

# Introduction to Computer Networks Lab1 Report

104021219 鄭余玄

## 1 使用說明

規格基本上是參考助教提供的規格書，細節列在實做過程中。而完整的使用方法和編譯方式都有使用 markdown 語法寫在 104021219\_readme.txt 檔案中。

### 1.1 使用方法

1. For socket server,

```
./104021219_ser.exe <Port>
```

2. For socket client:

```
./104021219_cli.exe <IPv4> <Port>
```

註：缺少任何參數皆會顯示錯誤，並且不能執行。

### 1.2 支援指令

1. `put <filename>`
2. `get <filename>`
3. `dir`
4. `rename <old filename> <new filename>`

### 1.3 編譯環境

1. Windows x64
2. Visual Studio 14.0
3. C++

## 2 實做過程

不論 client 或是 server 程式，一開始都需要檢查 argc 和 argv 是否有足夠的參數可以啟動，此外我有做 port 範圍的驗證檢查，若有違反規則的，或是 Winsock 建立錯誤，則程式就不會啟動。

在 server 端的部份，因為所有操作都是在 ./upload 資料夾下，所以程式一啟動就直接創立並且切換到資料夾下，之後就可以開始監聽 client 端的封包，若是指令封包，則會顯示 client IP 執行指令後回傳的結果。此外，對於 socket 傳送或接收超過 MAX\_SIZE 大小的檔案，只需要使用 while 迴圈不斷接收封包，並且寫入或讀取到目前的檔案內容指標。

在 client 端的部份則是，每次讀取一行使用者輸入的指令，並且判斷屬於哪個規範的類型，之後再依據不同的操作，向 server 傳送或接收不同的檔案和切換資料夾。此外，透過計算 socket 傳輸檔案的 byte stream 大小，就可以知道檔案大小。

put 的實做方式是，先檢查 client 是否有此檔案，若無則拒絕操作。若有則再詢問 server 是否有此檔案名稱，若無則 server 回傳 WTF；如果有，則 server 回傳 LGTM，並且 client 詢問是否要覆蓋 server 端檔案。在檢查和確認完成之後，如果沒有要繼續 put 會送出 G8，若有則會送出 OK，並且直接開始傳送檔案。

get 的實做方式與 put 有點類似，先檢查 client 是否有此檔案，若有則詢問是否要覆蓋 client 端檔案，若不覆蓋則操作失敗。若有則再詢問 server 是否有檔案，若無則 server 回傳 WTF；如果有，則 server 回傳 LGTM，並且操作失敗。檢查完成之後，也是一樣直接開始傳輸檔案。

dir 的實做方式則是，在 server 端使用 system 函數下 dir 指令並且將資料流重新導向置 GBY.txt 這個暫存檔案，避免資料夾顯示內容過多，就可以透過類似 get 指令傳輸到 client 端之後就可以刪除。client 讀取所有檔案內容後，經過部份內容處理後，就直接輸出結果，並且刪除 GBY.txt。因為 server 和 client 的 GBY.txt 是創立在 upload 和 download 資料夾外，所以不會有重複檔案名稱的問題。

rename 的實做方式是，server 會確認是否有舊檔案名稱，以及重新命名的檔案名稱是否有重複。若可以，就使用 system 函數下 ren 直接重新命名，並且回傳 G8。

### 2.1 實驗

在圖 2.1 和圖 2.1 中，使用了 703647 bytes 的純文字檔和 4137691 bytes 的 mp3 音檔來測試 socket 傳輸，而結果皆正確並且完全符合預期。

## 3 學到的東西及遇到的困難

因為我的開發環境是 Visual Studio 14.0，因此在引入 Winsock 時需要在程式碼中加上編譯參數：

```
#pragma comment(lib, "Ws2_32.lib")
```

此外都是在複習 c++ 讀寫檔案的一些方法，基本上沒有遇到特別的困難。

```

D:\Visual Studio 2015\Projects\104021219\Debug\104021219_ser.exe
Listening...
[127.0.0.1] dir success.
[127.0.0.1] put test.txt success (703647 bytes).
[127.0.0.1] get 111.txt success (703647 bytes).
[127.0.0.1] rename success.
[127.0.0.1] dir success.

D:\Visual Studio 2015\Projects\104021219\Debug\104021219_cli.exe
12/09/2016 12:31 PM <DIR> .
12/09/2016 12:31 PM <DIR> 703,647 111.txt
12/09/2016 11:35 AM 703,647 123.txt
12/09/2016 12:07 PM 703,647 12333.txt
12/09/2016 11:35 AM 703,647 1234.txt
12/09/2016 12:04 PM 4,137,691 12345.mp3
12/05/2016 03:49 PM 4,137,691 123456.mp3
12/07/2016 11:26 PM 703,647 123456789.txt
12/07/2016 08:25 PM 703,647 321.txt
12/09/2016 12:12 PM 4,137,691 6666.mp3
9 File(s) 16,634,955 bytes
2 Dir(s) 773,038,694,400 bytes free

put test.txt
put test.txt success (703647 bytes).
get 111.txt
get 111.txt success (703647 bytes).
rename 1234.txt abc.txt
rename 1234.txt abc.txt success.
dir
Directory of server:./upload/
12/09/2016 05:57 PM <DIR> .
12/09/2016 12:31 PM <DIR> 703,647 111.txt
12/09/2016 11:35 AM 703,647 123.txt
12/09/2016 12:07 PM 703,647 12333.txt
12/09/2016 12:04 PM 4,137,691 12345.mp3
12/05/2016 03:49 PM 4,137,691 123456.mp3
12/07/2016 11:26 PM 703,647 123456789.txt
12/07/2016 08:25 PM 703,647 321.txt
12/09/2016 12:12 PM 4,137,691 6666.mp3
12/09/2016 11:35 AM 703,647 abc.txt
12/09/2016 05:55 PM 703,647 test.txt
10 File(s) 17,338,602 bytes
2 Dir(s) 773,142,126,592 bytes free

```

Figure 1: Basic

```

Listening...
[127.0.0.1] dir success.
[127.0.0.1] put test.txt success (703647 bytes).
[127.0.0.1] get 111.txt success (703647 bytes).
[127.0.0.1] rename success.
[127.0.0.1] dir success.
[127.0.0.1] put test.txt fail.
[127.0.0.1] put test.txt success (703647 bytes).
[127.0.0.1] get 12345.mp3 success (4137691 bytes).
[127.0.0.1] get 12345.mp3 success (4137691 bytes).
[127.0.0.1] Invalid: file.
[127.0.0.1] Invalid: file.
[127.0.0.1] rename success.

D:\Visual Studio 2015\Projects\104021219\Debug\104021219_cli.exe
12/07/2016 11:26 PM 703,647 123456789.txt
12/07/2016 08:25 PM 703,647 321.txt
12/09/2016 12:12 PM 4,137,691 6666.mp3
12/09/2016 11:35 AM 703,647 abc.txt
12/09/2016 05:55 PM 703,647 test.txt
10 File(s) 17,338,602 bytes
2 Dir(s) 773,142,126,592 bytes free

put test.txt
Overwrite test.txt in server? [Y/n]n
put test.txt
Overwrite test.txt in server? [Y/n]Y
put test.txt success (703647 bytes).
put test123.txt
Invalid: test123.txt is not found!
get 12345.mp3
get 12345.mp3 success (4137691 bytes).
get 12345.mp3
Overwrite current 12345.mp3 file? [Y/n]Y
get 12345.mp3 success (4137691 bytes).
rename zzz.txt aaa.txt
Invalid: rename zzz.txt aaa.txt.
rename 111.txt 123.txt
Invalid: rename 111.txt 123.txt.
rename 111.txt zzz.txt
rename 111.txt zzz.txt success.

```

Figure 2: Error Handling