấu hình của chương trình. Mỗi khi chương trình được chạy/thoát thì thông tin sẽ được đọc/lưu trong file cấu hình

Gợi ý: có thể tổ chức nội dung file cấu hình gồm nhiều dòng, mỗi dòng là giá trị của một tùy chỉnh theo dạng

<Tùy chỉnh x> = <Giá trị>



XÂY DỰNG ỨNG DỤNG PHÂN TÁN

Chương 3: Lập trình Socket

Nguyễn Duy Phúc

duyphucit@live.com

Vĩnh Long, 15/10/2013

Khái niệm Socket

- Là đối tượng làm giao diện cho một điểm truy cập mạng TCP/IP
- Thông qua socket ta có thể gửi và nhận dữ liệu ở mức độ thấp với 2 giao thức TCP và UDP
- Một gói tin khi truyền sẽ có IP nguồn, đích
- 2 ứng dụng giao tiếp với nhau sẽ thêm số hiệu port vào gói tin để phân biệt
- *Thông thường khi lập trình ta nên chọn port lớn hơn 1024 để tránh trùng với port thông dụng*

TCP/IP và UDP

- Gói tin truyền theo TCP
 - Đảm bảo tính toàn vẹn, tin cậy, thứ tự của dữ liệu
 - Phù hợp với ứng dụng dạng như trao đổi file
- Gói tin truyền theo UDP
 - Đơn giản, không đảm bảo toàn vẹn, thứ tự của dữ liệu
 - Phù hợp với ứng dụng dạng voice, video, game

Socket trong .NET

- Sử dụng namespace System.Net và System.Net.Sockets
- Các lớp thường dùng: UdpClient, TcpClient, TcpListener, Socket, NetworkStream, IPAddress, IPEndPoint

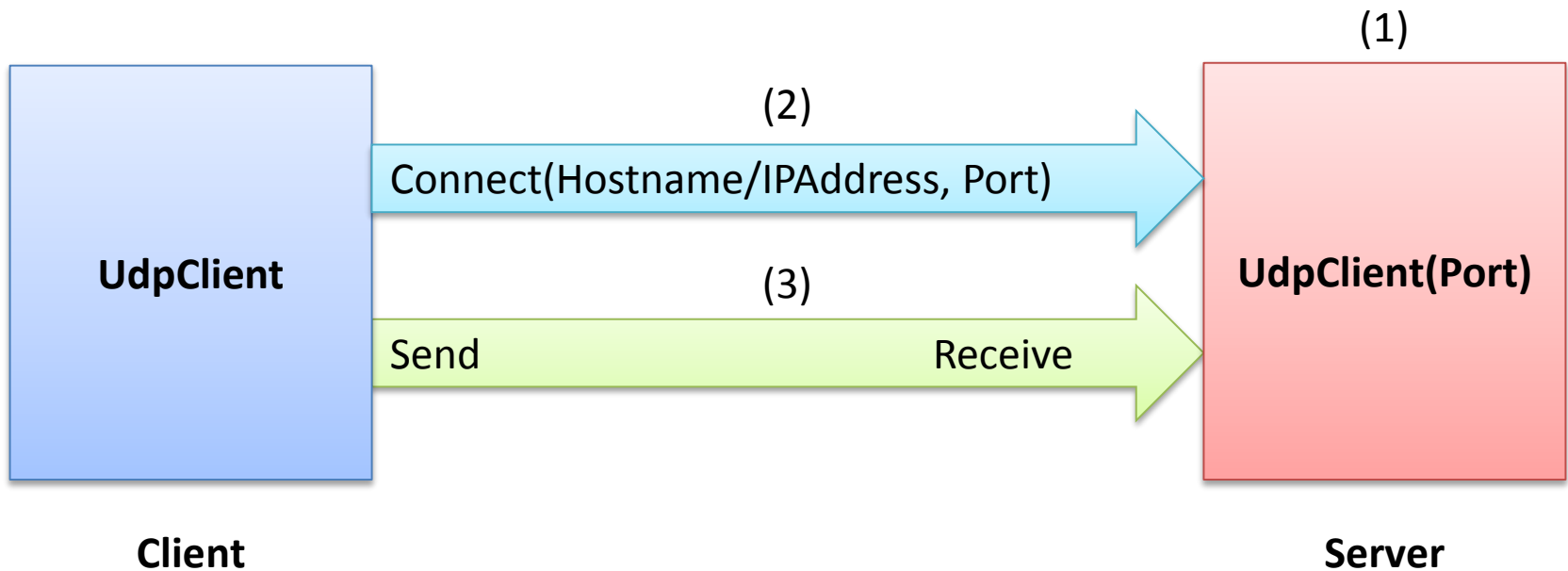
Ứng dụng UDP

□ Sử dụng class **UdpClient**

Phương thức / Thuộc tính	Mô tả
Khởi tạo (Constructor)	+ Tạo mới đối tượng UdpClient . Dạng: UdpClient() + Tạo mới đối tượng UdpClient , hoạt động như server Udp nhận dữ liệu ở cổng Port . Dạng: UdpClient(<Port>) Trong đó Port là số nguyên 32 bit
Connect(IPAddress, Port)	Kết nối với server Udp tại địa chỉ IPAddress , cổng Port
Connect(Hostname, Port)	Kết nối với server Udp tại máy có tên là Hostname, cổng Port
Send(buffer, count)	Gởi count byte dữ liệu trong buffer đến host đã kết nối
Receive(IPEndPoint)	Nhận dữ liệu từ máy client, kết quả trả về là mảng các byte nhận được, IPEndPoint chứa thông tin client
Close	Đóng kết nối

Ứng dụng UDP (2)

- Mô hình kết nối Client – Server



Ứng dụng UDP (3)

Ví dụ: Mã kết nối Server

```
Imports System.Net
```

```
Imports System.Net.Sockets
```

```
...
```

```
Dim server as UdpClient
```

```
server = New UdpClient(<port>)
```

```
...
```

Ứng dụng UDP (4)

Ví dụ: Mã kết nối Client

```
Imports System.Net
```

```
Imports System.Net.Sockets
```

```
...
```

```
Dim client as UdpClient
```

```
client = New UdpClient()
```

```
client.Connect(<IP/hostname>,<port>)
```

```
...
```

Ứng dụng UDP (5)

Ví dụ: Mã Client gửi dữ liệu cho Server

...

```
Dim buffer as Byte()
```

```
buffer = Encoding.UTF8.GetBytes("Hello World")
```

```
client.Send(buffer, buffer.Length)
```

...

Ứng dụng UDP (6)

Ví dụ: Mã Server nhận dữ liệu từ Client

...

While True

Dim ip as New IPEndPoint(IPAddress.Any, 0)

Dim buffer as Byte()

Dim str as String

buffer = server.Receive(ip)

str = Encoding.UTF8.GetString(buffer)

 'Chuỗi nhận được là str

 'Địa chỉ của client trong ip

End While

...

Phân luồng xử lý (Threading)

- ❑ Tách chương trình thành các luồng xử lý song song để tránh hiện tượng Form bị treo khi đợi một thao tác xử lý tốn nhiều thời gian.
- ❑ Sử dụng class **Thread** trong namespace **System.Threading**
- ❑ Ý tưởng:
 - Khai báo chương trình con chứa thao tác xử lý cần tách luồng
 - Tạo một đối tượng Thread có địa chỉ là chương trình con đã tạo

Phân luồng xử lý (2)

Ví dụ: Tách luồng cho server UDP

```
Imports System.Threading
```

```
Dim t as Thread
```

```
t = New Thread(New ThreadStart(AddressOf serverThread))  
t.Start()
```

```
Public Sub serverThread()  
    Dim server as New UdpClient(<port>)  
    While True  
        Dim ip as New IPEndPoint(IPAddress.Any, 0)  
        Dim buffer as Byte()  
        Dim str as String  
        buffer = server.Receive(ip)  
        str = Encoding.UTF8.GetString(buffer)  
    End While  
    server.Close()  
End Sub
```

Phân luồng xử lý (3)

- Mặc nhiên, .NET sẽ kiểm tra và báo lỗi khi luồng này truy xuất dữ liệu của luồng khác
- Để tắt chế độ kiểm tra, khi nạp form ta dùng lệnh

<form>. CheckForIllegalCrossThreadCalls = False

Ứng dụng TCP

- ❑ Có kết nối, đảm bảo truyền/nhận dữ liệu
- ❑ Các lớp liên quan:
 - TCPListener
 - TCPClient
 - NetworkStream

Ứng dụng TCP (2)

□ Lớp TcpListener

Phương thức / Thuộc tính	Mô tả
Khởi tạo (Constructor)	Tạo mới đối tượng TcpListener , nhận kết nối từ mạng có địa chỉ là IP , cổng kết nối là Port . Dạng: TcpListener(IP, Port) Trong đó: <ul style="list-style-type: none">+ IP có kiểu IPAddress+ Port là số nguyên 32 bit
Start()	Bắt đầu nhận kết nối
AcceptTcpClient()	Trả về một TcpClient khi nhận được kết nối
Stop	Ngưng nhận kết nối

Ứng dụng TCP (3)

□ Lớp **TcpClient**

Phương thức / Thuộc tính	Mô tả
Khởi tạo (Constructor)	Tạo mới đối tượng TcpClient
Connect(IPAddress, Port)	Kết nối với TcpListener tại địa chỉ IPAddress , cổng Port
Connect(Hostname, Port)	Kết nối với TcpListener tại máy có tên là Hostname , cổng Port
GetStream()	Trả về một NetworkStream để trao đổi dữ liệu trên kết nối hiện tại
Close()	Đóng kết nối

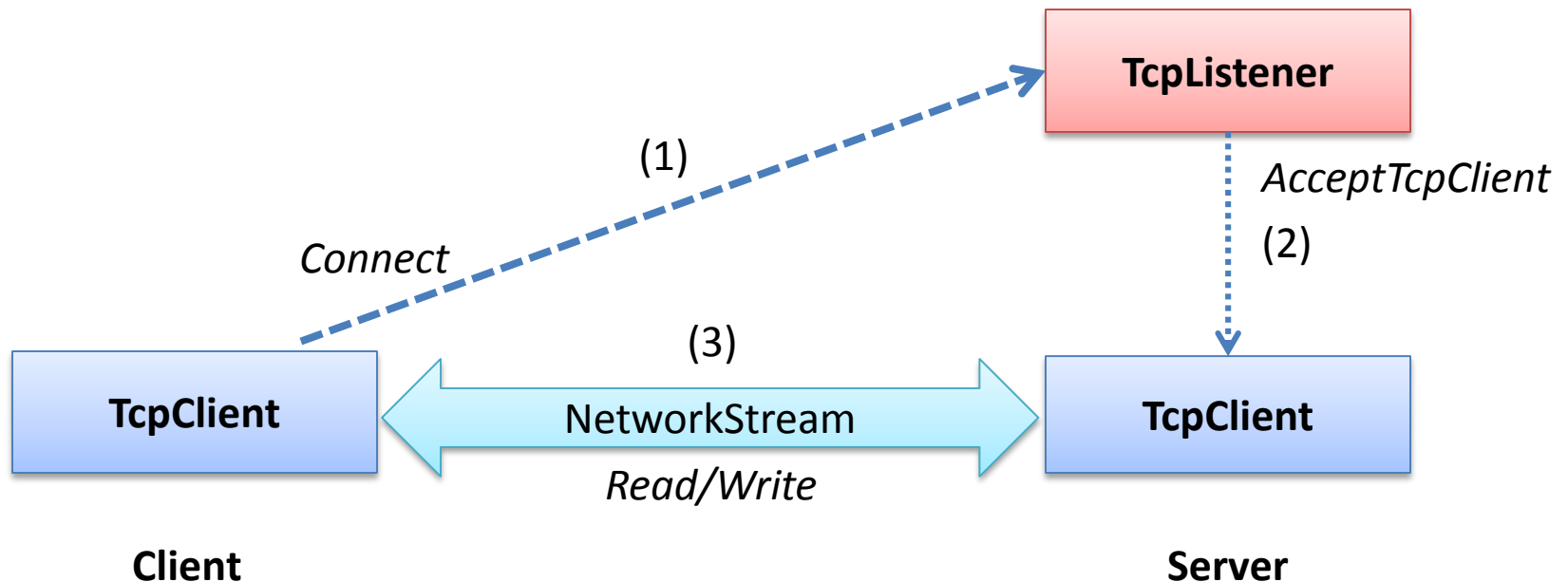
Ứng dụng TCP (4)

□ Lớp **NetworkStream**

Phương thức / Thuộc tính	Mô tả
Khởi tạo (Constructor)	Tạo mới đối tượng NetworkStream
Read(buffer, index, size)	Đọc size byte từ luồng, bắt đầu từ vị trí index , các byte đọc được lưu vào buffer . Kết quả trả về là số byte thực sự đọc được
Write(buffer, index, size)	Ghi size byte bắt đầu từ vị trí index chứa trong buffer vào luồng
Close()	Đóng luồng dữ liệu

Ứng dụng TCP (5)

- Mô hình kết nối Client – Server



Ứng dụng TCP (6)

Ví dụ: Mã kết nối của client

```
Imports System.Net
```

```
Imports System.Net.Sockets
```

```
...
```

```
Dim client as TcpClient
```

```
Dim ns as NetworkStream
```

```
client = New TcpClient(<hostname>,<port>)
```

```
ns = client.GetStream()
```

```
...
```


Ứng dụng TCP (7)

Ví dụ: Mã kết nối của server

```
Imports System.Net
```

```
Imports System.Net.Sockets
```

```
...
```

```
Dim lsn as TcpListener
```

```
Dim client as TcpClient
```

```
Dim ns as NetworkStream
```

```
lsn = New TcpListener(IPAddress.Any, <port>)
```

```
lsn.Start()
```

```
client = lsn.AcceptTcpClient()
```

```
ns = client.GetStream()
```

Ứng dụng TCP (8)

Ví dụ: Mã gửi dữ liệu qua NetworkStream

...

```
Dim buff as Byte()
```

```
buff = Encoding.UTF8.GetBytes(<text>)
```

```
ns.Write(buff, 0, buff.Length)
```

...

Ứng dụng TCP (9)

Ví dụ: Mã nhận dữ liệu từ NetworkStream

...

Dim buff as Byte()

Dim count as Int32

'Nhận 1 khối dữ liệu tối đa 1KB

Redim buff(1024)

count = ns.Read(buff, 0, 1024)

'count = số byte thực sự nhận được

...

Bài tập

Sử dụng UDP/TCP, viết các chương trình sau:

1. CT gửi và nhận dữ liệu một chiều. Việc nhận dữ liệu kết thúc khi chuỗi nhận được là “STOP”
2. CT minh họa client điều khiển giao diện của server bằng cách truyền lệnh.
3. CT chat đơn giản (chỉ có text)
4. CT gửi file qua mạng
5. **CT game đơn giản giữa 2 máy trên mạng**