Multi-window FTP Systems in Java: A Network Client

RAJKUMAR and BIJO THOMAS
Operating Systems Group
Centre for Development of Advanced Computing
2/1, Ramanashree Plaza, Brunton Road
Bangalore - 560 025, Karnataka, India
Email: raj@cdacb.ernet.in / kumar@engineer.com URL: http://cdacb.ernet.in/~raj

Abstract: Multithreaded programming is a conceptual paradigm for programming where programs are divided into two or more processes which can be run in parallel. With interactive, networked programs we find that there are many problems best solved concurrently. The work described here is a GUI based multi-threaded FTP package with which you can operate on any number of sites simultaneously.

The package is designed as a powerful sophisticated visual tool for transferring files over the network. Multiple ftp sites can be opened simultaneouly and the directory tree of each location is displayed in separate windows. The basic operations can be done using mouse drag and drop techniques. Also the entire set of commands are available by means of a menu system. The multithreading features of Java enable multiple file transfers at a time thereby reducing the effects of network communication latency. This tools also support file transfers between nodes of a cluster exhibiting a single system image In this paper we discuss the FTP overview, FTP model and package architecture.

Key-words: File-Transfer, Client, Server, Graphical User Interface, and Java.

1. Introduction

The File Transfer Protocol (FTP) is one of the oldest services on the network (intra/inter-net). With FTP, user must be able to remotely connect to an account on any Internet host, look at the files on that host, and transfer data back and forth between the two accounts. If you have an account on the remote host, you can use your password to log in to the account. If you don't have an account on the host, many Internet sites provide accounts onto which you can login as a guest (anonymous FTP servers) to retrieve files.

FTP specifies two essential components for transferring files between two remote machines,

1. FTP Server
2. FTP Client

FTP server listens to a predefined standard port on the machine which offers some FTP services. FTP client can be a user or a user program which communicates the FTP requests to the server port. The work described here is a Graphical User Interface(GUI) based implementation of an FTP Client particularly for cluster of workstations. The functionalities of the package are the following:

1. The user can connect to a remote account on any internet host, look at the files on that host, and transfer data back and forth between the two accounts.
2. User is able to open any number of sites and operate them simultaneously.

3. Each remote site as well as local host will be managed in separate sub-windows of the main frame.
4. Each window will display the directory structure of the corresponding site. The site window can be partitioned into two panels, left panel displays the directory tree and right panel displays the files of the directory selected from the left panel. The directory structure displayed on the screen will be updated to reflect the results of data transfer.
5. Multiple data transfers must be possible at a time, i.e., immediately after initiating a data transfer, the user interface shall be able to accept next user command.
6. All the commands will be available by means of mouse operations as well as detailed menu structure.
7. The package works without any change in almost all the platforms, still maintaining a similar look and feel.

The package is implemented in Java which offers multithreading and networking facilities. The language also supports the platform independent features required for the system.

2. Java Language

Java is a programming language developed by a team headed by James Gosling at Sun Microsystems Inc. In *The Java Language: A White Paper* [1], Sun describes Java as follows.

> *Java is simple, object oriented, distributed, interpreted, robust, secure, architecture-neutral, portable, high performance, multithreaded, and dynamic language. Though this is quite a string of buzzwords, the fact is that they all aptly describe the language.*

The Java environment [2] is a threefold set of specification and tools that allow developers to produce dynamic, portable and high performance programs as easily as possible, while giving the end user and administrator a secure robust runtime environment in which these applications can be run.

Java provides an object oriented, familiar, easy-to-use language with which applications can be developed. The Java architecture incorporates modern features and is tuned to distributed, heterogeneous platforms. The Java tools enable programmers and end users the flexibility to produce and use the content which will shape the internet in the next decade [3, 4, and 5].

3. FTP - An Overview

The objectives of FTP are the following:

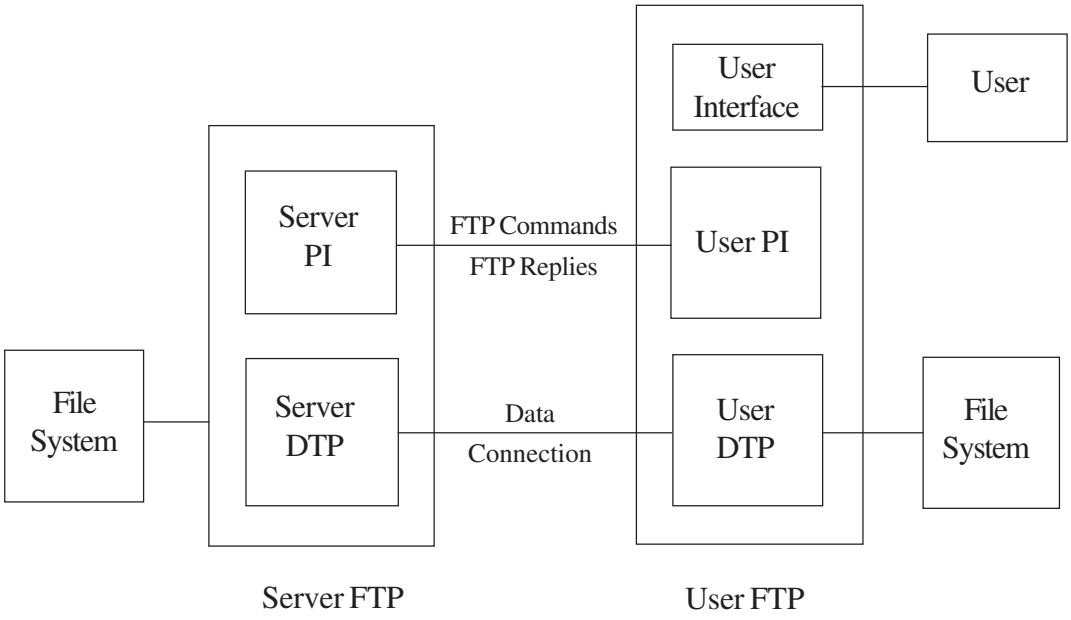
1) to promote sharing of files (computer programs and/or data),
2) to encourage indirect or implicit (via programs) use of remote computers,
3) to shield a user from variations in file storage systems among hosts, and
4) to transfer data reliably and efficiently.

FTP, though usable directly by a user at a terminal, is designed mainly for use by programs.

3.1 The FTP Model

With the above requirements in mind, the following model (shown in Figure 1) may be diagrammed for an FTP service [9]. The FTP related terms used in this section are explained in Appendix.

The communication channel from the user-PI to the server-PI is established as a TCP connection from the user to the standard server port. The user protocol interpreter is responsible for sending FTP commands and interpreting the replies received; the server-PI interprets commands, sends replies and



NOTES: 1. The data connection may be used in either direction.
2. The data connection need not exist all of the time.

Figure 1: Model for FTP use

directs its DTP to set up the data connection and transfer the data. If the second party to the data transfer (the passive transfer process) is the user-DTP, then it is governed through the internal protocol of the user-FTP host; if it is a second server-DTP, then it is governed by its PI on command from the user-PI. The FTP replies are discussed in the next section. In the model described in Figure 1, the user-protocol interpreter initiates the control connection. The control connection follows the Telnet protocol. At the initiation of the user, standard FTP commands are generated by the user-PI and transmitted to the server process via the control connection. Standard replies are sent from the server-PI to the user-PI over the control connection in response to the commands.

The FTP commands [1] specify the parameters for the data connection (data port, transfer mode, representation type, and structure) and the nature of file system operation (store, retrieve, append, delete, etc.). The user-DTP or its designate should "listen" on the specified data port, and the server initiate the data connection and data transfer in accordance with the specified parameters. It should be noted that the data port need not be in the same host that initiates the FTP commands via the control connection, but the user or the user-FTP process must ensure a" listen" on the specified data port. It ought to also be noted that the data connection may be used for simultaneous sending and receiving. The protocol requires that the control connections be open while data transfer is in progress. It is the responsibility of the user to request the closing of the control connections when finished using the FTP service, while it is the server who takes the action. The server may abort data transfer if the control connections are closed without command.

3.2 FTP Replies

Replies to File Transfer Protocol commands [9] are devised to ensure the synchronization of requests and actions in the process of file transfer, and to guarantee that the user process always knows the state of the Server. Every command must generate at least one reply, although there may be more than one; in the latter case, the multiple replies must be easily distinguished.

An FTP reply consists of a three digit number (transmitted as three alphanumeric characters) followed by some text. The number is intended for use by automata to determine what state to enter next; the text is intended for the human user. The three digits of the reply each have a special significance. The first digit denotes whether the response is good, bad or incomplete. A user-process that wants to know approximately what kind of error occurred (e.g. file system error, command syntax error) may examine the second digit, reserving the third digit for the finest gradation of information (e.g., RNTO command without a preceding RNFR) [9].

4. System Architecture

Here, we present a generalized diagram that is used to model the command and reply interchange in the package. For each command or command sequence there are three possible outcomes: success (S), failure (F), and error (E). In the state diagrams below we use the symbol B for "begin", and the symbol W for "wait for reply".

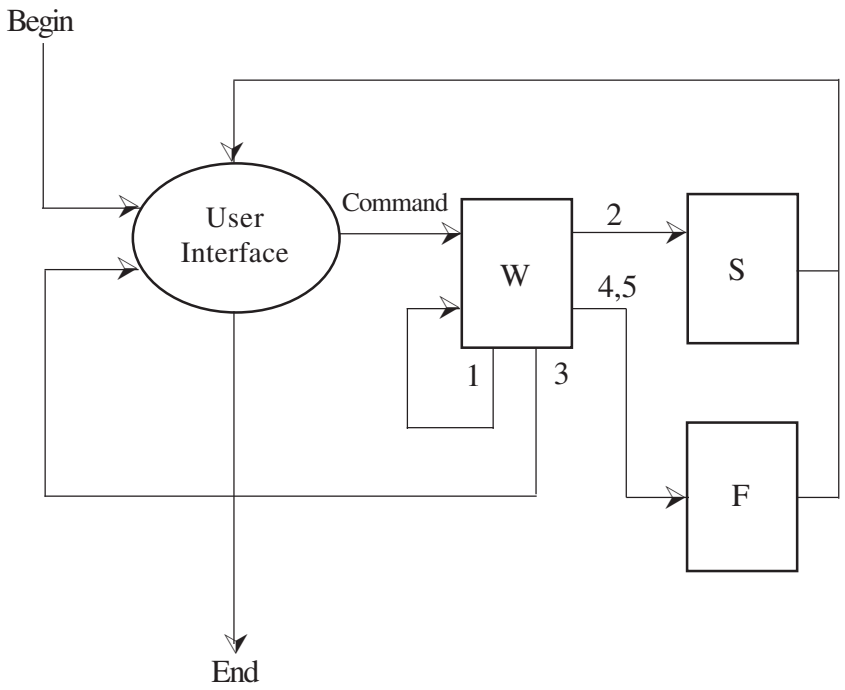


Figure 2. Flow of Control

5. Implementation

Initially when the system is opened, a main frame displays the local site's directory tree. Then user can open any number of remote ftp sites in separate windows. The sub-windows in the frame get automatically adjusted to provide space for the newly opening window. The file operations can be done in the same fashion irrespective of whether it is the local site or remote site. Appropriate dialog boxes are opened to receive user inputs for each command.

The hierarchy of classes developed for the system is closely related with the GUI hierarchy of the package. This is common to almost all the GUI based projects because, the action to be taken by such a system is determined by a sequence of GUI events.

5.1 GUI hierarchy

The major GUI windows used are shown below.

MainFrame
- MenuBar
- WorkSpace
- StatusBar

WorkSpace
- LocalSiteWindow
- FtpSiteWindow 1
- FtpSiteWindow 2
- |
- FtpSiteWindow N

LocalSiteWindow
- Title
- DisplayPanel
- StatusBar

FtpSiteWindow
- Title
- DisplayPanel
- Statusbar

DisplayPanel
- TreeScrollPane
- FileScrollPane

TreeScrollPane
- Title
- TreePanel

FileScrollPane
- TitleBox
- FilePanel

TitleBox
- Filed Headings
- Relocatable Date TabButton
- Relocatable Size TabButton

TreePanel
- N rows of indented labels
  - for each directory

FilePanel
- FileNameLabel 1
- |
- FileNameLabel N

FileNameLabel
- NameLabel
- DateLabel
- SizeLabel

5.2 Menu System

The menu hierarchy of the package is given below. This systems has a facility for displaying or ordering files on alphabetically, date of creation, or their size whether they are at local or remote sites. It also supports file transmission in both ASCII and binary modes with ASCII as the default mode.

File
- Open Site
- Close Site
- Copy File
- Rename File
- Delete File
- Exit

View
- Date
- Size

Options
- Ascii
- Binary
- Preferences

Help
- About

The multiwindow allows to open multiple sites as shown Figure 3 whether they are local or remote. When started, it opens local site and other sites have to be opened to initiate file transfers. The file transfer is multithreaded and hence, multiple transmissions can be initiated and processed simultaneously [6].

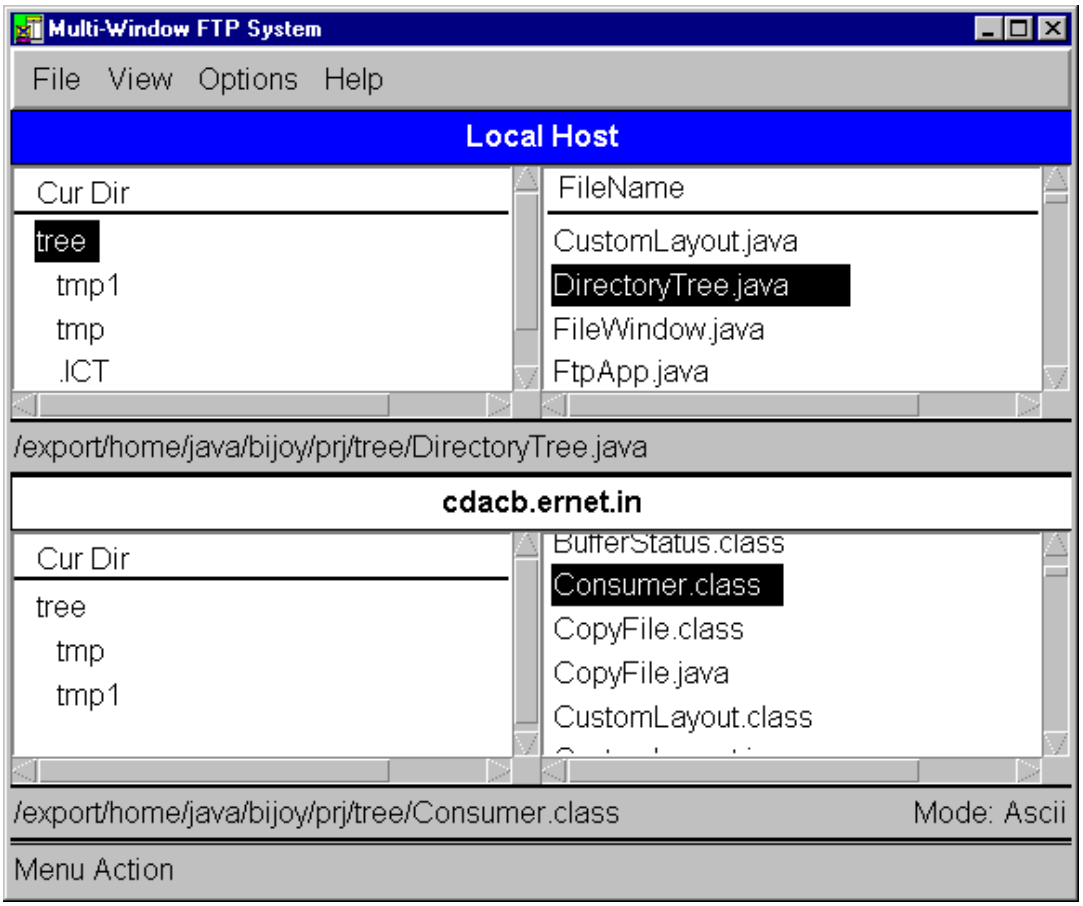


Figure 2. Opening of Multiple Sites

On Cluster of Workstations (COW), this system can be configure to open sites for all the node of COW to facilitate transmission of files with ease. This can also be configured to exhibit a single system image [7] on clusters at application level [8].

6. Conclusion and Future Work

The GUI based FTP package offers the following advantages over other similar packages.

- The package is a platform independent solution to the File Transfer needs of the network user. This becomes particularly important in today's networking environment which is built on a wide variety of platforms.
- Easier User interface.
- Multiple sites can be opened at a time.
- File Transfer operations can be done in the same fashion irrespective of whether they are local or remote sites.
- Operations are multithreaded, i.e., different file transfer operations can take place simultaneously, thereby reducing effect of network communication latency.

The main objective of the present system was to build a platform independent multithreaded tool for FTP and therefore a few complex parts of the File Transfer Protocol were not attempted. It supports only stream mode and file structure for data transfer. Now the system can easily be extended to include other modes and structures. It is to be noted that not all the sites over the network implement all the available modes and structures. Also the GUI does not support mouse drag and drop operations, since such facilities is not yet available as Java features. Sun has promised to provide these features in future version of Java, which will also be incorporated into our system.

7. Availability

The portable multiwindow FTP system developed in Java is available in the public domain by remote FTP access. At the site, `http://cdacb.ernet.in/~raj` goto the section *Research Publications*, and click on this paper title, which displays abstract of this paper. Click on Download Multiwindow FTP System Package, to download the package. The package is stored in the file `mwftp.zip`, which contains all source files compressed and zipped using `pkzip` utility. Suggestions for further improvement are welcome and they can be mailed to `raj@cdacb.ernet.in` / `kumar@engineer.com`.

References

[1] Sun Microsystems, *Java Language - A White Paper*, Sun Microsystems Computer Company, 1996.

[2] James Gosling,Frank Yellin & The Java Team, *The Java Application Programming Interface,* Volume I and II,Sun Microsystem, Addison Wesley, 1996.

[3] Tim Ritchey, *Java!*, New Riders Publishing, Indianapolis,Indiana,1996.

[4] Patrick Naughton, *The Java Handbook*, McGraw-Hill Inc,1996.

[5] Venugopal and Rajkumar, *Mastering C++*, Tata McGraw Hill Publications, 1997.

[6] Rajkumar and et. al., *PEACE Threads Interface on PARAS Microkernel*, Proceedings of the Third International Conference on Advanced Computing, Tata McGraw Hill Publications, India, 1995.

[7] Rajkumar, *Single System Image: Need, Approaches, and Supporting HPC Systems*, Proceedings of the Fourth International Conference on Parallel and Distributed Processing, Techniques and Applications (PDPTA'97), CSREA Publishers, Las Vegas, USA, 1997.

[8] Gregory F Pfister, *Clusters of computers for commercial processing: the invisible architecture*, Parallel and Distributed Processing, IEEE, Fall 1996.

[9] Postel and Reynolds, *File Transfer Protocols,* RFC 953.