类型：课程设计

基于 JAVA 的 FTP 文件传输系统设计与开发

简介：随着计算机网络技术的飞速发展，客户/服务 C/S(Client /Sersver) 结构成为开发 FTP的网络编程结构，Java 作为 Web 应用程序的开发技术也为更多的人所采用。

主题词：FTP；C/S；B/S；JAVA

引　言

FTP（File　Transfer　Protocol　）是文件传输协议的简称。FTP的主要作用，就是让用户连接上一个远程计算机（这些计算机上运行着 FTP服务器程序）查看远程计算机有哪些文件，然后把文件从远程计算机上拷到本地计算机，或把本地计算机的文件送到远程计算机去。

目前 FTP服务器软件都为国外作品，例如 Server_U 、IIS ，国内成熟的 FTP服务器软件很少，有一些如（Crob FTP Se rver），但从功能上看来远不能和那些流行的服务器软件媲美。

下面对这些软件简单的做一个比较：

IIS 只适用于 NT/2000/XPWindows操作系统，适合建个小型的同时在线用户数不超过 10 个的 FTP 服务器。它对账户的管理按照Windows用户账户方式进行；

比起 IIS 来，Server_U 的管理功能强大得多，而且设置也很方便。它是一款由 RobBeckers 开发的获奖的 FTP服务器软件，它功能强大又易于使用，支持 9x/ME/NT/2K 等全 Windows系列。FTP 服务器用户通过它用 FTP协议能在 internet 上共享文件。Serv-U 不仅100%遵从通用 FTP标准，也包括众多的独特功能可为每个用户提供文件共享完美解决方案。它并不是简单地提供文件的下载，还为用户的系统安全提供了相当全面的保护。例如：您可以为您的 FTP 设置密码、设置各种用户级的访问许可等等；

而 Crob FTP Server 从功能设置上可以看出，它沿用了像 Server_U 等主流 FTP 服务器软件的基本设置；并加入了不少人性化的功能；同时支持多服务器。（即在软件中可以在任意的有效端口上建立任意多的 FTP服务器并可同时运行，各服务器间互不相干的稳定运行）应该说进步是非常大的。并且可以应用于 Windows 95/98/ME/me/N/T2000 及最新的.NET操作系统上。

不过，纵观上面这些软件，它们都只能在 Windows操作系统中运行，并且功能过于强大，许多功能应用于我们的考试系统的话，并没有太大的意义，而且有些也没必要，于是就需要一个专用的，而且也能通用（应用于 UNIX等其他的 操作系统）的 FTP服务器。要求实习一些基本功能：①文件上传；②文件下载；③文件列表和存在检查、文件属性查询。

基于以上的要求，可以说在选择开发语言的时候，无疑我选择了 JAVA 语言。众所周知，JAVA 是一种可以编写跨平台应用软件的面向对象的程序设计语言。Java 编程语言的风格十分接近 C++语言。Java 继承了 C++语言面向对象技术的核心，舍弃了 C++语言中容易引起错误的指针（以引用取代）、运算符重载、多重继承（以接口取代）等成分，增加了自动垃圾回收器功能用于回收不再被引用的对象所占据的内存空间。Java 最初是为嵌入式小设备而设计的。起初 ，它为人们所喜爱的原因是由于可以用它来开发复杂的、以 Applet 为形式的客户端 Web 应用。然而，服务器端 JAVA 的应用的兴起才是 JAVA 编程最令人振奋的趋势。Java 本质上适合开发大型客户机/服务器

（Client/Server）应用。其跨平台、面向对象、内存保护的安全性、内在的对网络编程的支持以及丰富的 API 等特性使 JAVA 成为服务器端应用开发的一种理想语言。之所以选择 JAVA 语言，还有个重要的原因就是本人的个人所好。

在选择开发平台方面，JCreator、eclipse 、Jbuilder 等都是不错的选择，由于个人喜好我选择了 Jbuilder 作为我的开发平台，而在版本的选择上我选的是 Jbuilder 2005 。

通过阅读大量的文献质料，我发现用 JAVA 来开发 FTP 服务器还是具有一定的难度，主要是由于它不像开发 FTP 客户端的程序那样（Jbuilder 中提供了一类网络类库 sun.net.ftp.FtpClient. ，该类库主要提供了用于建立 FTP 连接的类。利用这些类的方法，编程人员可以远程登录到 FTP 服务器，列举该服务器上的目录，设置传输协议，以及传送文件。FtpClient 类涵盖了几乎所有 FTP 的功能，FtpClient 的实例变量保存了有关建立。），在 Jbuilder 中并没有像 FtpClient 类这样的，来提供给用于服务器端编程的类，所以我们只能从 FTP 的工作原理入手，来解决服务器端的程序编写问题。经过分析，大体思路如下：

在主函数中建立一个服务器套接字端口，等待客户端请求，一旦客户端请求被接受，服务器程序就建立一个服务器分线程，处理客户端的命令。如果客户端需要和服务器端进行文件的传输，则建立一个新的套接字连接来完成文件的操作。

在主函数中，完成服务器端口的侦听和服务线程的创建。线程类的主要设计都是在 run()方法中实现。用 run()方法得到客户端的套接字信息，根据套接字得到输入流和输出流，向客户端发送欢迎信息。在文件传输方面，主要处理从服务器中获得文件 RETR 和向服务器中发送文件 STOR，这两个命令的处理非常类似。处理 RETR 命令时，首先得到用户要获得的文件的名称，根据名称创建一个文件输入流，然后和客户端建立临时套接字连接，并得到一个输出流。随后，将文件输入流中的数据读出并借助于套接字输出流发送到客户端，传输完毕以后，关闭流和临时套接字；STOR 命令的处理也是同样的过程，只是方向正好相反。

当然，这只是大体的思路，具体的实现还涉及到许多细节上的问题了，从理论上来讲，利用 JAVA 来开发 FTP 服务器是可行的，也基本能完成我们预先所要达到的目标，实现一些基本功能，不过在检查文件的属性这方面还有待查找更多相关的质料。

另外，由于 Internet 网络带宽是有限的 ，所以可以考虑将在 Internet 上需要传输的数据文件压缩后再传输，则更有利于数据文件的快速传输，同时，压缩文件也可以节省服务器哦的外部存储空间。实现的时候，主要可应用 Java API 中 java.util.zip 包提供的 ZipEntry、ZipInputStream 和 ZipOutputStream 共 3 个类。本软件将尽量实现这个功能。

1 技术简介

1.1 Java 语言

Java 语言的优点主要表现在：简单、面向对象、多线程、分布性、体系结构中立、安全性等方面。

1.1.1 简单性

Java 与 C++语言非常相近，但 Java 比 C++简单，它抛弃了 C++中的一些不是绝对必要的功能，如头文件、预处理文件、指针、结构、运算符重载、多重继承以及自动强迫同型。Java 实现了自动的垃圾收集，简化了内存管理的工作。这使程序设计更加简便，同时减少了出错的可能。

1.1.2 面向对象

Java 提供了简单的类机制和动态的构架模型。对象中封装了它的状态变量和方法，很好地实现了模块化和信息隐藏；而类则提供了一类对象的原型，通过继承和重载机制，子类可以使用或重新定义父类或超类所提供的方法，从而既实现了代码的复用，又提供了一种动态的解决方案。

Java 是一种完全面向对象的程序设计语言，它除了数组、布尔和字符三个基本数据类型外的其它类都是对象，它不再支持全局变量。在 Java 中，如果不创建新类就无法创建程序，Java 程序在运行时必须先创建一个类的实例，然后才能提交运行。

Java 同样支持继承特性，Java 的类可以从其它类中继承行为，

但 Java 只支持类的单重继承，即每个类只能从一个类中继承。

Java 支持界面，界面允许程序员定义方法但又不立即实现，一个类可以实现多个界面，利用界面可以得到多重继承的许多优点而又没有多重继承的问题。

1.1.3 多线程

多线程使应用程序可以同时进行不同的操作，处理不同的事件。在多线程机制中，不同的线程处理不同的任务，他们之间互不干涉，不会由于一处等待影响其他部分，这样容易实现网络上的实时交互操作。

Java 程序可以有多个执行线程，如可以让一个线程进行复杂的计算，而让另一个线程与用户进行交互，这样用户可以在不中断计算线程的前提下与系统进行交互。多线程保证了较高的执行效率。

1.1.4 分布性

Java 是面向网络的语言。通过它提供的类库可以处理 TCP/IP 协议，用户可以通过 URL地址在网络上很方便的访问其他对象。

1.1.5 体系结构中立

Java 是一种网络语言，为使 Java 程序能在网络的任何地方运行，Java 解释器生成与体系结构无关的字节码结构的文件格式。Java 为了做到结构中立，除生成机器无关的字节码外，还制定了完全统一的语言文本，如 Java 的基本数据类型不会随目标机的变化而变化，一个整型总是 32 位，一个长整型总是 64 位。

为了使 Java 的应用程序能不依赖于具体的系统，Java 语言环境

还提供了用于访问底层操作系统功能的类组成的包，当程序使用这些包时，可以确保它能运行在各种支持 Java 的平台上。

java.lang:　一般的语言包。其中包括用于字符串处理、多线程、异常处理和数字函数等的类，该包是实现 Java 程序运行平台的基本包；

java.util:　实用工具包。其中包括哈希表、堆栈、时间和日期等；

java.io:　基于流模型的输入/输出包。该包用统一的流模型实现了各种格式的输入/输出，包括文件系统、网络和设备的输入/输出等；

java.net:　网络包。该包支持 TCP/IP 协议，其中提供了 socket 、URL 和 WWW 的编程接口；

java.awt:　抽象窗口工具集。其中实现了可以跨平台的图形用户界面组件，包括窗口、菜单、滚动条和对话框等；

java.applet:　支持 applet 程序设计的基本包。

1.1.6 安全性

用于网络、分布环境下的 Java 必须要防止病毒的入侵，Java 不支持指针，一切对内存的访问都必须通过对象的实例变量来实现，这样就防止了程序员使用欺骗手段访问对象的私有成员，同时也避免了指针操作中容易产生的错误。

1.2 JAVA 工具

1.2.1 JDK

(1) Java 编译器

Java 编译器将 Java 源代码文件编译成可执行的 Java 字节码。

Java 源代码文件的扩展名为 .java ，Java 编译器把这种扩展名的文

件编译成扩展名为.class 的文件。源文件中的每个类在编译后都将产生一个 class 文件，这意味一个 Java 源代码文件可能编译生成多个 class 文件。

(2) Java 解释器

Java 解释器对编译生成的字节码格式的可执行程序的运行提供支持，它是运行非图形 Java 程序的命令行工具。

(3) Appletviewer

它是 Java Applet 的简单测试工具，可使用它来测试 Java Applet 程序，而不需要 WWW 浏览器的支持。

1.2.2 Visual J++

Visual J++ 集成了可视化界面设计、交互式调试、代码编辑、联机帮助信息和介绍如何快速掌握该开发环境的实用向导等多项功能，同时具有能充分利用 Active X和 COM 新技术的优势。利用 Visual J++可创建交互性很强的 Internet 应用程序，是难得的 Java 开发系统。

1.3 Java 中输入/输出流概念

过滤流 DataInputStream 和 DataOutputStream 除了分别作为 FilterInputStream 和 FilterOutputStream 的子类外，还分别实现了接口 DataInput 和 DataOutput 。接口 DataInput 中定义的方法主要包括从流中读取基本类型的数据、读取一行数据、或者读取指定长度的字节数，如 readBoolean() readInt() 、readLine() 、readFully()

DataOutput 中定义的方法主要是向流中写入基本类型的数据或者写入一定长度的字节数组，如 writeChar() 、writeDouble() DataInputStream 可以从所连接的输入流中读取与机器无关的基本类型数据，用以实现一种独立于具体平台的输入方式；DataInputStream 可以向所连接的输出流写入基本类型的数据。

机制

Socket 是面向客户/服务器模型设计的，网络上的两个程序通过一个双向的通讯连接实现数据的交换，这个双向链路的一端称为一个 Socket 。 Socket 通常用来实现客户方和服务方的连接。客户程序可以向 Socket 写请求，服务器将处理此请求，然后通过 Socket 将结果返回给用户。

Socket 通信机制提供了两种通讯方式：有联接和无联接方式，分别面向不同的应用需求。使用有联接方式时，通信链路提供了可靠的，全双工的字节流服务。在该方式下，通信双方必须创建一个联接过程并建立一条通讯链路，以后的网络通信操作完全在这一对进程之间进行，通信完毕关闭此联接过程。使用无联接方式时其系统开销比无联接方式小，但通信链路提供了不可靠的数据报服务，不能保证信源所传输的数据一定能够到达信宿。在该方式下，通信双方不必创建一个联接过程和建立一条通讯链路，网络通信操作在不同的主机和进程之间转发进行。

集成开发环境介绍

是全球第一的跨平台 Java 开发环境，可以用于构建符合工业标准的 Java 应用系统，开发 EJB、Web、XML 以及数据库等各类应用程序。双向、可视化设计工具使得我们可以快速的构建各种 J2EE 应用程序，并部署至多种应用程序服务器，包括 BEA WebLogic、IBM WebSphere、Sun ONE Application Server、Oracle 10g Application Server 以及整合于 JBuilder 的 Borland Enterprise Server 。

鉴于此原因，在我们学习 Java 语言的时候，也没有理由杜绝使用 JBuilder 这个优秀的编译器。除非你看到集成开发环境就晕，呵呵（初学者和大师经常这样说）。当然如果你还是初学者，选择 Intellij IDEA 也不错，简单、清晰，他获得了 2003 年 Java 最佳编译器大奖，集成了很多先进的软件工程方法。但是做企业级的开发，JBuilder 无可争论的成为最佳选择。

Quick Start--- 原来 JBuilder 这么容易上手

正如你使用其它的编译器学习 Java 语言一样，你的几个想法之一便是“让我尽快的完成一个 Hello World 吧!”。我们暂且什么都不看，仅仅快速的完成一个 Hello World 程序。安装 JBuilder 似乎并不需要更多的向导大家都能够独立完成。那么赶快打开这个集成开发环境，让我们尽快完成我们的 Hello World 。运行 JBuilder 后我们看到了这样的界面：

赶快新建一个工程(一个习惯是,不管你建立的程序有多么的小,

便于有效的管理你的代码

和编译后生成的资源），在新建的工程中新建一个 HelloWorld 类文件 HelloWorld.java 。你的工程名、类文件名和类名最好是统一的，这是一个良好的编程习惯。需要注意的是：你得类文件名和类名必须是统一的，否则就会出现类似下面的报警错误，我们假设将类名命名为 HelloWorldd，而文件名为 HelloWorld, 编译后就会出现下面的提示信息：

declared in a file named HelloWorldd.java at line 12, columnl

完整的 HelloWorld 程序可以参考下面的建立：

写好你的程序后，编译运行之。右击工程栏当中的 HelloWorld.java, 选择 Debug Using Defaults 。除非上面的步骤出现问题，否则你将会很兴奋的看到，HelloWorld 在信息窗格中输出了！

这样看来似乎 JBuilder 没有那么复杂，对么？很多人花了 N长时间来调试一个 HelloWorld 程序，不是环境变量没有设置好就是忘记这个丢了那个。但你绝不能从此断定“搞定!我已经掌握了 JBuilder!”其实这只是给大家建立一个信心，告诉大家 JBuilder 其实并不像你想象的那样无法接受而已。想要彻底掌握 JBuilder ，我们需要认真的学习下面的指导。

AppBrowser-- 我以后天天对着你打开 JBuilder 集成开发环境的时候，我们面对的便是 AppBrowser。下面我们来逐一的介绍主要的组成部件：