

# 《网络编程技术课程设计》论文

2012 — 2013 学年第二学期

题    目：    **FTP 客户端设计与实现**

专业班级：    **软件工程 Java10-06 班**

组    员：\_\_\_\_\_

指导老师：\_\_\_\_\_

日    期：    **2013-07-04**

# 目录

摘要.....	2
关键词.....	2
1. 引言.....	2
2.FTP协议分析.....	2
2.1FTP 工作原理.....	2
2.2FTP 工作模式.....	4
2.3FTP 传输模式.....	6
2.3.1ASCII 传输方式.....	6
2.3.2二进制传输方式.....	6
3.FTP客户端的总体设计.....	6
3.1设计目的及工具的选择.....	6
3.1.1设计目的.....	6
3.1.2设计工具.....	7
3.1.3FtpClient 类.....	7
3.2客户端主要模块设计.....	7
3.2.1客户端功能模块.....	8
3.2.2客户端界面设计.....	9
3.3数据流程图.....	9
3.3.1客户端连接流程图.....	9
3.3.2上传子模块流程图.....	11
3.3.3下载子模块流程图.....	12
4.FTP客户端程序实现.....	14
4.1客户端显示界面实现.....	14
4.1.1数据输入界面详细设计.....	14
4.1.2服务器文件信息显示界面详细设计.....	14
4.1.3上下载队列显示界面和信息显示界面详细设计.....	15
4.1.4操作按钮模块详细设计.....	15
4.2功能模块实现.....	16
4.2.1连接服务器.....	16
4.2.2上传文件.....	16
4.2.3下载文件.....	17
5. 功能运行测试.....	17
6. 结语.....	20
参考文献.....	21
附录.....	22

## FTP客户端的设计与实现

**摘要：**FTP是Internet 上用来传送文件的协议。它是为了我们能够在Internet 上互相传送文件而制定的文件传送标准,规定了Internet 上文件如何传送。通过FTP协议,我们就可以跟Internet 上的FTP服务器进行文件的上传或下载。本文以实现一个简单易用的FTP客户端为目标,通过分析FTP协议的基本工作原理和FTP的数据传输原理,研究如何使用Java 工具对FTP客户端进行设计,选择Java 类库中的FtpClient 类来实现FTP客户端程序的上传下载等主要功能,并用eclipse 软件提供的一个支持可视化编程的开发环境,从而实现了FTP客户端的开发。

**关键词：**FTP协议 ,工作原理, FtpClient, 上传下载

### 引言

目前,FTP服务已经是网络上广泛使用的一种服务了,它是建立在FTP协议上的用于网络环境下进行文件传输的一种文件传输系统,几乎可以传输各种类型的文件,如文本文件,声音文件,图像文件等。一个文件传输系统有服务器端程序和客户端程序共同组成的,客户端程序提供用户操作的界面并完成文件的上传和下载。在网络应用中,FTP有着非常重要的地位,在Internet 中一个十分重要的就是资源,而各种各样的资源大多数都是放在FTP服务器中的,为了获得这些资源,我们必须使用FTP客户端软件。目前流行的有cuteFTP,WS\_FTP ByteCatcher,FTPVoyage等。为了方便用户的使用,本文将介绍一种基于Java的FTP客户端程序的设计与实现。

本文首先分析了FTP协议,FTP的工作原理等,然后介绍了如何使用Java工具来设计和实现一个简单的FTP客户端。使用本FTP客户端,可以使用户轻松的下载、上传网络资源。

## 2.FTP 协议分析

### 2.1FTP 的工作原理

FTP (FileTransferProtocol )，是文件传输协议的简称。用于 Internet 上的控制文件的双向传输。同时，它也是一个应用程序。用户可以通过它把自己机器与世界各地所有运行 FTP协议的服务器相连，访问服务器上的资源和信息。

FTP协议在 TCP/IP 协议栈中的位置如表 1:

表 1 TCP/IP 协议栈

HTTP FTP TELN SMTP DNS TFTP NMP	应用层
TCP UDP	传输层
IP	互联网络层
X25 ISDN LAN WLAN FDDI ATM	网络接口层

当启动 FTP从远程计算机拷贝文件时，事实上启动了两个程序：一个本地机器上的 FTP客户端程序，它向 FTP服务器提出拷贝文件的请求。另一个是启动在远程计算机的上的 FTP服务器程序，它响应请求把你指定的文件传送到你的计算机中。FTP采用“客户端/服务器”方式，用户要在自己的本地计算机上安装 FTP客户端程序。

从根本上说，FTP协议就是在网络中各种不同的计算机之间按照 TCP/IP 协议来传输文件。FTP 协议采用客户端/服务器(Client/Sever) 模式，由 FTP 客户端程序和 FTP服务器端程序组成。使用时，先启动 FTP客户端程序与远程主机建立连接，然后向远程主机发出传输命令，远程主机在收到命令后就给予响应，并执行正确的命令。但是 FTP有一个根本的限制，那就是，如果用户在某个主机上没有注册获得授权，即没有用户名和口令，就不能与该主机进行文件传输。但匿名 FTP服务器除外，它能够使用户与远程主机建立连接并以匿名身份从远程主机上拷贝文件，而不必是该远程主机的注册用户。用户使用特殊的用户名“Anonymoÿs或“guest ”就可有限制地访问远程主机上公开的文件。现在许多系统要求用户将 Email 地址作为口令，以便更好地对访问者进行跟踪。出于安全的目的，大部分匿名 FTP主机一般只允许远程用户下载文件，而不允许上载文件。也就是说，用户只能从匿名 FTP主机拷贝其需要的文件而不能把文件拷贝到匿名

FTP主机。另外，匿名 FTP主机还采用了其他一些保护措施以保护自己的文件不至于被用户修改和删除，并防止计算机病毒的侵入。匿名 FTP一直是 Internet 上获取信息资源的最主要方式，在 Internet 成千上万的匿名 FTP主机中存储着无以计数的文件，这些文件包含了各种各样的信息、数据和软件。人们只要知道特定信息资源的主机地址，就可以用匿名 FTP登陆。

## 2.2 FTP 工作模式

FTP采用 C/S（客户端/服务器）模式进行通信，但与其他 C/S 模式网络通信协议又有一个很大的区别。通常在进行 HTTP通信或是 TELNET通信的时候，只需要一个端口进行通信。即客户端只需要连接一个端口进行数据通信。如 TELNET的默认端口是 23，用户从头到尾都只需使用这个端口。但是 FTP通信除了有一个默认端口 21 之外，还需要其他的端口。其中默认端口（21）主要进行控制连接，进行命令协议和服务器端的响应码的传输；另外一个非标准端口主要进行数据传递，如文件的上载、下载等。至于非标准端口的产生则要根据用户选择的连接模式而定：如果客户选择的是主动模式（PORT），则需要用户端提供给服务器一个 IP 地址和一个非标准端口；而如果用户采用被动模式（PASV），则服务器端需要提供给客户端一个 IP 地址和一个非标准端口。

下表是几个著名的 TCP端口号：

表 2 TCP 端口号

TCP端口号	关键字	描述
20	FTP-DATA	文件传输协议数据
21	FTP	文件传输协议控制
23	TELENET	远程登录协议
25	SMTP	简单邮件传输协议
53	DOMAIN	域名服务器
80	HTTP	超文本传输协议
110	POP3	邮局协议
119	NNTP	新闻传送协议



FTP中字节大小有两个：逻辑字节大小和用于传输的字节大小。后者通常是8位，而前者可不一定是多少了。传输字节不必等于逻辑字节大小，也不必对数据结构进行解释。控制连接是建立在 USER-PI和 SERVER-PI之间用于交换命令与应答的通信链路。数据连接是传输数据的全双工连接，传输数据可以发生在服务器 DTP和用户 DTP之间也可以发生在两个服务器 DTP之间，FTP可以传输非连续的文件，这些文件的一部分称为页。服务器 DTP代表一种传输过程，它通常处于“主动”状态，它和侦听端口建立数据连接，它还可以为传输和存储设置参数，并根据 PI 的指令传输数据。当然，DTP也可以转入“被动”状态。服务器 FTP进程，它是和用户 FTP进程一起工作的，它由 PI 和 DTP组成。至于用户 FTP进程则是由 PI，DTP和用户接口组成的。

具体工作模式如图 1 所示：

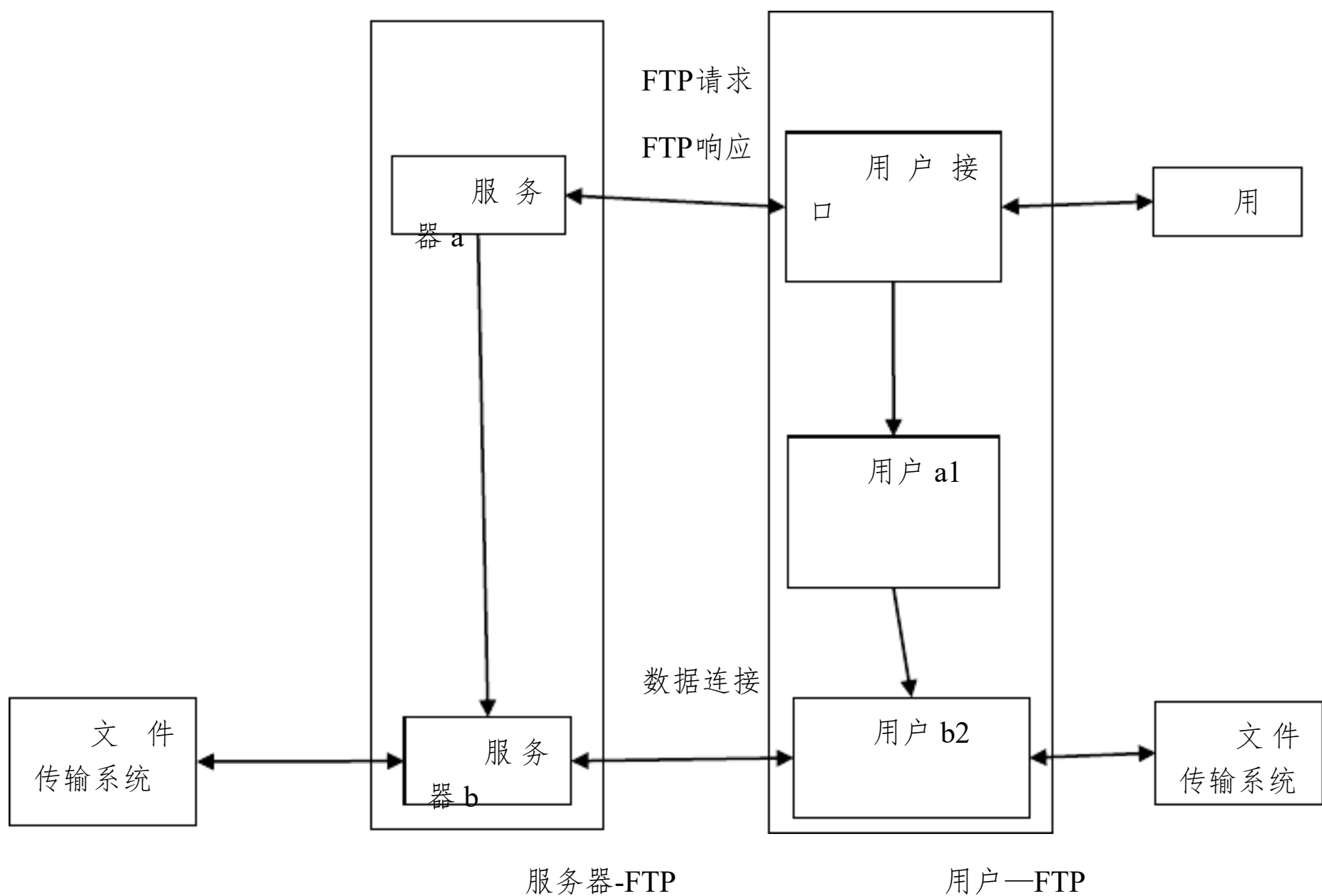


图 1 FTP工作模式图

上图中用户 PI 开始控制连接，控制连接与 Telnet 协议很象。在开始阶段，标准 FTP 命令由用户 PI 产生并通过控制连接传送到服务器进程。服务器 PI 向用户 PI 返回标准应答。FTP 命令指定数据连接参数和文件系统操作。用户 DTP 在特定数据端口侦听，服务器开始数据连接并以指定的参数开始数据传输。数据端口不必在开始 FTP 命令的机器上，但用户或用户 FTP 进程必须确定它在指定的数据端口上侦听。这个数据连接是全双工的。

### 2.3 FTP 传输模式

FTP 协议的任务是从一台计算机将文件传送到另一台计算机，它与这两台计算机所处的位置、联接的方式、甚至是是否使用相同的操作系统无关。假设两台计算机通过 ftp 协议对话，并且能访问 Internet，你可以用 ftp 命令来传输文件。每种操作系统使用上有某一些细微差别，但是每种协议基本的命令结构是相同的。FTP 的传输有两种方式：ASCII 传输模式和二进制数据传输模式。

#### 2.3.1 ASCII 传输方式

假定用户正在拷贝的文件包含的简单 ASCII 码文本，如果在远程机器上运行的不是 UNIX 当文件传输时 ftp 通常会自动地调整文件的内容以便于把文件解释成另外那台计算机存储文本文件的格式。但是常常有这样的情况，用户正在传输的文件包含的不是文本文件，它们可能是程序，数据库，字处理文件或者压缩文件（尽管字处理文件包含的大部分是文本，其中也包含有指示页尺寸，字库等信息的非打印字符）。在拷贝任何非文本文件之前，用 binary 命令告诉 ftp 逐字拷贝，不要对这些文件进行处理。

#### 2.3.2 二进制传输方式

在二进制传输中，保存文件的位序，以便原始和拷贝的是逐位一一对应的。即使目的地机器上包含位序列的文件是没意义的。例如，macintosh 以二进制方式传送可执行文件到 Windows 系统，在对方系统上，此文件不能执行。如果你在 ASCII 方式下传输二进制文件，即使不需要也仍会转译。这会使传输稍微变慢，也会损坏数据，使文件变得不能用。如果你知道这两台机器是同样的，则二进制方式对文本文件和数据文件都是有效的。

## 3 FTP 客户端总体设计

### 3.1 设计目的及工具选

#### 3.1.1 设计目的

FTP 早已应用在网络各个方面，比如我们在访问网络上的资源时，经常会遇到需要连接 FTP 服务器来下载，或者你制作了一个网站，想要将其上传至服务器，让大家能够访问它，这个时候就会用到 FTP 客户端。虽然平时使用 windows 自带的 IE 浏览器即可实现对 FTP 服务器的访问，但是 IE 只是个很粗糙的 FTP 客户端。IE 浏览器在 6.0 以下的版本不支持 PASV 方式访问服务器，而且 IE 浏览器在登录 FTP 服务器的时候，无法查看返回的信息，在出错的时候，无法找到错误的原因，因此在使用上很不方便。现在大多数的 FTP 客户端软件都是收费版的，而且以英文版居多，使用上操作很复杂，目前流行的有 CoreFtp、AceFtp、FtpVoyager 等，但是大多数用户并不习惯使用这些软件。本文所设计的 FTP 客户端则适合大多数用户使用，界面设计的很简单，使用户可以一目了然，而且具备 FTP 客户端的基本功能，可以作为下载、上传文件的工具来使用。

### 3.1.2 设计工具

程序设计中用到的主要工具就是 Java，Java 是 SUN 公司开发的一种面向对象的新一代网络编程语言，它可以在各种不同的机器、操作系统的网络环境中进行开发，具有解释型语言（如 Basic 语言）和编译型语言（如 C 语言）的特性。Java 摒弃了 C++ 中各种弊大于利的功能和许多很少用到的功能，用 Java 开发的程序可以在网络上传输，并运行于任何客户机上。使用 Java 语言可以方便的设计出网络应用类的程序，而且 Java 的移植性很好，在、其应用程序在各种机器上都能无差别的执行为了使 FTP 客户端能适应大多数平台下的使用，我选择使用 Java 语言来实现。

### 3.1.3 ftpclient 类

Java 类库的存在使它适合于一个不断发展的环境。在类库中我们可以自由地加入新的方法和实例变量而不会影响用户程序的执行。并且 Java 通过接口来支持多重继承，使其比严格的类继承具有更灵活的方式和扩展性。在处理复杂事物的时候，用到的一种基本手段就是抽象。抽象的目的就是区别事物之间的本质和不同，面向对象编程（OOB）的实质就是利用类和对象来建立抽象模型。类表示了对象的类别，是创建对象的蓝本。建立一个事物的抽象模型，本质上就是表达该事物的性质和行为。使用类来建立抽象模型，是通过在类中定义变量和方法来实现的。类中定义的属性是一种可存储值的变量，该变量代表着事物的具体性质。



在 J2SE 中提供了与客户端编程有关的 Api，它就是 `sun.net.ftp.FtpClient` 类，`FtpClient` 类封装了 FTP 协议的相关指令与实现细节，提供了一系列的方法，这些方法涵盖了几乎所有 ftp 的功能使用 `FtpClient` 类设计 FTP 客户端程序可以让编程人员轻松地实现登录远程 FTP 服务器、浏览服务器上的目录、设置传输协议、上传文件和下载文件等功能。

本文所使用的 `FtpClient` 类的构造方法和主要方法如下：

### 1.PublicFtpClient(Stringhostname)

此构造函数利用给出的主机名建立一条 FTP 连接，使用默认端口号。

### 2.PublicvoidopenServer(Stringhostname)

这个方法用于建立一条与指定主机上的 FTP 服务器的连接，使用默认端口号。

### 3.Publicvoidlogin(Stringusername, Stringpassword)

此方法利用参数 `username` 和 `password` 登录到 FTP 服务器。使用过 Internet 的用户应该知道，匿名 FTP 服务器的登录用户名为 `anonymous`，密码一般用自己的电子邮件地址。`Publicvoidbinary()`：该命令可把传输格式设置为二进制格式。

#### 3.2.1 客户端功能模块

FTP 客户端作为用户连接 FTP 服务器的工具，需要具备一些基本功能模块，如连接、上传、下载等，其中最主要的三大功能模块如下：

1、下载功能模块：连接用户指定的 FTP 服务器，获取服务器目录下的文件列表，

当用户选择项目为文件时，点击下载可以下载用户所需要的文件。选定文件后提示用户将文件存储在何处，用户选择存储区后开始下载任务。依靠 FTP 协议的支持，用户下载的文件没有类型和大小限制，即用户可已选择任何自己想要下载的文件，通过 FTP 客户端进行下载。

2、上传功能模块：用户可以自行选择本机上存储的文件，上传至 FTP 服务器。

在用户选择存储在服务器的位置后，点击上传，选择所需要上传的文件，即可将文件上传至服务器。

3、连接功能模块：将用户本机与指定的 IP 地址连接，验证用户名和密码，在

FTP服务器上注册。

有了这三个基本的功能模块，客户端的基本应用功能即可实现。

### 3.2.2 客户端界面设计

为了使用户能够方便的使用 FTP客户端，在设计中我将数据输入和信息显示整合在一个页面中，全部客户端仅包含一个主要的窗口，窗口中分别有以下五个界面：

1、数据输入界面：用来让用户输入服务器地址，用户名，密码。

2、服务器文件信息显示界面：主要用来显示服务器下存储的文件，方便用户清

楚的了解服务器中存储了哪些文件。

3、上下载队列显示界面：显示正在下载或上传的文件线程。

4、操作按钮界面：用户可以通过点击上传、下载、断开、关闭等按钮来实现客

户端的具体操作。

5、信息显示界面：显示当前连接信息和上传下载是否成功信息。这五个界面可以使用 eclipse 工具下的 swing 控件来实现，在布局时，将界面内的各个元素整合在一个框架中。

设计的主界面如图 2：



图 2 客户端界面

### 3.3 数据流程图

#### 3.3.1 客户端连接流程图

---

运行客户端后，首先是连接服务器，需要输入服务器 IP 地址及用户名和密码点击连接按钮后开始连接服务器。

下图为客户端连接流程图。

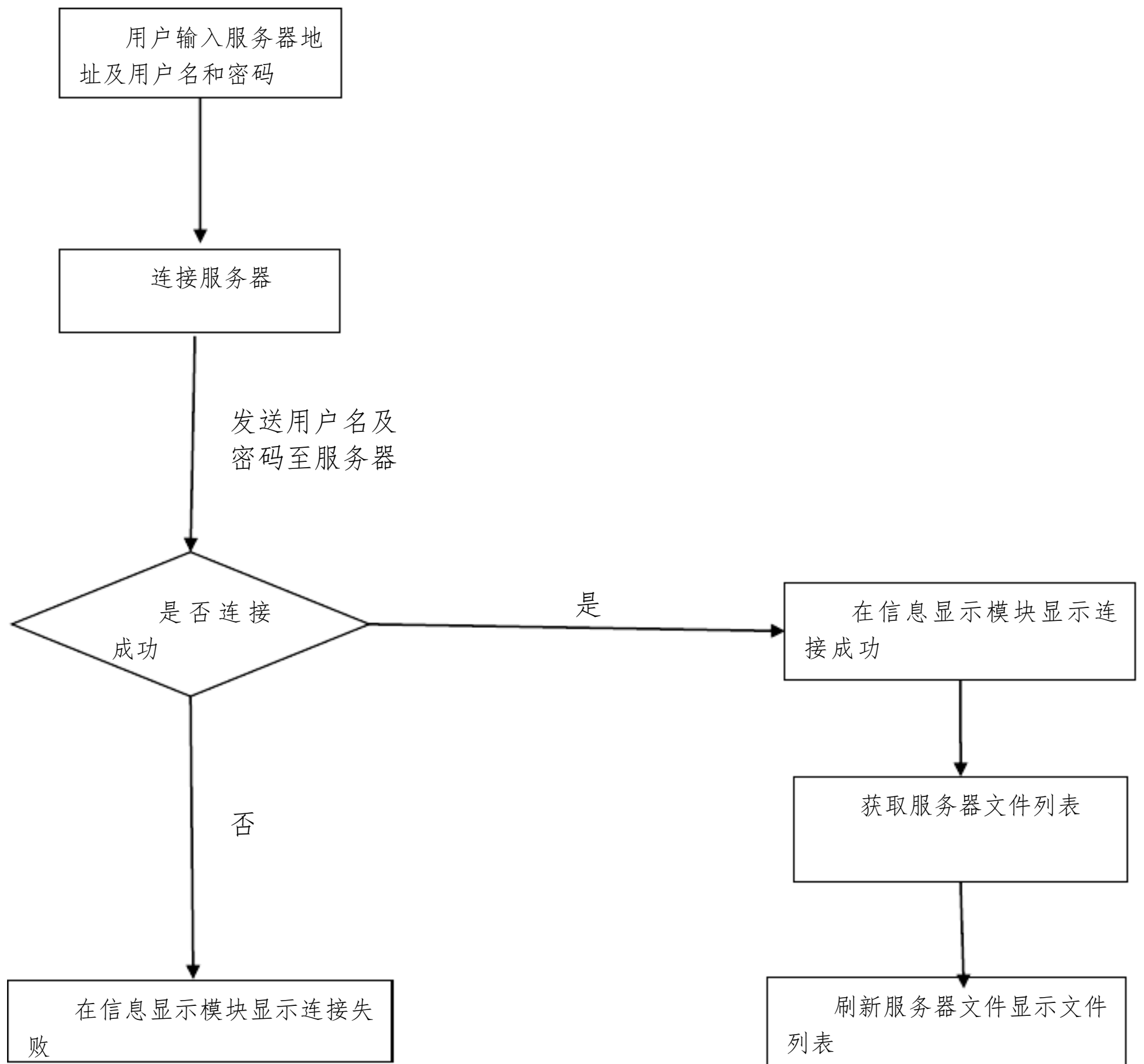


图 3 客户端连接流程图

连接的结果有两种，即图中显示的连接失败和连接成功。



### 3.3.2 上传子模块流程图

当用户点击上传按钮后，用户可以选择需要上传的文件，并指定上传位置，将本机上的文件上传至 FTP 服务器。下图为上传子模块流程图。

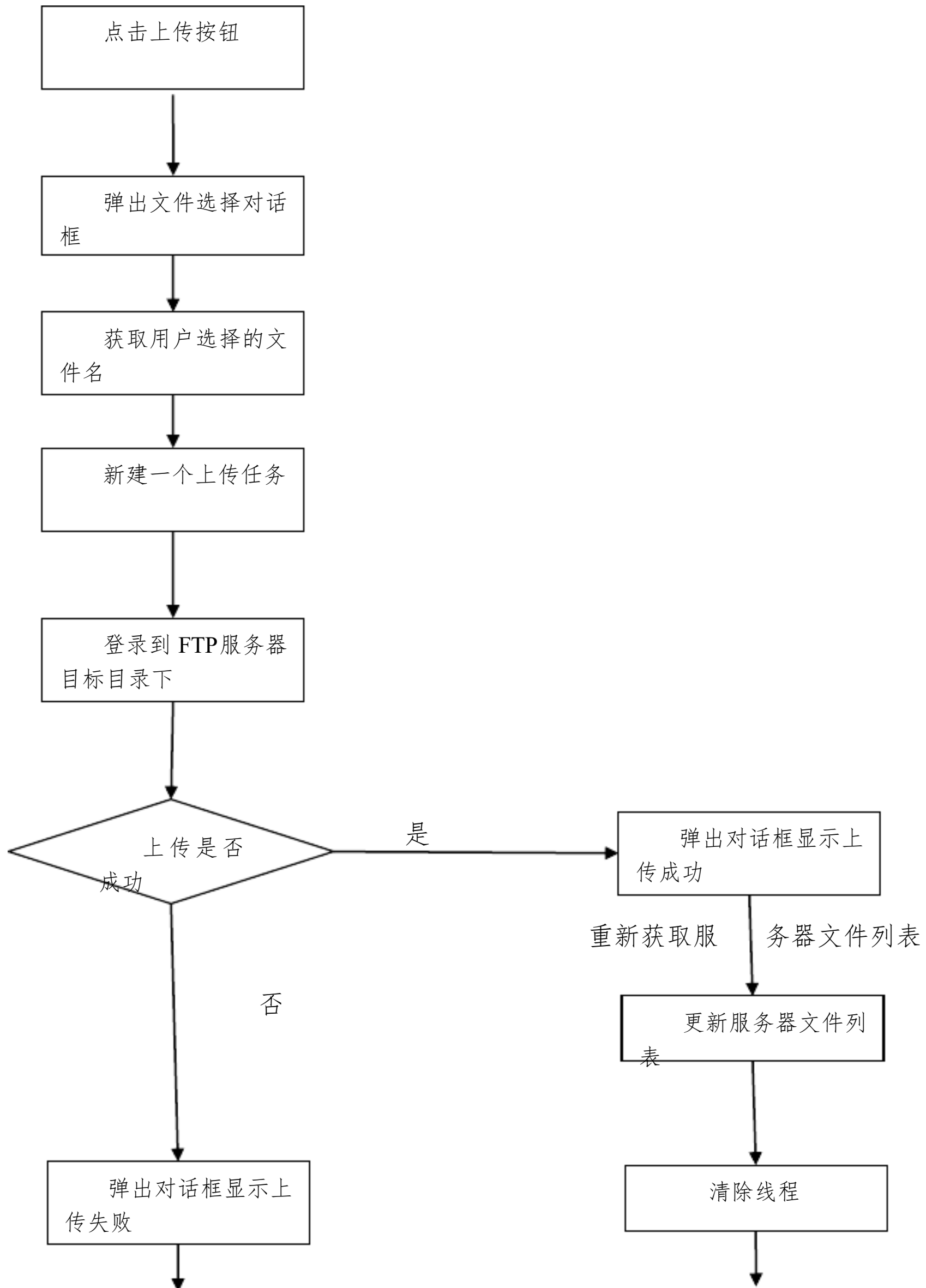


图 4 上传模块流程图

---

上传的结果有两种，上传失败和成功，失败时会提示用户上传失败，原因可能是权限，如果成功上传，则提示上传成功。

### 3.3.3 下载子模块流程图

当用户在服务器文件列表中选择想要下载的文件后，选择需要下载文件在本机上

的存储位置，将服务器上的文件下载至本机。

下图为上传子模块流程图。



以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/547124030061006160>