

國立臺南大學

資訊工程學系

網路程式設計

**FTP 客戶端**

**FTP Client**

學號：S10259002

姓名：黃詩豪

指導教授：楊文霖

中華民國 105 年 05 月 30 日

# 目錄

目錄.....	2
圖目表.....	3
表目錄.....	3
一、系統需求與目標（System Objectives） .....	4
二、系統主要功能描述（Description of requirement operations） .....	5
2-1、連接 FTP Server 主機 .....	5
2-2、上傳、下載、刪除功能.....	5
2-3、相容之傳輸模式.....	5
四、系統設計（System Design） .....	7
4-1、設計方法與工具.....	7
4-2、軟體組織架構.....	7
4-3、軟體元件模組設計.....	7
五、系統說明與簡易使用者手冊（System User Guide） .....	10
5-1、系統初始畫面.....	10
5-2、Linux Mint 17.2 系統下之操作 .....	11
5-3、Window 7 系統下之操作 .....	15

## 圖目表

Fig. 1 主動模式連接因為防火牆或防毒軟體造成無法連線 .....	4
Fig. 2 Linux Server 的 LIST 指令回應 .....	5
Fig. 3 Microsoft OS Server 的 MLSD 指令回應 .....	5
Fig. 4 FTP 回應 Code 對應說明 .....	6
Fig. 5 系統初始畫面 .....	10
Fig. 6 「主機名稱」未輸入 .....	11
Fig. 7 成功連線 vsftpd Server 於 linux 系統上 .....	11
Fig. 8 以拖曳方式上傳多個選定之中文檔案 .....	12
Fig. 9 上傳「123」資料夾至/networkTest/ftpctest/目錄下 .....	12
Fig. 10 下載英文、數字檔案至本地端目錄 .....	13
Fig. 11 下載「123」資料夾至本地目錄 .....	13
Fig. 12 提示是否刪除所選定之檔案及目錄 .....	14
Fig. 13 成功刪除選定之檔案及目錄 .....	14
Fig. 14 結束連接顯示 Server 回應「221 Goodbye」訊息 .....	15
Fig. 15 成功連接遠端 Filezilla FTP Server 於 Window 7 系統上 .....	15
Fig. 16 上傳中、英文檔案名稱皆能成功 .....	16
Fig. 17 下載與刪除遠端中、英文檔案皆能成功 .....	16
Fig. 18 結束連接 Server 顯示回應 .....	17

## 表目錄

Table. 1 FtpFile 類別功能說明表格 .....	8
Table. 2 FtpClient 類別功能說明表格 .....	9
Table. 3 MainUI 類別功能說明表格 .....	10

## 一、系統需求與目標 (System Objectives)

設計一個具有圖形使用者介面 FTP Client 程式，內含下列 ftp 通訊協定命令：USER、PASS、CWD、QUIT、PORT、PASV、SYST、FEAT、TYPE A、TYPE I、RETR、STOR、DELE、RMD、MKD、PWD、LIST、NLST、MLSD。

- **傳輸模式：**FTP 連接模式分為主動式 (Active Mode，指令：PORT) 與被動式 (Passive Mode，指令：PASV)，傳統上採用主動式連接之方式，告訴 Server 本地端 Client 在哪個 port 等待並接收資料。後者，基於 Server 連接到 Client 過程之安全性疑慮，所發展出另外一種模式，是由 Server 收到 PASV 指令後，回傳 Server IP 與 port 號碼，被動地等待 Client 連接，然後傳輸資料。但主動模式可能因為防火牆、防毒軟體之因素，可能發生連接不成功之情況，如 Fig. 1 所示，因此再本系統中，優先使用被動模式，如果被動模式無法使用，則改用主動模式連接。

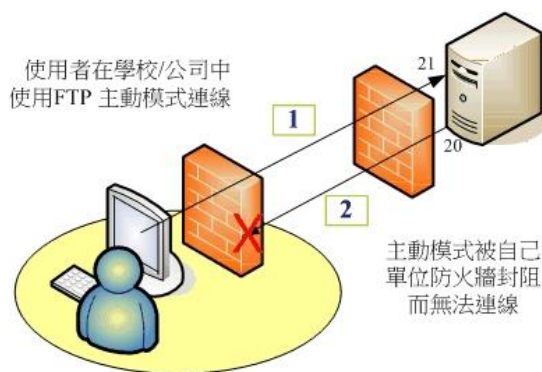


Fig. 1 主動模式連接因為防火牆或防毒軟體造成無法連線

- **上傳、下載、刪除檔案功能：**為了使用上更方便，使用拖拉方式達成上傳與下載之操作，刪除則是當使用者按下鍵盤上的 delete 鍵後，詢問是否要刪除選擇之目錄或檔案，確定則刪除所選定的目錄及檔案，另外，也實作「整個資料夾」及「選定多個目錄及檔案」的上傳、下載、刪除功能，另外，基於錯誤操作而造成不可逆之動作，「刪除目錄或檔案」的功能只限定於 Server 目錄中，因此能達到更佳之使用者體驗。
- **FTP Server 作業系統清單指令：**對於不同之作業系統，如 UNIX、Linux、Microsoft OS，對應之系統清單指令並不相同，以 UNIX 和 Linux 是 LIST 指令，Microsoft OS 是 MLSD 指令，兩者指令差別在於回應的資料不同，前者較為詳細，通常為權限、擁有者、群組、大小、最後修改時間、名稱組成，如 Fig. 2 所示；後者為型態（檔案或目錄）、最後修改時間、大小（檔案）、名稱，如 Fig.3 所示。故本系統也將對此作為區分，利用 FEAT 指令取得 Server 是使用哪種清單指令。

- **Server 回應與操作紀錄顯示：**實作兩個區塊顯示系統傳送之指令與 Server 的回應，以及上傳、下載、刪除的檔案紀錄，且在每次連線時，清除舊有的紀錄，以確保不與先前連線搞混。

```
-rwxrwxr-x 1 501 501 12739 May 18 14:40 index.html
drwxrwxr-x 2 501 501 4096 Jan 22 07:52 js
```

Fig. 2 Linux Server 的 LIST 指令回應

```
type=file;modify=20140619054248;size=753; googleMap.html
type=dir;modify=20160528142223; hshpathcomp
type=file;modify=20130801123529;size=238; httpwd.php
```

Fig. 3 Microsoft OS Server 的 MLSD 指令回應

## 二、系統主要功能描述（Description of requirement operations）

### 2-1、連接 FTP Server 主機

- 輸入主機名稱或主機 IP 位置連接
- 輸入使用者帳號與使用者密碼，並以默認連接埠 21 連接 Server
- 未輸入使用者資訊時以匿名方式登入
- 連接失敗或登入失敗之錯誤訊息處理並顯示於介面上

### 2-2、上傳、下載、刪除功能

- 可選定單一或多個目錄或檔案
- 直接取代存在之目錄或檔案
- 拖拉之方式上傳與下載檔案
- 鍵盤 delete 按鍵刪除選定的檔案及目錄
- 因安全性疑慮，故只能對遠端目錄執行刪除動作

### 2-3、相容之傳輸模式

- 先使用被動式傳輸模式，如果無法連接，再使用主動式傳輸模式
- 相容本地端測試以主動式傳輸模式連接

## 三、系統分析（System Analysis）

系統以圖形化介面設計方便與程式使用者互動，然後將後端與前端主要功能分開撰寫，後端實作 FTP 協定之指令發送、傳送與接收資料介面，以及 Server 回應之處理，前端參照 FileZilla FTP Client 介面設計方式，實作本地端檔案系統樹，本地端檔案系統表格，遠端檔案系統表格，兩個操作記錄顯示面版，另外，採用拖放方式（Drag and Drop，DnD）上傳與下載目錄及檔案，以及遞迴搜尋是否還存在目錄或檔案，以此實現整個資料夾之上傳與下載功

能，最後，考量本地端檔案誤刪之情況，只允許遠端目錄下使用鍵盤之 delete 按鍵刪除目錄及檔案，且於按下按鍵時，會提示使用者是否確定刪除，也能確保操作上之正確性。

另外，FTP Client 與 Server 進行 Data Transfer 之模式以被動模式優先執行，若被動模式無法建立連接，再以主動模式連接，因此能解決 Client 於多重 Router 環境下使用主動模式所造成 Server 無法找到 Client 所在位置之問題。然後 Server 對於指令之回應使用另外的 Thread 處理，以解決 Multi-line Response 之問題，且為了與 Main Thread 溝通，建立同步化之狀態判讀，根據 FTP 回應 Code 定義如 Fig. 4 所示，當回應為 1xx 種類時，表示 Server 即將開始傳送資料，當 2xx 回應表示 Server 成功執行指令，故於 Main Thread 會暫停直到接收到 Server 回應才繼續執行，因此能取得 Server 之回應，並對回應加以處理，若回應為 3xx、4xx、5xx 時，本系統只處理特殊狀況，例如：421 Timeout。

Range	Purpose
1xx	Positive Preliminary reply The requested action is being initiated; expect another reply before proceeding with a new command. (The user-process sending another command before the completion reply would be in violation of protocol; but server-FTP processes should queue any commands that arrive while a preceding command is in progress.) This type of reply can be used to indicate that the command was accepted and the user-process may now pay attention to the data connections, for implementations where simultaneous monitoring is difficult. The server-FTP process may send at most, one 1xx reply per command.
2xx	Positive Completion reply The requested action has been successfully completed. A new request may be initiated.
3xx	Positive Intermediate reply The command has been accepted, but the requested action is being held in abeyance, pending receipt of further information. The user should send another command specifying this information. This reply is used in command sequence groups.
4xx	Transient Negative Completion reply The command was not accepted and the requested action did not take place, but the error condition is temporary and the action may be requested again. The user should return to the beginning of the command sequence, if any. It is difficult to assign a meaning to "transient", particularly when two distinct sites (Server- and User-processes) have to agree on the interpretation. Each reply in the 4xx category might have a slightly different time value, but the intent is that the user-process is encouraged to try again. A rule of thumb in determining if a reply fits into the 4xx or the 5xx (Permanent Negative) category is that replies are 4xx if the commands can be repeated without any change in command form or in properties of the User or Server (e.g., the command is spelled the same with the same arguments used; the user does not change his file access or user name; the server does not put up a new implementation.)
5xx	Permanent Negative Completion reply The command was not accepted and the requested action did not take place. The User-process is discouraged from repeating the exact request (in the same sequence). Even some "permanent" error conditions can be corrected, so the human user may want to direct his User-process to reinitiate the command sequence by direct action at some point in the future (e.g., after the spelling has been changed, or the user has altered his directory status.)
6xx	Protected reply The RFC 2228 introduced the concept of protected replies to increase security over the FTP communications. The 6xx replies are Base64 encoded protected messages that serves as responses to secure commands. When properly decoded, these replies fall into the above categories.

Fig. 4 FTP 回應 Code 對應說明

現今多數 FTP Server 無不是使用 Filezilla FTP Server 就是使用 vsftpd 架設，雖然兩者皆支援 UTF-8 編碼，但是傳送指令是使用 Client 之作業系統的命令提示字元傳送，因此當 Client 於 Microsoft OS 是以 BIG5 為編碼，傳送至以 UTF-8 之 Server 所在之作業系統時，會發生編碼系統不一致之問題，導致中文文字亂碼的產生，因此在傳送指令之過程時，對指令以 UTF-8 編碼，再送出至 Server，且於接收時同樣使用 UTF-8 編碼接收資料，如此一來，能解決不同系統之間傳送而造成中文目錄及檔案名稱之亂碼問題。

## 四、系統設計 (System Design)

### 4-1、設計方法與工具

視窗互動介面設計使用物件導向設計，且是以前端圖形顯示與後端資料處理方式設計，系統分析工具則是使用 Word 與 Draw.io，程式開發工具為 Java 語言，在 Linux Mint 17.2 作業系統上使用 Eclipse MARS.2 開發，程式可於已安裝 JVM 之環境執行。

### 4-2、軟體組織架構

使用物件導向方式建立此系統，前端圖形化介面使用 Swing library，後端資料處理先建立 FtpClient 處理指令發送、接收與傳送資料 Socket、Server 回應之 Thread；FtpFile 處理當使用 LIST 或 MLSD 指令時，Server 所回應的目錄檔案清單訊息；前端建立本地端檔案系統樹及表格，遠端則建立檔案系統表格，並且建立兩個顯示面版，將 Server 回應與使用者操作記錄顯示。為了更方便操作，採用 DnD 方式上傳與下載目錄及檔案，並使用鍵盤上之 delete 按鍵進行遠端目錄及檔案刪除操作。

### 4-3、軟體元件模組設計

- FtpFile 類別 (FtpFile Class)

功能名稱	功能說明
isDirecotry()	判斷 FtpFile 是否為目錄
isFile()	判斷 FtpFile 是否為檔案
getAbsolutePath()	取得 FtpFile 於遠端之絕對路徑
getAuth()/setAuth(String auth)	取得/設定權限資訊
getOwner()/setOwner(String owner)	取得/設定擁有者資訊
getGroup()/setGroup(group)	取得/設定群組資訊
getSize()/setSize(String size)	取得/設定檔案大小
getLastTime()/setLastTime(String lastTime)	取得/設定最後修改時間

getName()/setName(String name)	取得/設定檔案名稱
getParent()/setParent(String parent)	取得/設定上層資料夾目錄
getType()/setType(String type)	取得/設定檔案類型

Table. 1 FtpFile 類別功能說明表格

● FTP 客戶端類別 (FtpClient Class)

功能名稱	功能說明
send(String cmd)	傳送 FTP 協定的指令
dataConnect(String cmd)	先建立被動模式連接，若無法建立，則使用主動模式連接，並回傳與 Server 溝通之 data socket
doOpen(String host, int port)	建立與 Server 之 control socket 連接
doQuit()	結束當前之連接
doLogin(String name, String password)	當建立連接後，使用者登入
doLs()	取得 Server 當前目錄下之所有目錄及檔案清單，並回傳 FtpFile 物件清單
doNls()	取得 Server 當前目錄下之所有目錄及檔案名稱清單
doAscii()	設定當前以 ASCII 模式傳輸資料
doBinary()	設定當前以 binary 模式傳輸資料
doOpts()	設定 Server 以 UTF-8 編碼
doGet(String f, String local, String remote)	下載遠端檔案
doPut(File f, String remote)	上傳本地檔案
doDelete(String remote)	刪除遠端檔案
doCd(String remote)	切換遠端當前目錄
doRmd(String remote)	刪除遠端目錄



doMkd(String remote)	建立遠端目錄
doPwd()	顯示遠端當前工作目錄
doType()	顯示當前 Server 使用的資料傳輸是以 ASCII 還是以 binary
doFeat()	取得 Server 一些特殊規定，用已判斷是否使用 LIST 或 MLSD
sendMsgPane(String msg, Enum MSG_TYPE type)	將訊息發送至前端顯示面版
getFtpFile(String file)	處理 LIST 及 MLSD 指令回應之訊息，以 Regular Expression 進行 Parsing

Table. 2 FtpClient 類別功能說明表格

- 主介面類別 (MainUI Class)

功能名稱	功能說明
getRemoteList(String dirStr, String parent)	取得遠端目錄及檔案，並顯示於遠端表格中
getLocalList(String dirStr)	取得本地目錄及檔案，並顯示於本地端表格中
readableFileType(String filename, Boolean type)	將檔案類型轉變為較方便讀取之格式
readableFileSize(long size)	將檔案大小轉變為以 B、KB、MB、GB、TB 之單位顯示，以增加可讀性
realFileSize(String size)	將檔案大小轉為以 Byte 為單位數值
getParentDir(String cur)	取得本地端路徑之上層路徑
getDirectory(String cur)	將本地端路徑正規化，適應 Linux 與 Microsoft OS 所使用的檔案分隔符號
getRemoteParentDir(String cur)	取得遠端路徑之上層路徑
getRemoteDir(String cur)	正規化遠端路徑

getRoot()	取得本地端根目錄
uploadProcessor(File root, String remoteRoot)	以遞迴之方式上傳以 root 為目錄或檔案至遠端路徑底下
downloadProcessor(FtpFile tf, String remoteRoot, String localRoot)	以遞迴之方式下載遠端 tf 目錄或檔案至本地端目錄下
deleteProcessor(FtpFile tf, String remoteRoot)	以遞迴之方式刪除遠端以 tf 為檔案或目錄

Table. 3 MainUI 類別功能說明表格

## 五、系統說明與簡易使用者手冊（System User Guide）

### 5-1、系統初始畫面

系統初始畫面，如 Fig. 5Fig. 6 所示，從上至下、左至右分別是主機、使用者、密碼、連接埠、連接按鈕、結束連接按鈕，指令與回應顯示面版，檔案及目錄操作工作區，包含本地端目錄樹與檔案目錄表格，遠端檔案目錄表格，最後是上傳、下載、刪除操作記錄顯示面版。輸入面版會對「未輸入主機」進行檢查，但是對使用者允許匿名登入操作，故不檢查，如所示。

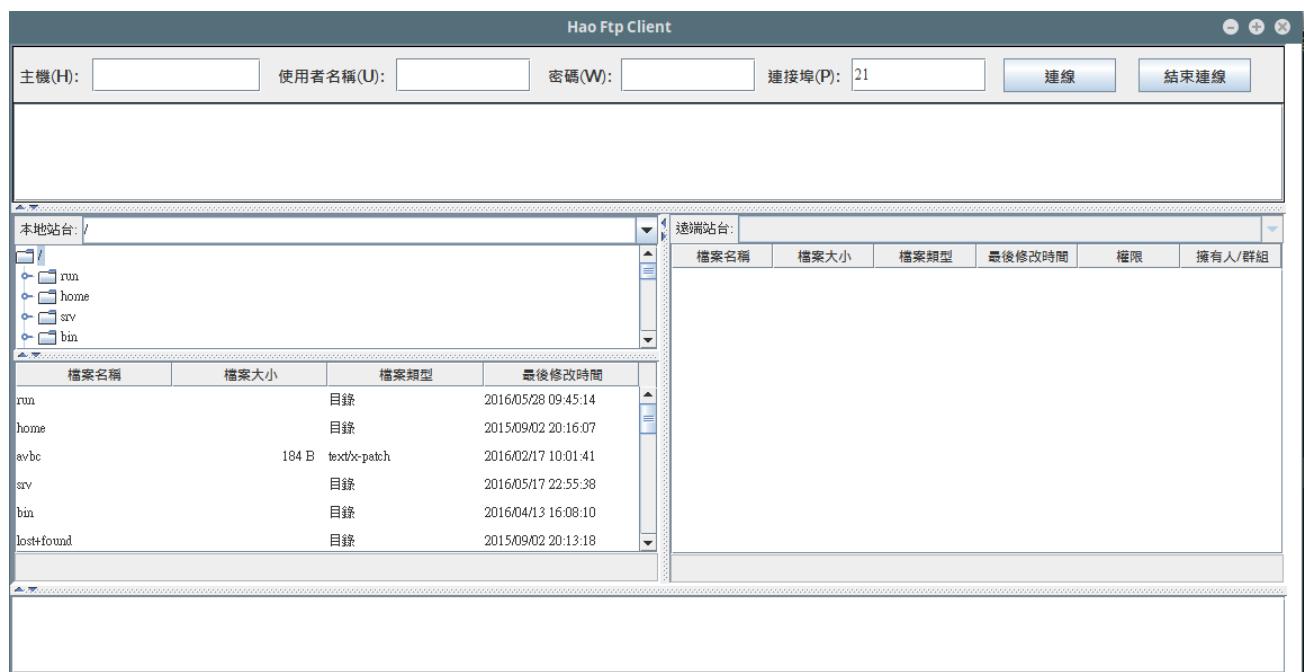


Fig. 5 系統初始畫面



Fig. 6 「主機名稱」未輸入

## 5-2、Linux Mint 17.2 系統下之操作

本系統於 Linux Mint 17.2 系統下執行，連接遠端以 vsftpd 架設於 linux 系統下之 FTP Server，當輸入主機、使用者、密碼後，按下「連線」，會顯示連線成功資訊與遠端檔案目錄表格會顯示檔案及目錄資訊，如 Fig. 7 所示。系統能使用滑鼠拖曳檔案或目錄上傳至遠端目錄下，Fig. 8 是上傳中文檔案，Fig. 9 是上傳整個資料夾，Fig. 10 是下載英文檔案，Fig. 11 下載整個資料夾，Fig. 12 是當使用者按下「delete」按鍵提示是否真的要刪除檔案，Fig. 13 是刪除遠端檔案及目錄之結果，最後 Fig. 14 是按下「結束連線」時顯示 Server 回應於面版中顯示。

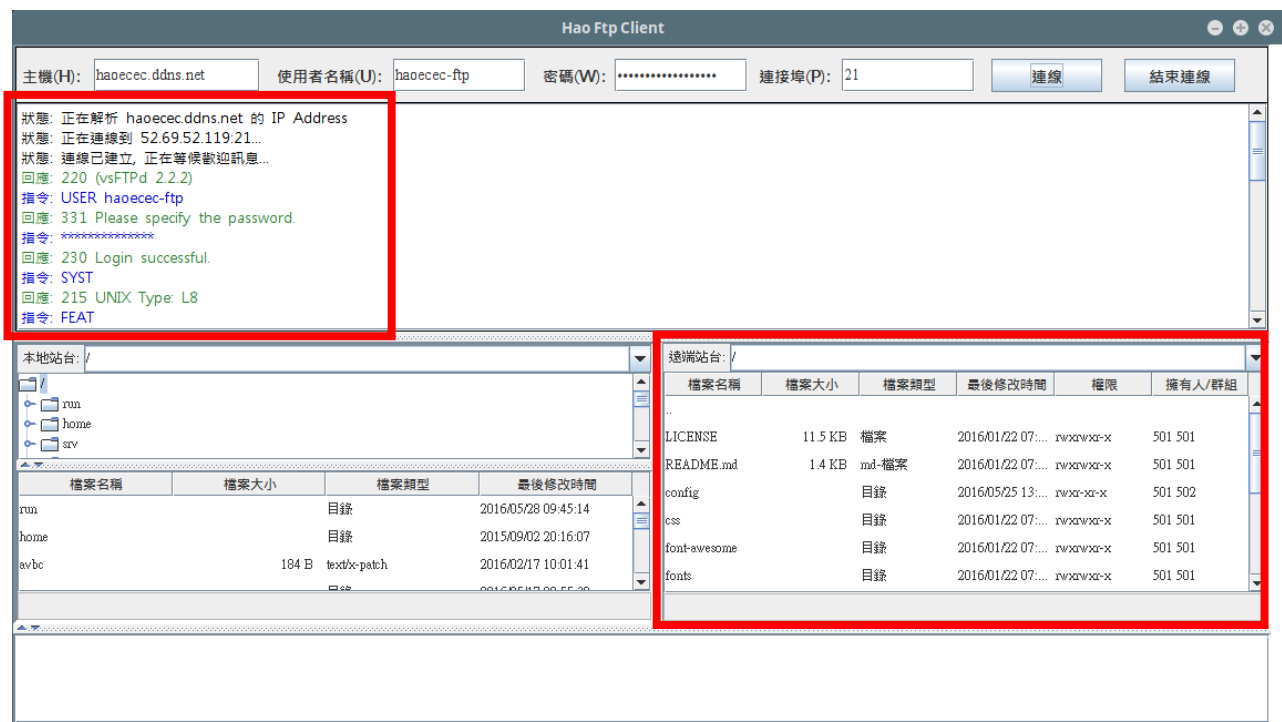


Fig. 7 成功連線 vsftpd Server 於 linux 系統上

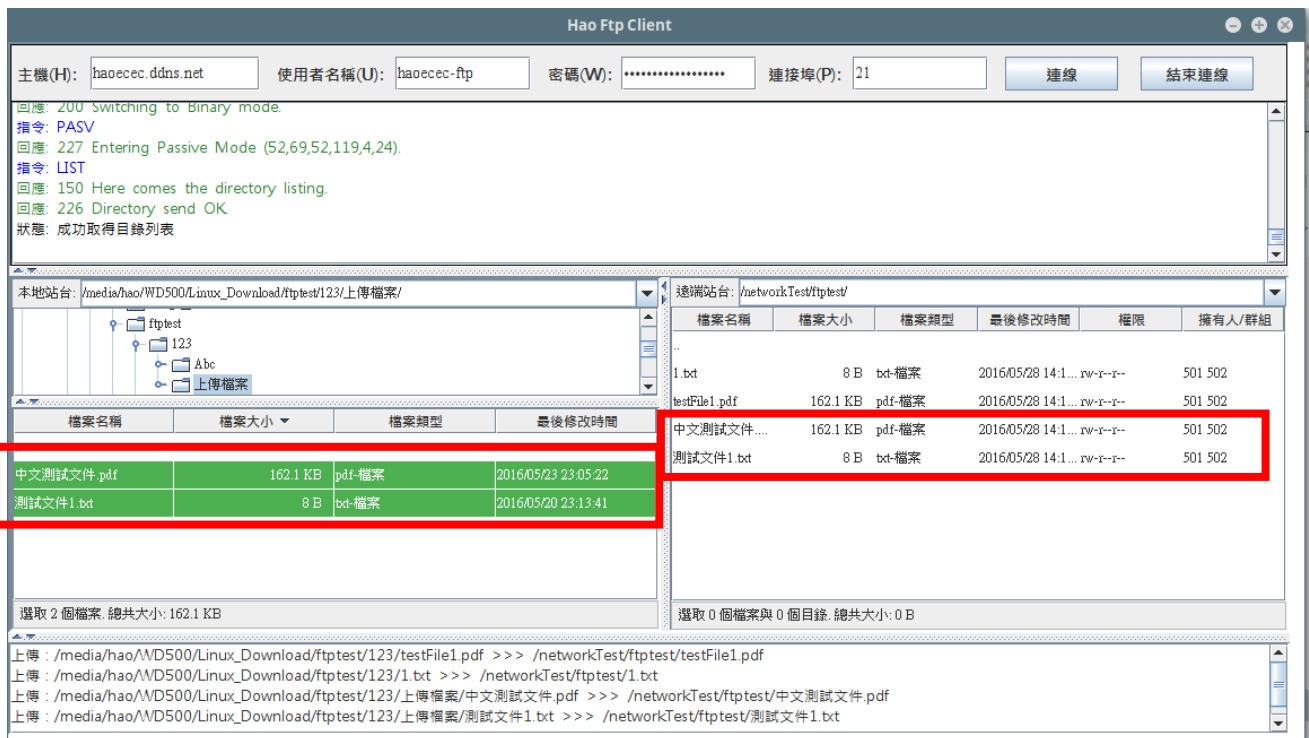


Fig. 8 以拖曳方式上傳多個選定之中文檔案

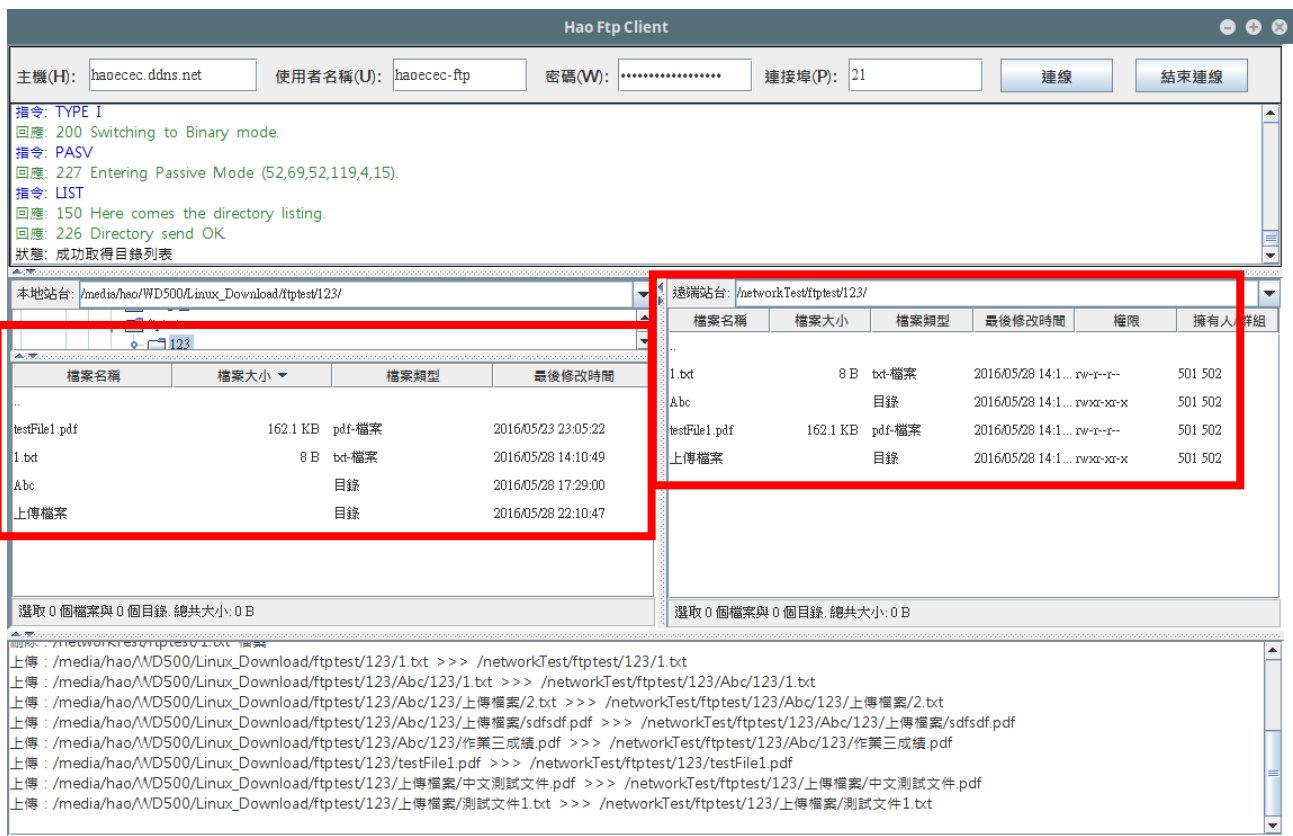


Fig. 9 上傳「123」資料夾至/networkTest/ftpctest/目錄下

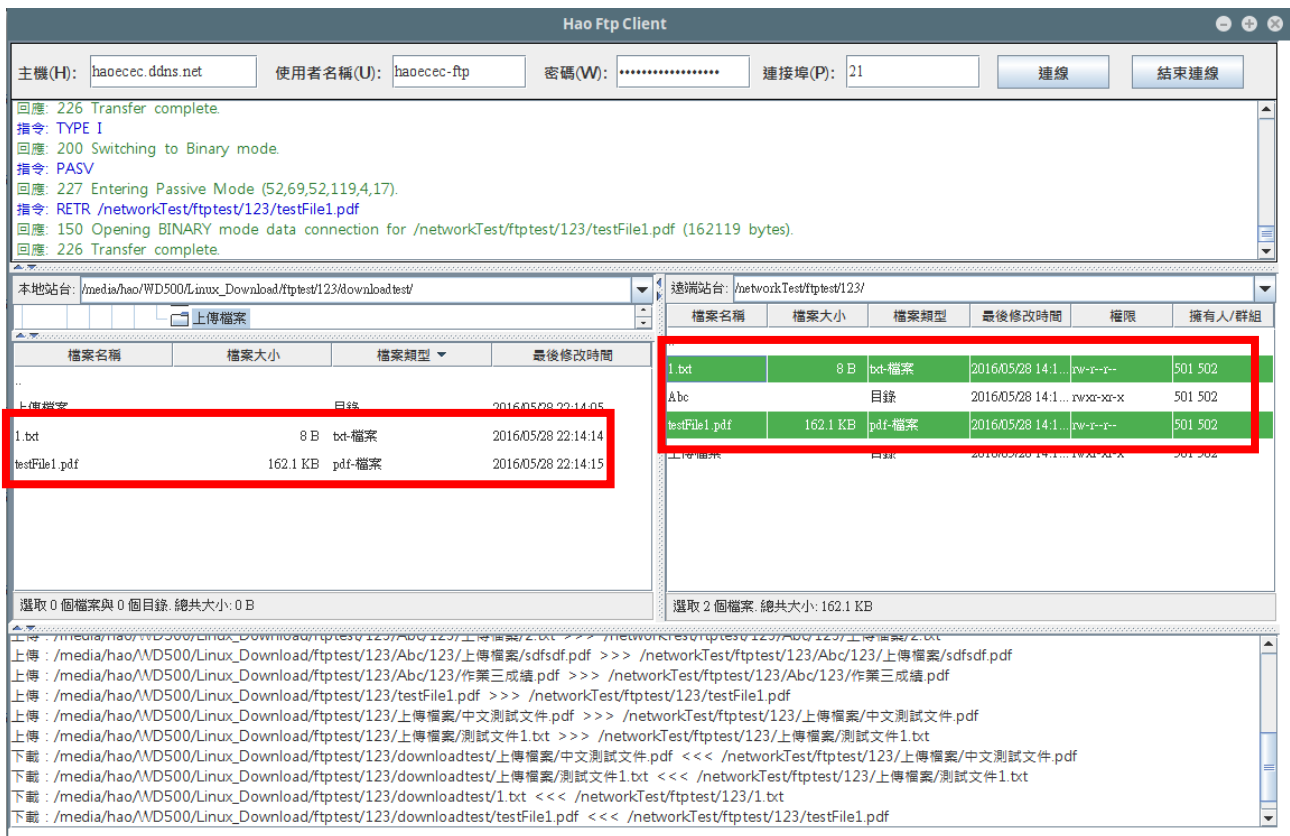


Fig. 10 下載英文、數字檔案至本地端目錄



Fig. 11 下載「123」資料夾至本地目錄

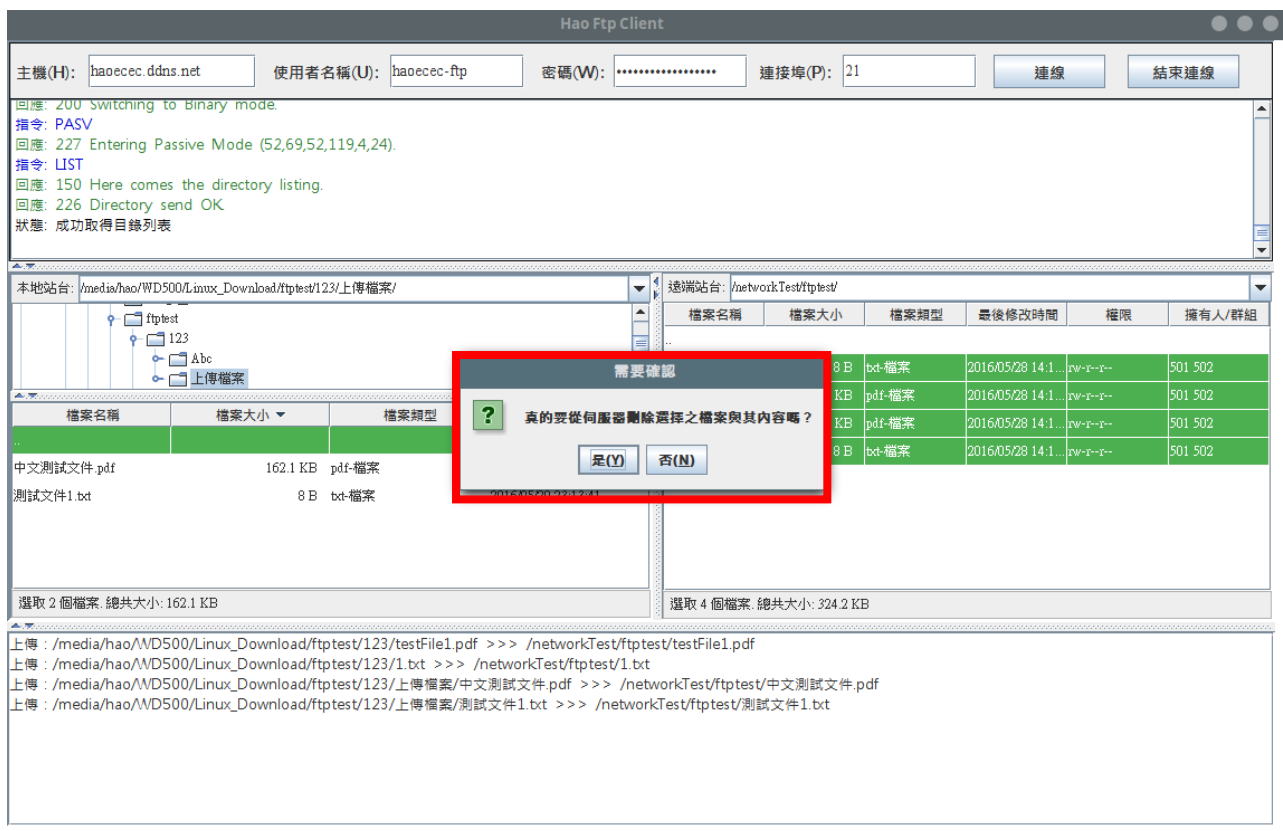


Fig. 12 提示是否刪除所選定之檔案及目錄

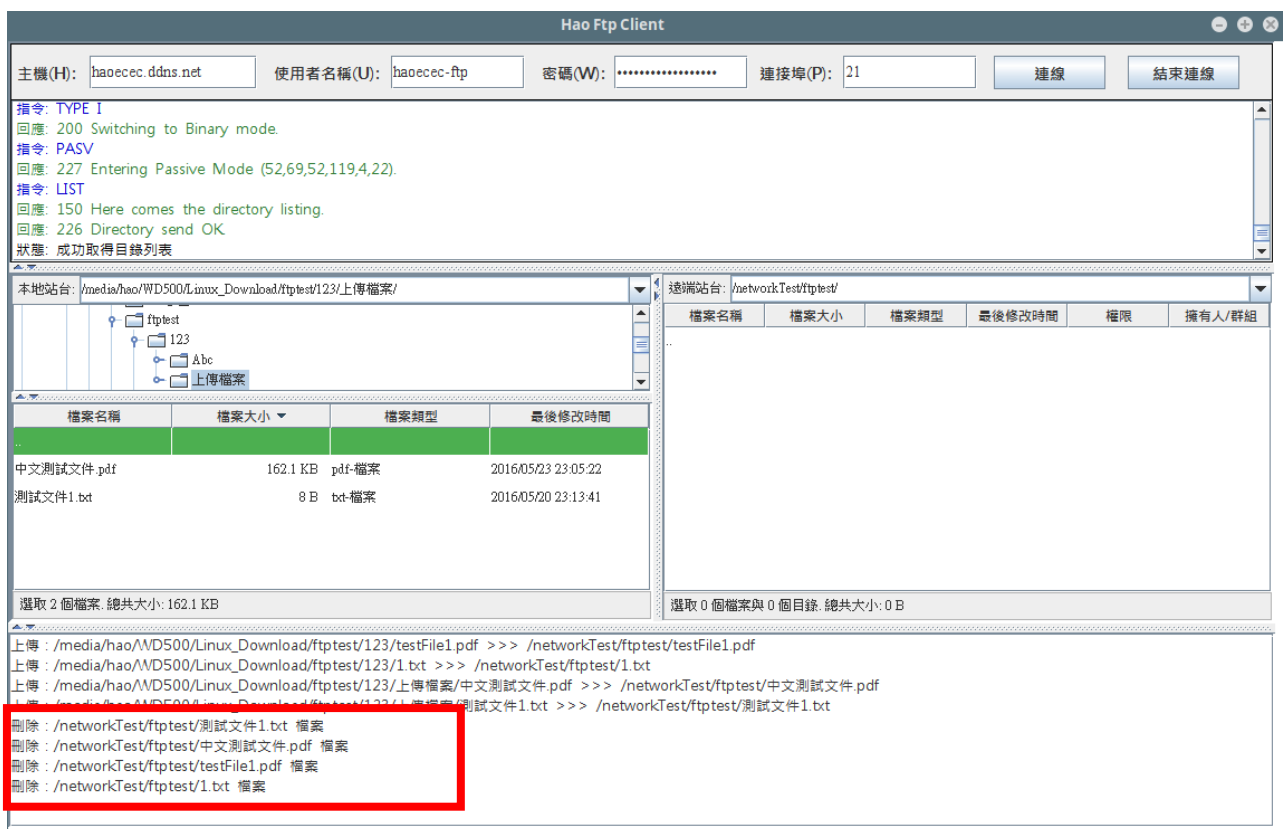


Fig. 13 成功刪除選定之檔案及目錄



Fig. 14 結束連接顯示 Server 回應「221 Goodbye」訊息

### 5-3、 Window 7 系統下之操作

本系統於 Window 7 下連接遠端以 Filezilla FTP Server 在 Window 7 系統上架設之 Server，顯示連線成功如 Fig. 15 所示，並且能夠選定多個檔案，不論是中文或是英文檔案皆能正常上傳如 Fig. 16 所示，然後是遠端下載中文或英文檔案皆能正常下載，以及刪除遠端檔案也正常操作，如 Fig. 17 所示，最後是結束連線，遠端 Server 正常回應訊息，如 Fig. 18 所示。

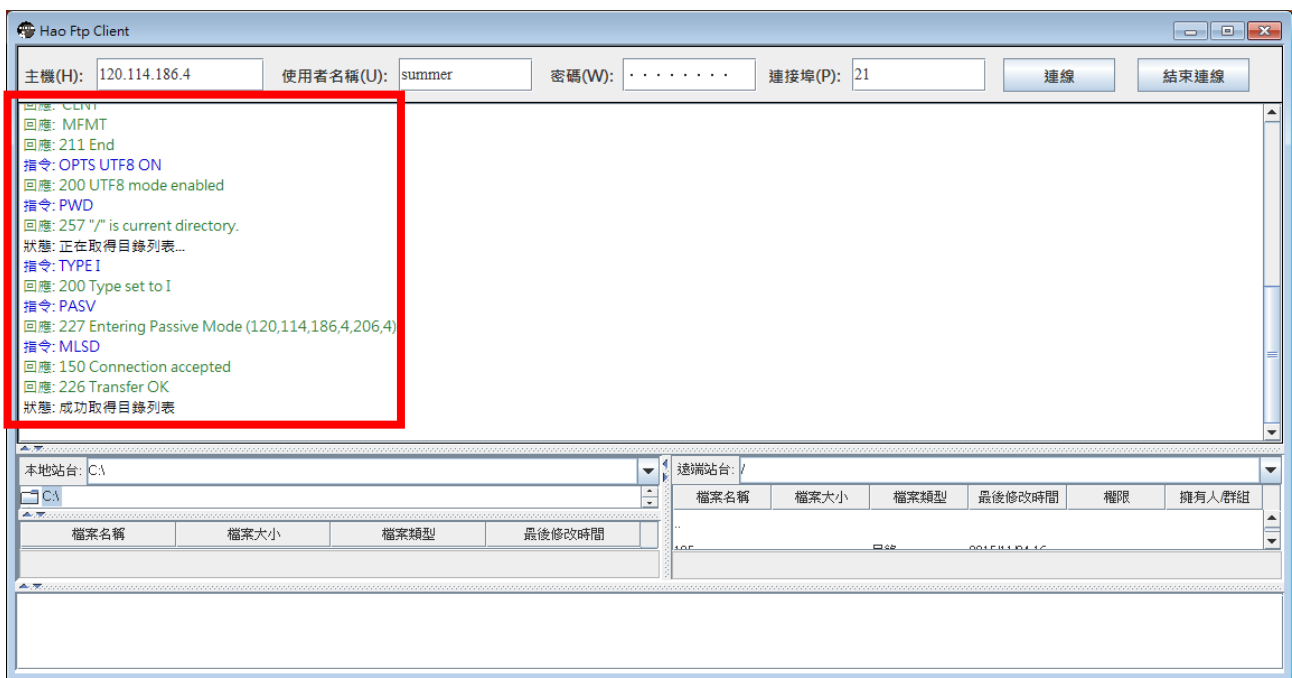


Fig. 15 成功連接遠端 Filezilla FTP Server 於 Window 7 系統上



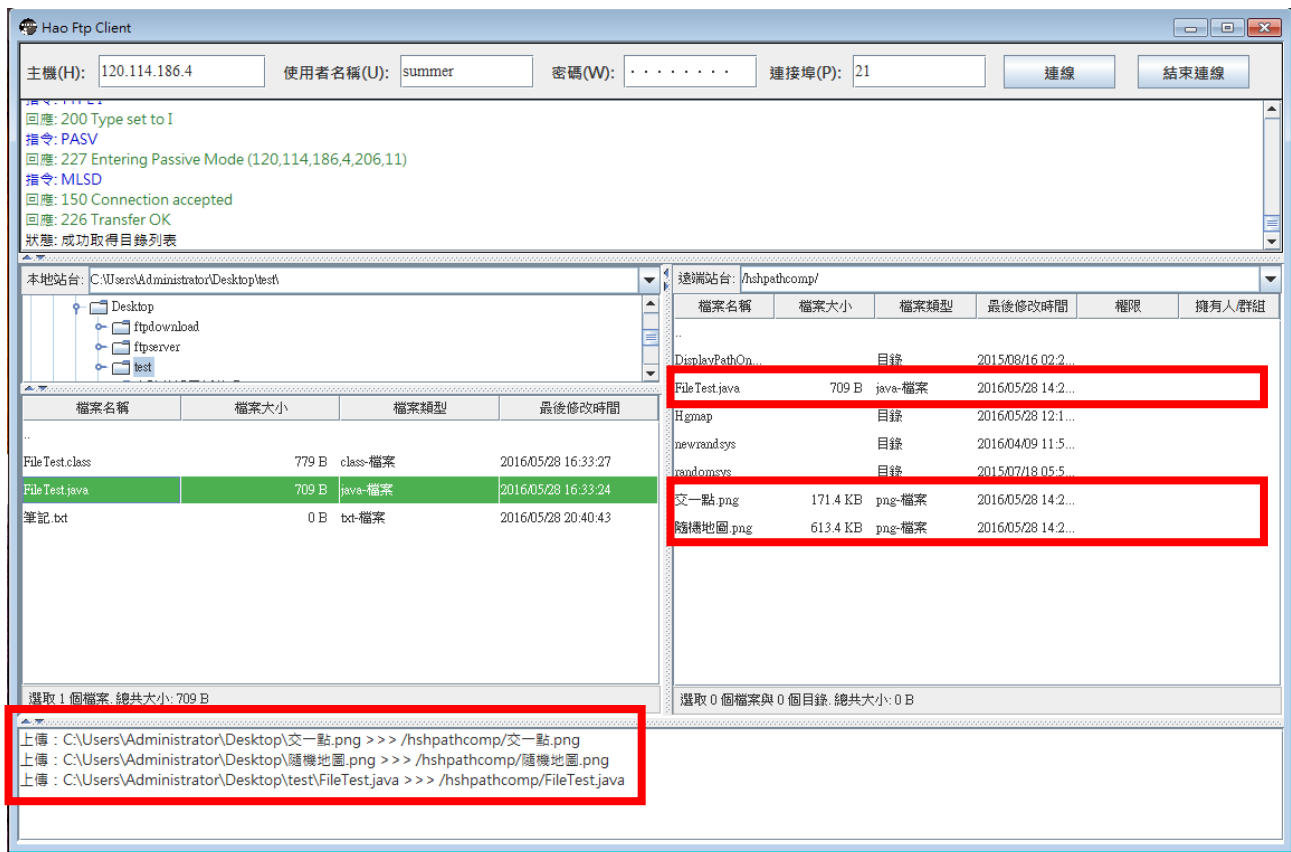


Fig. 16 上傳中、英文檔案名稱皆能成功

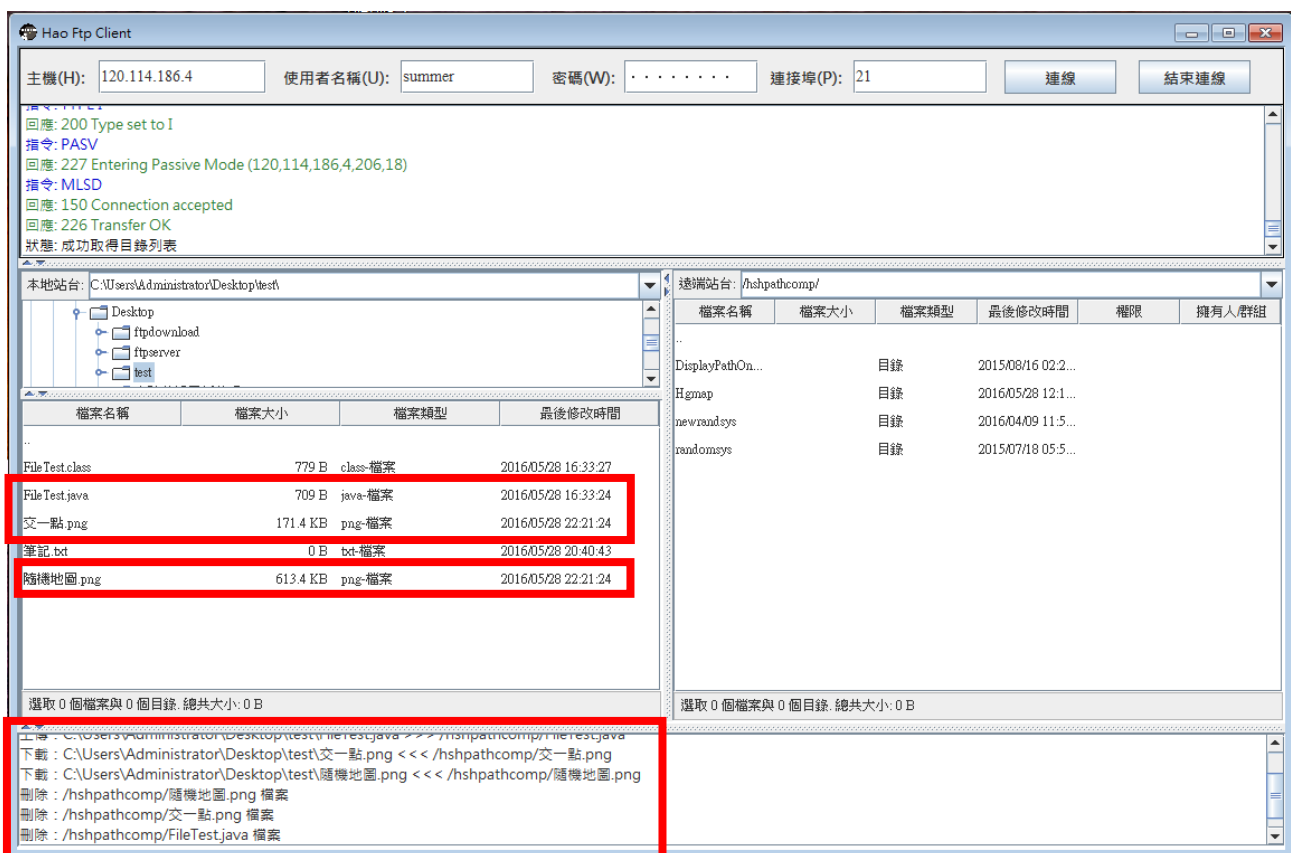


Fig. 17 下載與刪除遠端中、英文檔案皆能成功



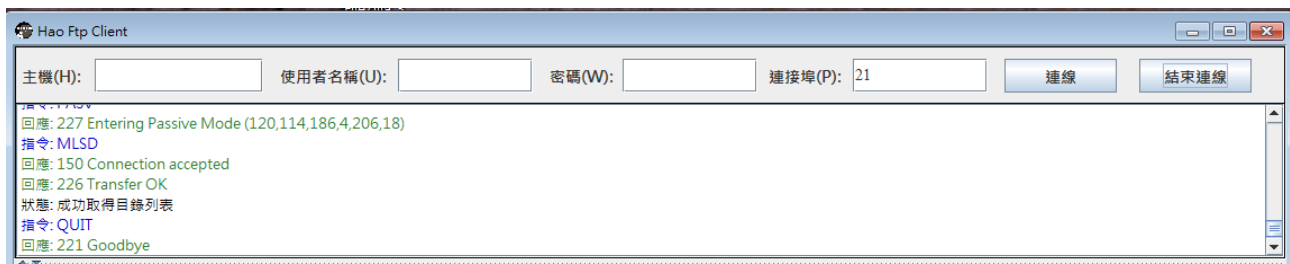


Fig. 18 結束連接 Server 顯示回應