

此 FTP 能模擬的功能與特色有以下：

1. 上傳檔案
2. 下載檔案
3. 列出 server 端資料夾的檔案資訊
4. 重新命名檔案(Extra)
5. 附有 Welcome 資訊，每次執行輸入指令時都會顯示適當、client 和 server 各自不同的提示資訊，包含進度、執行成功與否，並精心調整文字排版

程式實作部分：

有利用到的特殊 header file：

<stdio.h>	-> rename function
<sys/types.h>	->包含可能會用到的相關檔案型別
<winsock2.h>	->socket 相關函式庫
<dirent.h>	->directory 相關函式庫
<unistd.h>	->一些關於檔案讀寫的函式庫
<sys/stat.h>	->取得檔案資訊的函式庫
<time.h>	->將檔案時間轉換成 localtime

Server 端：

共包含七個 function，並限制單一 socket 最大容量為 1024 byte。

int main (int argc, char **argv)

開啟 server 前的前置作業，首先會利用 WSStartup 啟用 winsock，並確認版本有沒有符合，接著利用 serverSocket = socket(PF_INET, SOCK_STREAM, 0) 創建 serverSocket，並將 server 的 IP address 設為 INADDR_ANY(預設為 127.0.0.1)，port number 讀取 argv[1] (在 run cmd 時必須輸入 port number)。接著利用 bind 將 serverSocket 和 serverAddress 建好聯繫，並執行 listen。以上若都成功的話，則會跑出 server 端的歡迎資訊，然後等著 client 發出連線申請，利用 accept 接收，成功的話會顯示 client 端的 IP，並進入 **Connection_handler** function

void Connection_handler (SOCKET sockfd)

此為主宰連線的 main function，透過 recv(sockfd, command, MAX_SIZE, 0)接收來自 client 的 socket，利用'command'內的字串來判斷 client 送出何種指令，

分別為“exit”, “dir”, “get”, “put” 和 “rename”，並各自進入所負責的 function，若接收到非上述五種字串，則輸出 “Invalid Command”。

void Welcome_handler (SOCKET sockfd)

無特別功能，僅是在進入 Connection_handler 時 server 傳出歡迎資訊，用來模擬真實的 FTP 連線情況，以及作為第一次利用 send function，利用 socket 來傳送資訊的測試。

void File_listing_handler (SOCKET sockfd)

列出./upload 資料夾內的檔案資訊，當 command 為“dir”時進入。首先利用 DIR* pDir 紀錄 ./upload 的資料夾位置(pDir = opendir("./upload"))，接著透過 struct dirent* pDirent = readdir(pDir) 進行 while 迴圈，不斷的讀取 ./upload 下的所有檔案(為求簡單化略過上層和下層的資料夾不顯示)，並利用 socket 傳回檔名、檔案大小、最後修改時間。

取得檔案大小和修改時間，這裡要透過 stat function，先將檔案的完整路徑存在 buf，作為參數傳給 stat 便可得到檔案的眾多資訊 (此處只取用 file size & st_mtime 最後修改時間)，又因為 st_mtime 預設為從 1970 年算到現在的“秒數”，不符合現代人需求，再透過 localtime & strftime 將其轉換成我們熟悉的時間表示法。

void File_sending_handler (SOCKET sockfd, char filename[])

將檔案以 socket 的形式傳給 client，當 command 為“get”時進入。首先利用 FILE *fp = fopen(path, "rb")(path 為檔案完整路徑)，來開啟目標檔案，接著透過 fseek(fp, 0, SEEK_END) file_size = ftell(fp) rewind(fp) 等步驟取得檔案存取位置、file size 並傳回給 client 作為接收用。

傳送這邊因為限制 MAX_SIZE = 1024 byte，我們利用一個 while 迴圈將檔案切成一次不大於 1024 的 packet，經由 send socket 批次傳送給 client，透過變數 write_sum 來記錄檔案傳送完畢沒，讀取檔案則是利用 fread。等到傳送完成，藉由 sleep 的小停頓來模擬真實的 FTP，最後再傳回成功資訊。如果找不到檔案，則顯示錯誤資訊，並傳出能代表錯誤資訊“E_R_R_O_R”的 socket 給 client。

void File_receiving_handler (SOCKET sockfd, char filename[])

接受以 socket 形式來自 client 的檔案，當 command 為“put”時進入。同樣利用 FILE *fp = fopen(path, "wb")開啟“將來”的檔案位置(path 為之後寫入的地方)，接著等待 client 傳來的資訊，若 buf 為“E_R_R_O_R”即代表 client 找不到此檔案，無法上傳。如果接收到的是歡迎資訊，則接著接收 file size。跟下載一樣一次只能接收 1024 byte，透過變數 read_sum 檔案接收完畢沒，寫入檔案則是利用 fwrite。最後當檔案接收完成，會顯示 file size。

void File_Rename (SOCKET sockfd, char oldname[], char newname[]) (EXTRA)

額外功能，將目標檔案改成新檔名，當 command 為"rename"時進入。開啟資料夾和檔案的方式和 File_listing_handler function 一樣，只是尋找時利用 strcmp(pDirent->d_name, oldname) ==0 ?判斷是否為目標檔案，若有找到則透過 rename function，傳入舊檔案位置和新檔案位置(含檔名)來進行更名動作，如果回傳值為 1 代表更名成功，回傳成功資訊給 client。假使最後沒有找到此檔案，則回傳錯誤訊息給 client。

Client 端：

共包含四個 function，並限制單一 socket 最大容量為 1024 byte。

int main (int argc, char **argv)

與 server 雷同，先建立 clientSocket，然後將 serverAddress 設為 argv[1]、port number 設為 argv[2](在 run cmd 時必須輸入兩個參數)，接著 inet_pton 會測試是否為有效 IP，通過後利用 connect function 向 server 送出連線申請。

void Connection_handler(SOCKET sockfd)

此為主宰連線的 function，一開始會透過 recv 接受來自 server 端的歡迎資訊，接著會提示使用者"Enter The Command: "，合法的指令有"exit"，"dir"，"get"，"put" 和 "rename"，其餘皆視為 Invalid command 不予理會。因為 dir & rename 兩種指令 client 屬於被動等待 server 回應，並不另外設 function 處理，在 dir 時會依序接收 server 傳來的檔案資訊，做好排版後輸出在螢幕上，直到接收到特殊的訊息"o_v_e_r"，代表所有的檔案資訊皆取得。Rename 方面則是單純的送出兩次 socket，分別為 oldname & newname，之後等待 server 回應。

void File_download_handler (SOCKET sockfd, char filename[])

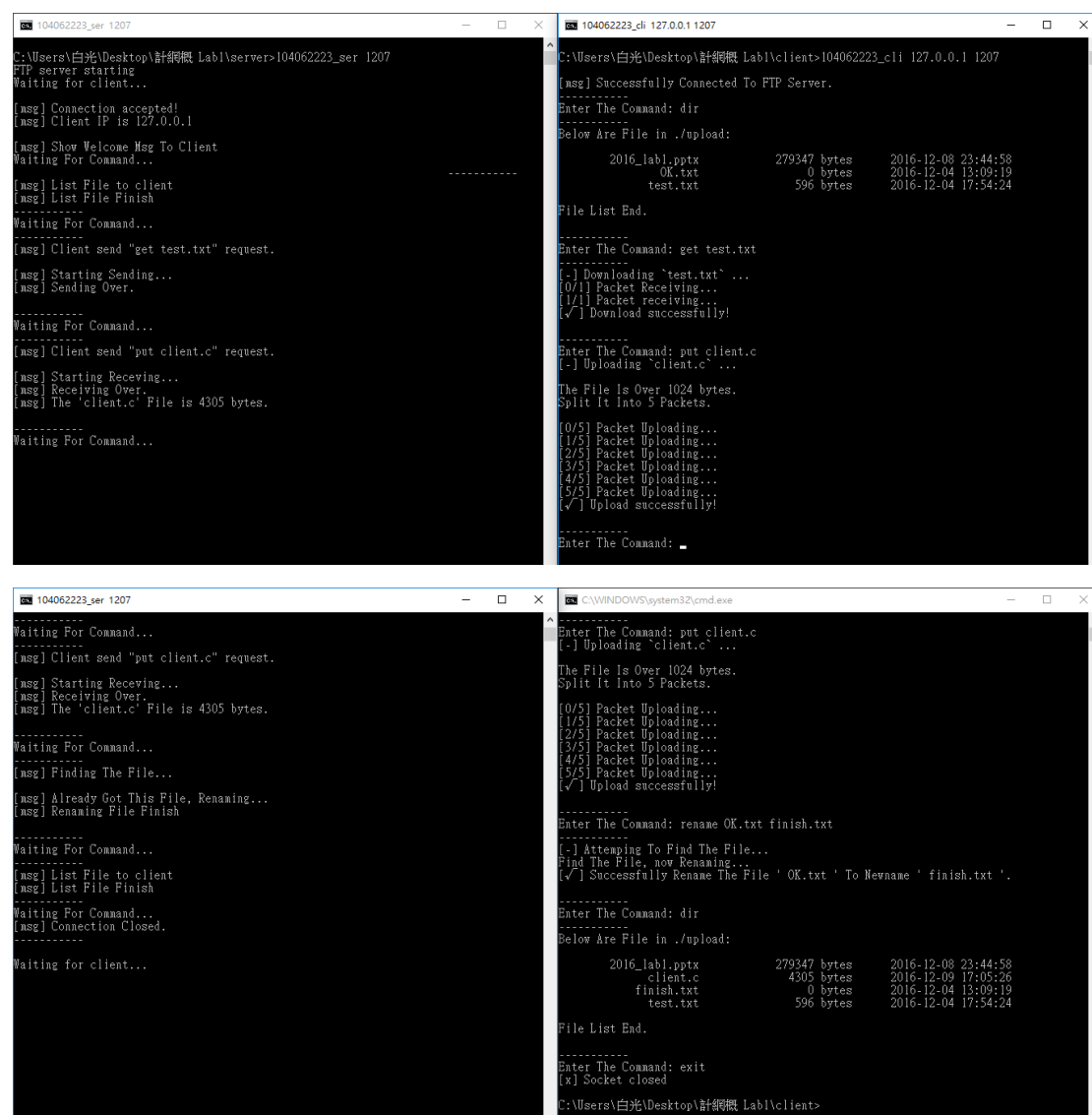
從 server 下載檔名為 filename 的檔案，當 command 為"get"時進入。基本上功能和 server 的 File_receiving_handler 一模一樣，這裡不贅述相同的功能，不一樣的主要是顯示資訊，client 會依據 file size，算出總共會切成幾個 packet，在接受時經由已接收的 byte size 來顯示[?/total num of pkts] Packet Receiving...，告訴使用者接收的進度。

void File_upload_handler (SOCKET sockfd, char filename[])

上傳檔名為 filename 的檔案，當 command 為"put"時進入。基本上功能和 server 的 File_sending_handler 也是一樣，這裡不贅述相同的功能，另外同 download，client 端會計算總共要切成幾個 packet 來上傳，並透過已傳送的 byte size 來顯示[?/total num of pkt] Packet Uploading...，告訴使用者上傳的進度。因為這裡 client 為主導方，成功和失敗的資訊由自己判斷後 print out，而非

由 server send socket 得知。

Screenshot：



```
104062223_ser 1207
C:\Users\白光\Desktop\計網課 Lab1\server>104062223_ser 1207
FTP server starting
Waiting for client...
[msg] Connection accepted!
[msg] Client IP is 127.0.0.1
[msg] Show Welcome Msg To Client
Waiting For Command...
[msg] List File to client
[msg] List File Finish
Waiting For Command...
[msg] Client send "get test.txt" request.
[msg] Starting Sending...
[msg] Sending Over.
Waiting For Command...
[msg] Client send "put client.c" request.
[msg] Starting Receiving...
[msg] Receiving Over.
[msg] The 'client.c' File is 4305 bytes.
Waiting For Command...

104062223_cli 127.0.0.1 1207
C:\Users\白光\Desktop\計網課 Lab1\client>104062223_cli 127.0.0.1 1207
[msg] Successfully Connected To FTP Server.
Enter The Command: dir
Below Are File in ./upload:
2016_lab1.pptx      279347 bytes    2016-12-08 23:44:58
OK.txt              0 bytes        2016-12-04 13:09:19
test.txt            596 bytes      2016-12-04 17:54:24
File List End.
Enter The Command: get test.txt
[-] Downloading 'test.txt' ...
[0/1] Packet Receiving...
[1/1] Packet receiving...
[✓] Download successfully!
Enter The Command: put client.c
[-] Uploading 'client.c' ...
The File Is Over 1024 bytes.
Split It Into 5 Packets.
[0/5] Packet Uploading...
[1/5] Packet Uploading...
[2/5] Packet Uploading...
[3/5] Packet Uploading...
[4/5] Packet Uploading...
[5/5] Packet Uploading...
[✓] Upload successfully!
Enter The Command:

104062223_ser 1207
Waiting For Command...
[msg] Client send "put client.c" request.
[msg] Starting Receiving...
[msg] Receiving Over.
[msg] The 'client.c' File is 4305 bytes.
Waiting For Command...
[msg] Finding The File...
[msg] Already Got This File, Renaming...
[msg] Renaming File Finish
Waiting For Command...
[msg] List File to client
[msg] List File Finish
Waiting For Command...
[msg] Connection Closed.
Waiting for client...

C:\WINDOWS\system32\cmd.exe
Enter The Command: put client.c
[-] Uploading 'client.c' ...
The File Is Over 1024 bytes.
Split It Into 5 Packets.
[0/5] Packet Uploading...
[1/5] Packet Uploading...
[2/5] Packet Uploading...
[3/5] Packet Uploading...
[4/5] Packet Uploading...
[5/5] Packet Uploading...
[✓] Upload successfully!
Enter The Command: rename OK.txt finish.txt
[-] Attempting To Find The File...
Find The File, now Renaming...
[✓] Successfully Rename The File ' OK.txt ' To Newname ' finish.txt '.
Enter The Command: dir
Below Are File in ./upload:
2016_lab1.pptx      279347 bytes    2016-12-08 23:44:58
client.c            4305 bytes      2016-12-09 17:05:26
finish.txt           0 bytes         2016-12-04 13:09:19
test.txt            596 bytes      2016-12-04 17:54:24
File List End.
Enter The Command: exit
[✗] Socket closed
C:\Users\白光\Desktop\計網課 Lab1\client>
```

所遇到的問題與心得：

問題最多的大概就是如何找到合適的 **function**，過去寫程式很少接觸到 **File** 讀取、寫入的部分，像是 **fread**, **fwrite** 對我而言都極為陌生，更別提關於 **dirent.h** 和 **stat.h** 這兩個根本沒聽過的 **header** 檔(基本上除了 **winsock2.h** 的 **header** 檔都是 **Google** 相關功能而學來的)，像是 **dir** 的部分，光是找到 **Dir** 這個型別就花了不少時間，而後續的 **stat**、**localtime** 轉換又是另一段掙扎。

但這些 **function** 在實作真正能應用的程式上，想必都是相當重要且基本的函式，雖然因為時間的問題，大部分的函式我還是僅停留在「會用」，對實際的原理一知半解的情形，但這次的 **lab** 對於 **socket** 的使用方式，以及實際 **File** 相關操作的程式方面，確實收穫不少。