

# CHƯƠNG 6 TRUYỀN THÔNG VỚI FILE SERVER

ThS. Trần Bá Nhiệm  
Website:  
[sites.google.com/site/tranbanhiem](http://sites.google.com/site/tranbanhiem)  
Email: [tranbanhiem@gmail.com](mailto:tranbanhiem@gmail.com)

## Nội dung

- Giới thiệu
- Microsoft file sharing
- Netware file sharing
- Tổng quan về FTP

## Giới thiệu

- Để hosting website chúng ta cần phải có một thao tác là đưa các trang web đã thiết kế lên server của ISP hoặc công ty cho thuê. Quá trình đưa lên đó thông qua một giao thức truyền file hoặc FTP
- FTP là cơ chế chính để truyền file giữa các máy tính thông qua môi trường Internet

29/06/2011

Chương 6: Truyền thông với file server

3

## Giới thiệu

- FTP được định nghĩa trong RFC 1350 thay thế một giao thức cũ hơn là Trivial file Transfer Protocol (TFTP).
- TFTP rất ít khi dùng, chủ yếu để truyền các file cấu hình giữa các router hoặc khởi động máy tính không dùng đĩa cứng

29/06/2011

Chương 6: Truyền thông với file server

4

## Microsoft file sharing

- Một công nghệ mạnh mẽ của Microsoft là Common Internet File (CIF) – giao thức chia sẻ file rất tự nhiên của Windows 2000 và XP. Nó là phần mở rộng của giao thức Server Message Block (SMB) có trước đó, cung cấp tính năng ổ đĩa mạng và chia sẻ máy in; an toàn hơn, nhanh hơn FTP

29/06/2011

Chương 6: Truyền thông với file server

5

## Microsoft file sharing

- Windows file sharing phổ biến trong mạng văn phòng để chia sẻ máy in, trung tâm lưu trữ file
- Về phương diện lập trình, đó là một ý tưởng công nghệ áp dụng trên môi trường mà tất cả người dùng trên cùng 1 hệ thống mạng

29/06/2011

Chương 6: Truyền thông với file server

6

## Microsoft file sharing

- NETBIOS và NETBEUI là tên chính xác của Microsoft file, print sharing
- Một biến thể của NETBIOS là NBT chạy trên IP, còn các dạng khác thì không – chúng dùng NETBIOS hostname
- Hostname được phân giải thành địa chỉ vật lý

29/06/2011

Chương 6: Truyền thông với file server

7

## Netware file sharing

- Xuất hiện cách đây khoảng hơn hàng chục năm, tuy nhiên là giao thức truyền file nhanh nhất
- Được xây dựng trên nền tảng Internetworking packet exchange / Sequenced Packet Exchange (IPX/SPX) do vậy không định tuyến được. Cần phải có bộ phiên dịch chuyển các gói sang TCP/IP nên hiệu suất giảm xuống

29/06/2011

Chương 6: Truyền thông với file server

8

## Netware file sharing

- Netware là “trái tim” của hệ thống Netware server chạy trên hệ điều hành Novell
- Các server dùng dịch vụ Netware directory service (NDS) để chứng thực và cấp quyền
- Các client Novell vẫn được hỗ trợ trong DOS, Windows, Macintosh và UNIX. Client định vị server dùng Novell core protocol (NCP)
- .NET không hỗ trợ giao tiếp với Netware

29/06/2011

Chương 6: Truyền thông với file server

9

## Cài đặt FTP server

- Để truy cập vào FTP server, ta cần biết địa chỉ IP và phải có username/password
- Trước tiên cần thực hiện cài đặt FTP server.
- Có thể sử dụng công cụ hỗ trợ sẵn trong Windows hoặc cài đặt các phần mềm khác như: FileZilla (địa chỉ: <http://filezilla-project.org/index.php>)

29/06/2011

Chương 6: Truyền thông với file server

10

## Cài đặt FTP server trên Windows

- Thao tác: Control Panel→Add/Remove Programs→Add or Remove Windows Components→Internet Information Services→Details→FTP Service

29/06/2011

Chương 6: Truyền thông với file server

11

## Cài đặt FTP trên Windows

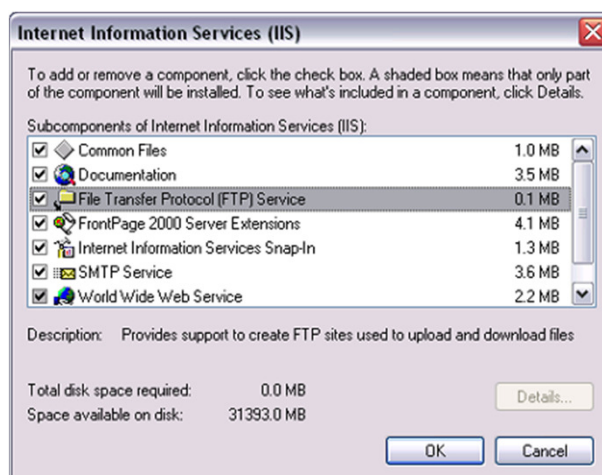
- Để quản trị và tinh chỉnh cấu hình FTP server, thực hiện: Control Panel→Administrative Tools→Internet Information Services→FTPSite→Default FTP site. Click phải và chọn Properties.
- Tiến hành chỉnh sửa các thiết lập như cho phép số kết nối đồng thời tối đa, thời gian timeout, thư mục vật lý, cho phép đăng nhập ẩn danh,...

29/06/2011

Chương 6: Truyền thông với file server

12

## Cài đặt FTP trên Windows

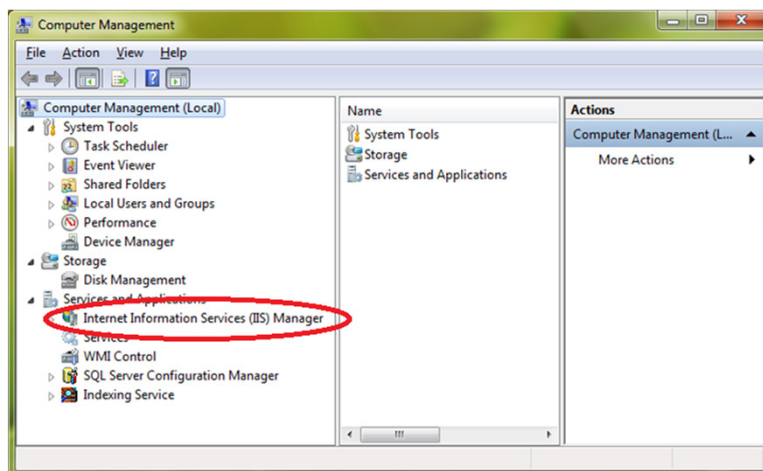


29/06/2011

Chương 6: Truyền thông với file server

13

## Cài đặt FTP trên Windows

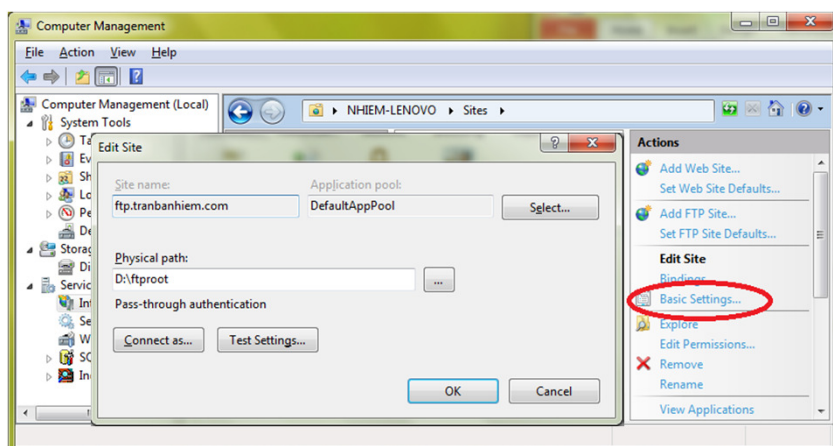


29/06/2011

Chương 6: Truyền thông với file server

14

## Cài đặt FTP trên Windows

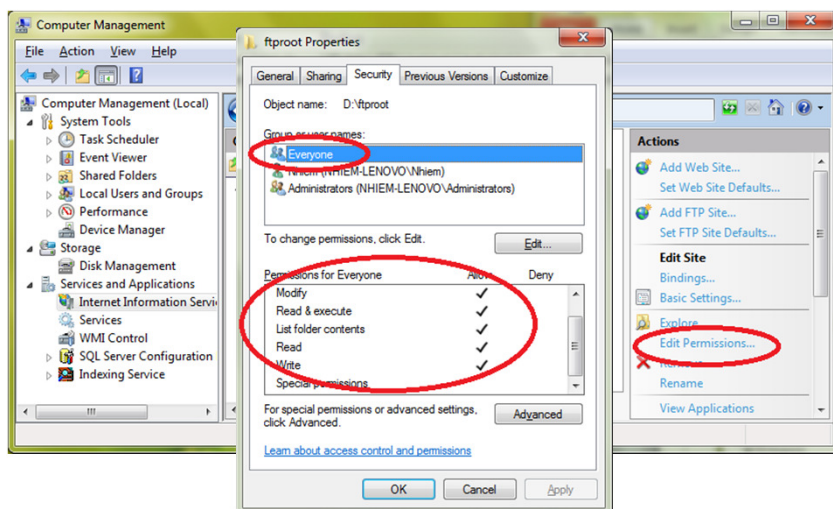


29/06/2011

Chương 6: Truyền thông với file server

15

## Cài đặt FTP trên Windows



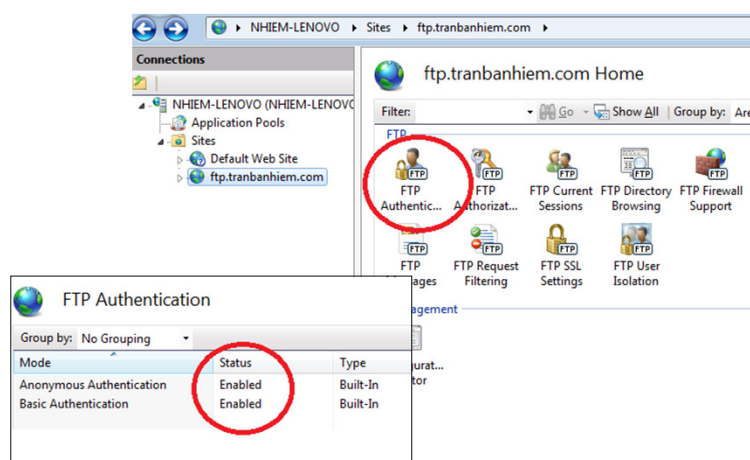
29/06/2011

Chương 6: Truyền thông với file server

16



## Cài đặt FTP trên Windows



29/06/2011

Chương 6: Truyền thông với file server

17

## Tổng quan về FTP

- FTP hoạt động trên 2 port: 21 cho socket điều khiển và 20 (hoặc cao hơn) cho socket dữ liệu.
- FTP được định nghĩa trong RFC 959
- Giống như các giao thức email, các lệnh giao dịch giữa client và server tương đối dễ hiểu và ngắn thành các dòng, tuy nhiên FTP thông qua lệnh telnet không được thuận tiện

29/06/2011

Chương 6: Truyền thông với file server

18

## Tổng quan về FTP

- Khởi tạo màn hình dòng lệnh: Start --> Run...  
gõ vào lệnh cmd.
- Cú pháp: **FTP** <IP của server>
- Sau khi kết nối thành công, xuất hiện màn hình hỏi:
  - **User:** <Nhập tên user>
  - **Pass:** <Nhập mật khẩu>
- Lúc này xuất hiện dấu nhắc dòng lệnh, gõ tiếp các lệnh theo nhu cầu, cú pháp như sau:

29/06/2011

Chương 6: Truyền thông với file server

19

## Tổng quan về FTP

Lệnh	Ý nghĩa
?	<i>Giúp đỡ</i>
ascii	chuyển chế độ truyền là ASCII
binary	chuyển chế độ truyền là nhị phân
bye	Thoát khỏi FTP (giống lệnh quit)
cd	Thay đổi thư mục trên máy chủ
close	Kết thúc kết nối đến máy tính khác
close brubeck	Đóng kết nối hiện tại với máy tính khác, nhưng vẫn giữ môi trường lệnh FTP
delete	xóa 1 file trong thư mục từ xa hiện hành (trên server)
get	Để copy 1 file từ máy từ xa về máy cục bộ
get ABC DEF	Copy file ABC trong thư mục từ xa hiện hành lên file DEF trong máy cục bộ (ghi đè).

29/06/2011

Chương 6: Truyền thông với file server

20

## Tổng quan về FTP

Lệnh	Ý nghĩa
get ABC	Copy file ABC trong thư mục từ xa hiện hành lên file ABC trong máy cục bộ (cùng tên).
help	Xem danh sách các lệnh FTP sẵn sàng
lcd	thay đổi thư mục trên máy ở xa
ls	Liệt kê tên các file trên thư mục hiện hành trong máy ở xa
mkdir	Tạo thư mục mới trong máy ở xa
mget	Copy nhiều file từ máy ở xa về máy hiện hành; sẽ được nhắc y/n cho mỗi file
mget *	Copy tất cả các file từ máy ở xa về máy hiện hành; sẽ được nhắc y/n cho mỗi file. Dùng ký tự đại diện *.
put	Copy nhiều file từ máy hiện hành lên máy ở xa; sẽ được nhắc y/n cho mỗi file
open	Mở 1 kết nối với máy tính khác

29/06/2011

Chương 6: Truyền thông với file server

21

## Tổng quan về FTP

Lệnh	Ý nghĩa
open brubeck	Mở 1 kết nối mới với máy tính, dùng chế độ brubeck;
put	Copy 1 file từ máy hiện hành lên máy ở xa
pwd	Xem thư mục hiện hành trên máy ở xa
quit	Thoát khỏi môi trường FTP (giống lệnh bye)
rmdir	Xóa thư mục hiện hành trên máy ở xa

29/06/2011

Chương 6: Truyền thông với file server

22

## Tổng quan về FTP

- FTP cũng chấp nhận file script như tham số dòng lệnh, nên có thể chạy tự động
- Ví dụ: tạo file script.ftp như sau  
open www.eej.ulst.ac.uk  
anonymous  
me@myemail.com  
cd lib  
get libtermcap.so.2.0.8  
quit

29/06/2011

Chương 6: Truyền thông với file server

23

## Tổng quan về FTP

- Để thực hiện script trên, mở dấu nhắc DOS, chuyển đến thư mục lưu file script.ftp
- Thực thi lệnh: ftp -s:script.ftp
- Kỹ thuật trên chỉ có tính minh họa để hiểu cơ chế dùng một ứng dụng FTP chứ không quan tâm đến hiệu suất truyền file

29/06/2011

Chương 6: Truyền thông với file server

24

## Mã trạng thái FTP

Mã	Ý nghĩa
1xx	Phản hồi bước đầu thành công. Lệnh đã bắt đầu thực thi trên server
2xx	Phản hồi bước đầu thành công. Lệnh đã hoàn tất trên server
3xx	Phản hồi trung gian thành công. Lệnh đã chấp nhận, nhưng chưa thực thi trên server
4xx	Phản hồi không thành công tạm thời. Lệnh đã bị từ chối, nhưng có thể được kích hoạt lại trên server
5xx	Phản hồi không thành công hoàn toàn. Lệnh đã bị từ chối, và không thể được kích hoạt lại trên server

29/06/2011

Chương 6: Truyền thông với file server

25

## Mã trạng thái FTP

- Ví dụ:  
220 Serv-U FTP-Server v2.5k for WinSock ready...  
USER secret  
331 User name okay, need password.  
PASS (hidden)  
230 User logged in, proceed.  
PWD  
257 "/" is current directory.  
TYPE A  
200 Type set to A.  
PASV  
227 Entering Passive Mode (212,17,38,3,11,144)  
LIST -aL  
150 Opening ASCII mode data connection for /bin/ls.  
226 Transfer complete.

29/06/2011

Chương 6: Truyền thông với file server

26

## Cách dùng các port

- Khi soạn email, kết thúc nội dung của email được đánh dấu bằng chuỗi ký tự <enter>.<enter>. Chuỗi này sẽ được loại bỏ khi gửi đi
- Với FTP, những chuỗi ký tự như vậy sẽ có thể nằm trong nội dung, nếu gỡ bỏ có thể làm hỏng file. Để tránh vấn đề trên, FTP dùng port 21 để gửi/nhận lệnh, mỗi lệnh kết thúc bởi <enter>

29/06/2011

Chương 6: Truyền thông với file server

27

## Cách dùng các port

- Dữ liệu với độ dài thay đổi được gửi qua lại giữa client và server trên một kết nối tạm thời ở port 20, được mở khi có dữ liệu và đóng khi kết thúc truyền dữ liệu đó
- Hầu hết FTP client đóng vai trò yêu cầu và FTP server đóng vai trò phục vụ
- FTP ở chế độ Passive: client hướng dẫn server lắng nghe trên port dữ liệu không phải port mặc định là 20

29/06/2011

Chương 6: Truyền thông với file server

28

## Cách dùng các port

- Phản hồi cho lệnh PASV sẽ luôn luôn chứa danh sách 6 số đặt trong dấu ngoặc được phân tách bởi dấu phẩy. 4 số đầu đại diện cho địa chỉ IP, 2 số cuối thể hiện port để server lắng nghe dữ liệu
- Ví dụ: 192,168,5,7,4,91 có nghĩa server có địa chỉ IP là 192.168.5.7 và đang lắng nghe trên port 1115 ( $= 4 \times 256 + 91$ )

29/06/2011

Chương 6: Truyền thông với file server

29

## Cách dùng các port

- Server sẽ thực hiện lắng nghe trên port sớm nhất có thể sau khi nhận lệnh PASV. Nó trả lời bằng thông điệp 227 để chỉ thị cho biết đang lắng nghe trên port đó
- Khi client kết nối vào port này, server trả lời bằng thông điệp 150
- Nếu client không kết nối trong khoảng thời gian quy định (khoảng vài giây), server sẽ phát thông điệp timeout 425.

29/06/2011

Chương 6: Truyền thông với file server

30

## Cách dùng các port

- Server sẽ gửi dữ liệu yêu cầu trên port đã mở và đóng kết nối lại sau khi hoàn thành gửi dữ liệu xong, phát ra thông điệp 226
- Tiến trình tương tự diễn ra theo thứ tự ngược khi upload lên server. Khi lệnh PASV được phát ra và client kết nối vào port do server chỉ định. Client sau đó đặt nội dung file lên socket mới và đóng lại sau khi gửi file xong

29/06/2011

Chương 6: Truyền thông với file server

31

## FTP: tiến trình bắt tay

- Cơ chế chứng thực: FTP chấp nhận username / password dạng text thô, nên có thể nhìn thấy được với bất kỳ người nào dùng trình phân tích
- FTP trên SSL được khuyến cáo khi website truyền những thông tin quan trọng

29/06/2011

Chương 6: Truyền thông với file server

32



## FTP: tiến trình bắt tay

- FTP server cũng cho phép truy cập ẩn danh (anonymous). Khi ấy username là anonymous và password tùy ý. Đây là thiết lập mặc định của dịch vụ Microsoft FTP
- Khi kết nối vào FTP server ở port 21, server sẽ phản hồi như sau:  
220 <some message><enter>

29/06/2011

Chương 6: Truyền thông với file server

33

## FTP: tiến trình bắt tay

- Sau đó diễn ra giống quá trình bắt tay của POP3, các lệnh USER và PASS được gửi lần lượt:  
USER <username><enter>
- Server phản hồi:  
331 <some message><enter>
- Client gửi:  
PASS <password><enter>

29/06/2011

Chương 6: Truyền thông với file server

34

## FTP: tiến trình bắt tay

- Server phản hồi nếu đăng nhập thành công:  
230 <some message><enter>
- Server phản hồi nếu đăng nhập thất bại:  
530 <some message><enter>
- Sau thời điểm này, nếu đăng nhập thành công thì người dùng có thể truy cập vào FTP server với quyền hạn tương ứng.

29/06/2011

Chương 6: Truyền thông với file server

35

## FTP: tiến trình bắt tay

- Một số FTP server sẽ hủy kết nối với những người dùng không có thao tác gì để tiết kiệm tài nguyên. Do đó, có thể dùng lệnh NOOP để báo cho server biết nhằm tránh hiện tượng trên. Diễn biến như sau:  
NOOP<enter>  
200 <message><enter>

29/06/2011

Chương 6: Truyền thông với file server

36

## FTP: tiến trình bắt tay

- Để đóng kết nối, client đơn giản chỉ cần đóng kết nối TCP bên dưới bằng cách phát ra lệnh QUIT. Diễn biến như sau:  
QUIT<enter>  
221 <message><enter>

29/06/2011

Chương 6: Truyền thông với file server

37

## Lập trình upload file

- Khai báo namespace:  
using System.IO;  
using System.Reflection;  
using System.Threading;
- Khai báo sử dụng Internet Transfer Control (ITC) – là một legacy COM. Thay vì including COM control này vào project thông qua toolbox thì chúng ta có thể gọi trực tiếp trong code

29/06/2011

Chương 6: Truyền thông với file server

38

## Lập trình upload file

- Cách gọi COM như vậy hơi phức tạp, nhưng thuận lợi hơn trong việc binding sau này (đối tượng chỉ được nạp vào khi chạy chương trình, không phải lúc biên dịch). Đồng thời có thêm khả năng chịu được lỗi nếu COM control bị xóa, ứng dụng của chúng ta vẫn làm việc, mặc dù không đáp ứng chức năng theo yêu cầu

29/06/2011

Chương 6: Truyền thông với file server

39

## Lập trình upload file

- Mỗi COM control có một programmatic ID, (hoặc ProgID) duy nhất, được lưu trong registry có dạng <project name>.<Class name>.<version>. Trong trường hợp của chúng ta, ProgID là InetCtls.Inet và không cần <version>.
- Khi chương trình thực thi, một đối tượng của class trên sẽ được khởi tạo.

29/06/2011

Chương 6: Truyền thông với file server

40

## Lập trình upload file

- Tại thời điểm thiết kế, không thể biết được các phương thức cũng như thuộc tính của đối tượng của class `InetCtls.Inet`, do đó kiểu trả về phải là object tổng quát.
- Để gọi phương thức của đối tượng chưa biết kiểu (tại thời điểm thiết kế), chúng ta phải dùng phương thức `InvokeMember`

29/06/2011

Chương 6: Truyền thông với file server

41

## Lập trình upload file

- Để kích hoạt phương thức `execute` trên đối tượng và truyền 2 tham số cho nó, ta cần định nghĩa 2 tham số đó. Tham số đầu là địa chỉ FTP server. Tham số thứ hai là lệnh FTP mà ta muốn thực thi.

29/06/2011

Chương 6: Truyền thông với file server

42

## Lập trình upload file

```
FileInfo thisFile = new FileInfo(tbFile.Text);
Type ITC;
object[] parameter= new object[2];
object ITCObject;
ITC = Type.GetTypeFromProgID("InetCtls.Inet");
ITCObject = Activator.CreateInstance(ITC);
parameter[0] = (string)tbServer.Text;
parameter[1] = (string)"PUT " + thisFile.FullName + " /" +
thisFile.Name;
ITC.InvokeMember("execute",
BindingFlags.InvokeMethod, null, ITCObject, parameter);
```

29/06/2011

Chương 6: Truyền thông với file server

43

## Lập trình upload file trên .NET

- ITC có một số giới hạn, chứa một số lỗi và hiệu suất thực thi kém. Hơn nữa nó không tự nhiên với .NET
- Chúng ta sẽ nghiên cứu một phương pháp triển khai FTP hoàn toàn trên cơ sở nền tảng của .NET

29/06/2011

Chương 6: Truyền thông với file server

44

## Lập trình FTP trên .NET

- Tạo 1 project mới có 1 form
- Tạo 4 textbox có tên: tbServer, tbUsername, tbPassword, tbStatus
- Tạo 4button có tên btnLogon, btnUpload, btnDownload, btnRoot
- Tạo 2 listbox có tên: lbFiles, lbFolders
- Thiết lập kết nối với FTP server như sau:

29/06/2011

Chương 6: Truyền thông với file server

45

## Lập trình FTP trên .NET

- Khai báo các biến public trong frmMain:  

```
public frmLogon LogonForm = new frmLogon();
public NetworkStream NetStrm;
public string RemotePath = "";
public string server = "";
```
- Đối tượng của lớp NetworkStream dùng để kết nối lệnh FTP (port 21)
- Kết nối dữ liệu không bền vững (truyền xong đóng lại ngay) nên không cần khai báo toàn cục
- Thiết lập kết nối với FTP server như sau:

29/06/2011

Chương 6: Truyền thông với file server

46

## Lập trình FTP trên .NET

```
TcpClient clientSocket = new TcpClient(tbServer.Text,
21);
NetStrm = clientSocket.GetStream();
StreamReader RdStrm = new
StreamReader(MainForm.NetStrm);
tbStatus.Text = RdStrm.ReadLine();
tbStatus.Text = sendFTPcmd("USER " +
tbUsername.Text + STR_Constant);
tbStatus.Text = sendFTPcmd("PASS " +
tbPassword.Text + STR_Constant);
if (tbStatus.Text.Substring(0, 3) != "230")
    MessageBox.Show("Failed to log in");
```

29/06/2011

Chương 6: Truyền thông với file server

47

## Lập trình FTP trên .NET

- Phương pháp cơ bản là tạo một đối tượng NetStrm thuộc lớp NetworkStream để mở ra socket. Các lệnh FTP (dưới dạng string) được gửi qua socket này
- Một đối tượng của lớp StreamReader được tạo ra nhằm đọc nội dung gửi trên stream này.

29/06/2011

Chương 6: Truyền thông với file server

48



## Lập trình FTP trên .NET

```
public string sendFTPcmd(string cmd)
{
    byte[] szData;
    string returnedData = "";
    StreamReader RdStrm = new StreamReader(NetStrm);
    szData = Encoding.ASCII.GetBytes(cmd.ToCharArray());
    NetStrm.Write(szData, 0, szData.Length);
    tbStatus.Text += "\r\nSent:" + cmd;
    returnedData = RdStrm.ReadLine();
    tbStatus.Text += "\r\nRcvd:" + returnedData;
    return returnedData;
}
```

29/06/2011

Chương 6: Truyền thông với file server

49

## Lập trình FTP trên .NET

```
public string Upload( )
{
    openFileDialog.ShowDialog();
    NetworkStream passiveConnection;
    FileInfo fileParse = new
    FileInfo(openFileDialog.FileName);
    FileStream fs = new
    FileStream(openFileDialog.FileName, FileMode.Open);
    byte[] fileData = new byte[fs.Length];
    fs.Read(fileData, 0, (int)fs.Length);
    passiveConnection = createPassiveConnection();
```

29/06/2011

Chương 6: Truyền thông với file server

50

## Lập trình FTP trên .NET

```

string cmd = "STOR " + fileParse.Name + "\r\n";
tbStatus.Text += "\r\nSent:" + cmd;
string response = sendFTPcmd(cmd);
tbStatus.Text += "\r\nRcvd:" + response;
passiveConnection.Write(fileData, 0, (int)fs.Length);
passiveConnection.Close();
MessageBox.Show("Uploaded");
tbStatus.Text += "\r\nRcvd:" + new
StreamReader(NetStrm).ReadLine();
}

```

29/06/2011

Chương 6: Truyền thông với file server

51

## Lập trình FTP trên .NET

```

public string Download( )
{
    byte[] fileData;
    saveFileDialog.ShowDialog();
    fileData = sendPassiveFTPcmd("RETR " +
    lbFiles.SelectedItem.ToString() + "\r\n");
    FileStream fs = new FileStream(saveFileDialog.FileName,
    FileMode.CreateNew);
    fs.Write(fileData, 0, fileData.Length);
    fs.Close();
    MessageBox.Show("Downloaded");
}

```

29/06/2011

Chương 6: Truyền thông với file server

52

## Lập trình FTP trên .NET

```

byte[] szData;
System.Collections.ArrayList al = new ArrayList();
byte[] RecvBytes = new byte[Byte.MaxValue];
Int32 bytes;
Int32 totalLength = 0;
szData =
System.Text.Encoding.ASCII.GetBytes(cmd.ToCharArray());
NetworkStream passiveConnection;
passiveConnection = createPassiveConnection();
tbStatus.Text += "\r\nSent:" + cmd;
StreamReader commandStream = new
StreamReader(NetStrm);
NetStrm.Write(szData, 0, szData.Length);
}

```

29/06/2011

Chương 6: Truyền thông với file server

53

## Lập trình FTP trên .NET

```

public byte[] sendPassiveFTPcmd(string cmd)
{
    byte[] szData;
    System.Collections.ArrayList al = new ArrayList();
    byte[] RecvBytes = new byte[Byte.MaxValue];
    Int32 bytes;
    Int32 totalLength = 0;
    szData =
    System.Text.Encoding.ASCII.GetBytes(cmd.ToCharArray());
    NetworkStream passiveConnection;
    passiveConnection = createPassiveConnection();
    tbStatus.Text += "\r\nSent:" + cmd;
    StreamReader commandStream = new StreamReader(NetStrm);
    NetStrm.Write(szData, 0, szData.Length);
}

```

29/06/2011

Chương 6: Truyền thông với file server

54

## Lập trình FTP trên .NET

```

while (true)
{
    bytes = passiveConnection.Read(RecvBytes,
    0, RecvBytes.Length);
    if (bytes <= 0) break;
    totalLength += bytes;
    al.AddRange(RecvBytes);
}
al = al.GetRange(0, totalLength);
tbStatus.Text += "\r\nRcvd:" + commandStream.ReadLine(); // 125
tbStatus.Text += "\r\nRcvd:" + commandStream.ReadLine(); // 226
return (byte[])al.ToArray((new byte()).GetType());
}

```

29/06/2011

Chương 6: Truyền thông với file server

55

## Lập trình FTP trên .NET

```

private NetworkStream createPassiveConnection()
{
    string[] commaSeperatedValues;
    int highByte = 0;
    int lowByte = 0;
    int passivePort = 0;
    string response = "";
    response = sendFTPCmd("PASV\r\n");
    // 227 Entering Passive Mode (127,0,0,1,4,147).
    commaSeperatedValues = response.Split(",").ToArray();
    highByte = Convert.ToInt16(commaSeperatedValues[4]) * 256;
}

```

29/06/2011

Chương 6: Truyền thông với file server

56

## Lập trình FTP trên .NET

```

commaSeperatedValues[5] =
commaSeperatedValues[5].Substring(0,
commaSeperatedValues[5].IndexOf(""));
lowByte = Convert.ToInt16(
commaSeperatedValues[5]);
passivePort = lowByte + highByte;
TcpClient clientSocket = new TcpClient(server,
passivePort);
NetworkStream pasvStrm = clientSocket.GetStream();
return pasvStrm;
}

```

29/06/2011

Chương 6: Truyền thông với file server

57

## Lập trình FTP trên .NET

```

private void btnRoot_Click(object sender,
EventArgs e)
{
    RemotePath = "/";
    sendFTPcmd("CWD ^r\n");
    getRemoteFolders();
}

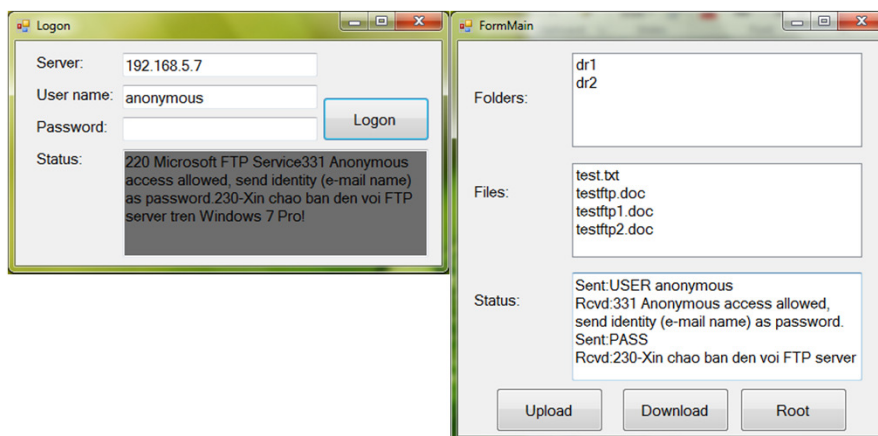
```

29/06/2011

Chương 6: Truyền thông với file server

58

## Lập trình FTP trên .NET



29/06/2011

Chương 6: Truyền thông với file server

59

## Hỗ trợ FTP trên .NET 2.0

- .NET 2.0 hỗ trợ FTP thông qua class WebClient, FtpWebRequest
- Cho phép lập trình ở mức socket hoặc thông qua các đối tượng COM

29/06/2011

Chương 6: Truyền thông với file server

60

## Hỗ trợ FTP trong WebClient

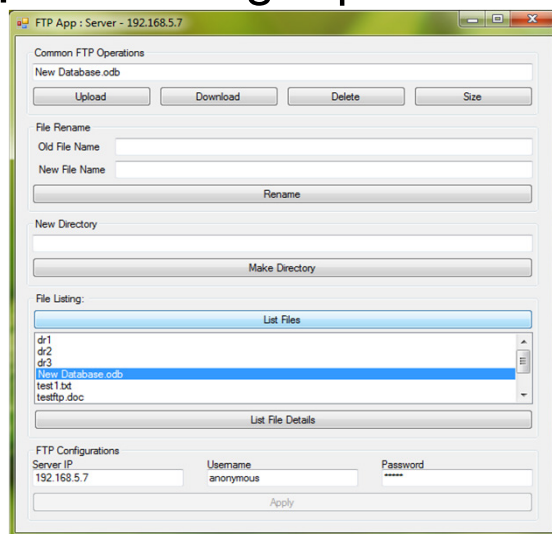
```
public void downloadFile()
{
    string filename = "ftp://ms.com/files/dotnetfx.exe";
    WebClient client = new WebClient();
    client.DownloadFile(filename,"dotnetfx.exe");
}
```

29/06/2011

Chương 6: Truyền thông với file server

61

## Hỗ trợ FTP trong FtpWebRequest



29/06/2011

Chương 6: Truyền thông với file server

62

## Hỗ trợ FTP trong FtpWebRequest

```
private void Upload(string filename)
{
    FileInfo fileInf = new FileInfo(filename);
    string uri = "ftp://" + ftpServerIP + "/" + fileInf.Name;
    FtpWebRequest reqFTP;
    reqFTP = (FtpWebRequest)FtpWebRequest.Create(new
    Uri("ftp://" + ftpServerIP + "/" + fileInf.Name));
    reqFTP.Credentials = new NetworkCredential(ftpUserID,
    ftpPassword);
    reqFTP.KeepAlive = false;
    reqFTP.Method = WebRequestMethods.Ftp.UploadFile;
    reqFTP.UseBinary = true;
    reqFTP.ContentLength = fileInf.Length;
    int buffLength = 2048;
    byte[] buff = new byte[buffLength];
```

29/06/2011

Chương 6: Truyền thông với file server

63

## Hỗ trợ FTP trong FtpWebRequest

```
int contentLen;
FileStream fs = fileInf.OpenRead();
try {
    Stream strm = reqFTP.GetRequestStream();
    contentLen = fs.Read(buff, 0, buffLength);
    while (contentLen != 0) {
        strm.Write(buff, 0, contentLen);
        contentLen = fs.Read(buff, 0, buffLength);
    }
    strm.Close();
    fs.Close();
}
catch(Exception ex) {
    MessageBox.Show(ex.Message, "Upload Error");
}
}
```

29/06/2011

Chương 6: Truyền thông với file server

64



## Hỗ trợ FTP trong FtpWebRequest

```
private void Download(string filePath, string fileName)
{
    FtpWebRequest reqFTP;
    try
    {
        FileStream outputStream = new FileStream(filePath + "\\\" + fileName,
        FileMode.Create);
        reqFTP = (FtpWebRequest)FtpWebRequest.Create(new Uri("ftp://" +
        ftpServerIP + "/" + fileName));
        reqFTP.Method = WebRequestMethods.Ftp.DownloadFile;
        reqFTP.UseBinary = true;
        reqFTP.Credentials = new NetworkCredential(ftpUserID, ftpPassword);
        FtpWebResponse response = (FtpWebResponse)reqFTP.GetResponse();
        Stream ftpStream = response.GetResponseStream();
        long cl = response.ContentLength;
```

29/06/2011

Chương 6: Truyền thông với file server

65

## Hỗ trợ FTP trong FtpWebRequest

```
int bufferSize = 2048;
int readCount;
byte[] buffer = new byte[bufferSize];
readCount = ftpStream.Read(buffer, 0, bufferSize);
while (readCount > 0) {
    outputStream.Write(buffer, 0, readCount);
    readCount = ftpStream.Read(buffer, 0, bufferSize);
}
ftpStream.Close();
outputStream.Close();
response.Close();
}
catch (Exception ex) {
    MessageBox.Show(ex.Message);
}
}
```

29/06/2011

Chương 6: Truyền thông với file server

66

## Hỗ trợ FTP trong FtpWebRequest

```
public void DeleteFTP(string fileName)
{
    try
    {
        string uri = "ftp://" + ftpServerIP + "/" + fileName;
        FtpWebRequest reqFTP;
        reqFTP = (FtpWebRequest)FtpWebRequest.Create(
            new Uri("ftp://" + ftpServerIP + "/" + fileName));
        reqFTP.Credentials = new NetworkCredential(ftpUserID,
            ftpPassword);
        reqFTP.KeepAlive = false;
        reqFTP.Method = WebRequestMethods.Ftp.DeleteFile;
        string result = String.Empty;
    }
}
```

29/06/2011

Chương 6: Truyền thông với file server

67

## Hỗ trợ FTP trong FtpWebRequest

```
FtpWebResponse response =
(FtpWebResponse)reqFTP.GetResponse();
long size = response.ContentLength;
Stream datastream = response.GetResponseStream();
StreamReader sr = new StreamReader(datastream);
result = sr.ReadToEnd();
sr.Close();
datastream.Close();
response.Close();
}
catch (Exception ex) {
    MessageBox.Show(ex.Message, "FTP 2.0 Delete");
}
}
```

29/06/2011

Chương 6: Truyền thông với file server

68

## Hỗ trợ FTP trong FtpWebRequest

```
private void btnFileSize_Click(object sender,
EventArgs e)
{
    long size = GetFileSize(txtUpload.Text.Trim());
    MessageBox.Show(size.ToString() + " bytes");
}
```

29/06/2011

Chương 6: Truyền thông với file server

69

## Hỗ trợ FTP trong FtpWebRequest

```
private string[] GetFilesDetailList()
{
    string[] downloadFiles;
    try {
        StringBuilder result = new StringBuilder();
        FtpWebRequest ftp;
        ftp = (FtpWebRequest)FtpWebRequest.Create(new
Uri("ftp://" + ftpServerIP + "/"));
        ftp.Credentials = new NetworkCredential(ftpUserID,
ftpPassword);
        ftp.Method =
WebRequestMethods.Ftp.ListDirectoryDetails;
```

29/06/2011

Chương 6: Truyền thông với file server

70

## Hỗ trợ FTP trong FtpWebRequest

```

WebResponse response = ftp.GetResponse();
StreamReader reader = new
StreamReader(response.GetResponseStream());
    string line = reader.ReadLine();
while (line != null)    {
    result.Append(line);
    result.Append("\n");
    line = reader.ReadLine();
}
result.Remove(result.ToString().LastIndexOf("\n"),
1);

```

29/06/2011

Chương 6: Truyền thông với file server

71

## Hỗ trợ FTP trong FtpWebRequest

```

        reader.Close();
        response.Close();
        return result.ToString().Split('\n');
    }
    catch (Exception ex) {
        System.Windows.Forms.MessageBox.Show(ex.Me
ssage);
        downloadFiles = null;
        return downloadFiles;
    }
}

```

29/06/2011

Chương 6: Truyền thông với file server

72

## Hỗ trợ FTP trong FtpWebRequest

```
public string[] GetFileList()
{
    string[] downloadFiles;
    StringBuilder result = new StringBuilder();
    FtpWebRequest reqFTP;
    try {
        reqFTP = (FtpWebRequest)
FtpWebRequest.Create(new Uri("ftp://" + ftpServerIP +
"/"));
        reqFTP.UseBinary = true;
        reqFTP.Credentials = new NetworkCredential(
ftpUserID, ftpPassword);
```

29/06/2011

Chương 6: Truyền thông với file server

73

## Hỗ trợ FTP trong FtpWebRequest

```
        reqFTP.Method =
WebRequestMethods.Ftp.ListDirectory;
        WebResponse response = reqFTP.GetResponse();
        StreamReader reader = new StreamReader(
response.GetResponseStream());
        string line = reader.ReadLine();
        while (line != null) {
            result.Append(line);
            result.Append("\n");
            line = reader.ReadLine();
        }
```

29/06/2011

Chương 6: Truyền thông với file server

74

## Hỗ trợ FTP trong FtpWebRequest

```

        result.Remove(result.ToString().LastIndexOf('\n'), 1);
        reader.Close();
        response.Close();
        return result.ToString().Split('\n');
    }
    catch (Exception ex) {
        System.Windows.Forms.MessageBox.Show(ex.Message);
        downloadFiles = null;
        return downloadFiles;
    }
}

```

29/06/2011

Chương 6: Truyền thông với file server

75

## Hỗ trợ FTP trong FtpWebRequest

```

private void Rename(string currentFilename, string
newFilename)
{
    FtpWebRequest reqFTP;
    try {
        reqFTP =
(FtpWebRequest)FtpWebRequest.Create(new Uri("ftp://"
+ ftpServerIP + "/" + currentFilename));
        reqFTP.Method =
WebRequestMethods.Ftp.Rename;
        reqFTP.RenameTo = newFilename;
        reqFTP.UseBinary = true;
    }
    catch { }
}

```

29/06/2011

Chương 6: Truyền thông với file server

76

## Hỗ trợ FTP trong FtpWebRequest

```

        reqFTP.Credentials = new NetworkCredential(ftpUserID,
ftpPassword);
        FtpWebResponse response =
(FtpWebResponse)reqFTP.GetResponse();
        Stream ftpStream = response.GetResponseStream();
        ftpStream.Close();
        response.Close();
    }
    catch (Exception ex) {
        MessageBox.Show(ex.Message);
    }
}

```

29/06/2011

Chương 6: Truyền thông với file server

77

## Hỗ trợ FTP trong FtpWebRequest

```

private void MakeDir(string dirName)
{
    FtpWebRequest reqFTP;
    try
    {
        reqFTP = (FtpWebRequest)FtpWebRequest.Create(new
Uri("ftp://" + ftpServerIP + "/" + dirName));
        reqFTP.Method =
WebRequestMethods.Ftp.MakeDirectory;
        reqFTP.UseBinary = true;
        reqFTP.Credentials = new NetworkCredential(ftpUserID,
ftpPassword);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

```

29/06/2011

Chương 6: Truyền thông với file server

78

## Hỗ trợ FTP trong FtpWebRequest

```
FtpWebResponse response =  
(FtpWebResponse)reqFTP.GetResponse();  
Stream ftpStream =  
response.GetResponseStream();  
ftpStream.Close();  
response.Close();  
}  
catch (Exception ex)  
{  
    MessageBox.Show(ex.Message);  
}  
}
```

29/06/2011

Chương 6: Truyền thông với file server

79

## Bài tập

- Cài đặt các chương trình đã minh họa trong bài giảng của chương bằng ngôn ngữ C# hoặc VB.NET

29/06/2011

Chương 6: Truyền thông với file server

80