

【高等電腦網路作業四】

Deadline : 2012/12/28

主旨：

請各位同學實作一個類似 FTP 的網路系統，系統是 multi client-server 架構的網路程式。請在 UDP 通訊協定上實作一支以 stop-and-wait 為基礎的檔案傳輸程式，另外還要以 checksum 以及 resend 實作 error control。

題目：

myftp.c myftp.h myftpClient.c myftpServer.c

根據作業提供的檔案，設計 myftp.c 程式，實作 myftp.h 所定義的函示。myftpServer 作為提供檔案下載的 server，myftpClient 為連繫 server 下載檔案的程式。

Part1：

1. 以 **Superuser** 權限執行 server，因為 Server 可能會有多張網卡，會收到重複的廣播訊息，因此 Server 需要綁定網卡才能確保執行正確。

2. 啟動畫面：

Server：

```
sa@kaohsiung:~/hw/4$ sudo ./myftpServer 44003 testfile
network interface = eth0
network port = 44003
MyFtp Server start!
share file : testfile
wait client!
```

Client：

```
sa@tainan:~$ ./myftpClient 44003 testfile
```

※port 為 XX+學號末三碼，大學部 XX 為 22，碩一 XX 為 33，碩二 XX 為 44，博班 XX 為 55

※testfile 為所要傳送或接收的檔案名稱

3. 程式需設計 **timeout** 機制，時間為三秒鐘，不論 Client 端或是 Server 端皆須設置：

Server：(傳檔時)

```
file transmission start
wait client time out
```

Client：(連線時)

```
sa@tainan:~$ ./myftpClient 44003 testfile
no server answer!
```

Client：(下載時)

```
download to file : <client_testfile>
wait server time out
```

4. Server 流程：

a. 取得裝置名稱並綁定：

`getIFname()` `initServAddr()`

```
sa@kaohsiung:~/hw/4$ sudo ./myftpServer 44003 testfile
network interface = eth0
network port = 44003
Myftp Server start!
share file : testfile
wait client!
```

b. 監聽，並且在收到廣播訊息(`struct bootServerInfo`)後，查找 client 所要求的檔案。存在的話則回傳 server 位址和用於傳送的 port 號：

`listenClient()`

```
share file : testfile
wait client!
filename:testfile
buff.connectPort : 44843
buff.servAddr : 140.117.170.97
client from 140.117.170.95 connect!!
myftp connect port : 44843
```

※傳送的 port 號計算規則：原本的 port 上加上 1~999 的亂數

c. 如果建立連線後則依照指定流程(註 1)傳送檔案：

`startMyftpServer()`

※DATA 封包中所設定的資料大小需控制在 512 bytes

※送出一份 DATA(block=1)必須收到 ACK(block=1)才可進行下一筆資料傳遞

※封包需要配合使用 checksum(註 2)的認證，如果結果不正確則要回傳 Error 要求重傳

※Server 需要在一個 client 傳送檔案時，同時也能接受新的 client 連入

5. Client 流程：

a. 設置 client：

`initCliAddr()`

b. 丟出廣播訊息(`struct bootServerInfo`)查詢有相對應檔案的 server：

`findServerAddr()`

```
buff = testfile
sendto ready!
find MyftpServer servaddr : 140.117.170.97
Myftp connectPort : 44786
```

c. 找到則依照指定流程(註 1)發送要求並接收檔案：

startMyftpClient()

※封包需要配合使用 checksum(註 2)的認證，如果結果不正確則要回傳 Error 要求重傳

※下載後的檔案請加入 **client_**在開頭，例如：testfile -> client_testfile

6. 封包格式：

Packet	Opcode	Checksum	
FRQ	FRQ(2)	Checksum(2)	filename
DATA	DATA(2)	Checksum(2)	Block() Data(512)
ACK	ACK(2)	Checksum(2)	Block()
ERROR	ERROR(2)	Checksum(2)	Block()

7. 最終參考執行畫面：

Server：

```
sa@kaohsiung:~/hw/4$ sudo ./myftpServer 44003 testfile
[sudo] password for sa:
MyFtp Server start!
share file : testfile
wait client!
client from 140.117.170.95 connect!!
myftp connect port : 44843
share file : testfile
wait client!
file transmission start
send file : <testfile> to 140.117.170.95
2542707 bytes sent
file transmission finish!!
```

Client：

```
sa@tainan:~$ ./myftpClient 44003 testfile
find MyftpServer servaddr : 140.117.170.97
Myftp connectPort : 44843
file transmission start!!
download to file : <client_testfile>
get file : <testfile> from 140.117.170.97
2542707 bytes received
file transmission finish!!
```

註 1：

一個資料大小需要送多次 DATA 的流程

Client		Server	
Sendto	->	Recvfrom	//Client 廣播找 Server
Recvfrom	<-	Sendto	//Server 回傳
<hr/>			
Sendto(FRQ)	->	Recvfrom	//傳遞要求
Recvfrom	<-	Sendto(DATA, block=1)	//開始傳送
Sendto(ACK, block=1)	->	Recvfrom	
Recvfrom	<-	Sendto(DATA, block=2)	
Sendto(ACK, block=2)	->	Recvfrom	
...			
Recvfrom	<-	Sendto(DATA, block=n)	
Sendto(ACK, block=0)	->	Recvfrom	//收完檔案

※Client 收到的封包中，DATA 容量 < 512bytes，表示為檔案已經傳送完成，回傳 block = 0 的 ACK 給 server。

註 2：

封包送出前須先將 checksum 欄位填 0，再經由 checksum 演算法運算後回填。接收時則直接將封包大小進行 checksum 運算，正確結果為 0，反之則為錯誤封包。詳細可參考 ICMP。簡單來說則是將封包以每兩個 bytes 為單位取出並相加，在將總和轉成 1 補數。

※ACK,/ERROR 封包用 size = 6 計算

※DATA 以 DATA size + 6 計算

※FRQ 以 filename string size + 4 計算

註 3：

提供的檔案中所下的註解，以及函示的寫法為一個參考依據。如有自己的寫法也可，但資料交換的流程請務必遵照註 1 的方式，否則 Part2 Part3 將無法順利執行。

Part2：

請讓 FTP 程式能傳送大型檔案，請自行使用 32M 以上的檔案測試

Part3 :

與助教的 Server 進行連線。

※請確定自己本身的 Client-Server 連線沒問題再進行與助教的連線測試

※Port 號預設為 44027，檔案名稱預設為 testfile

※testfile 大小為 35597268

限制：

- (1) 請在 Ubuntu 系統上使用 C 語言寫本次作業並進行測試，Demo 時只接受助教所指定之 Ubuntu 作業伺服器上的執行結果。
- (2) 本作業必須上傳能編譯本作業和刪除編譯後額外產生的檔案之 Makefile，內容不拘，未寫要扣分，make 時如有出現 error 也會扣分。
- (3) 嚴禁抄襲其他同學作業，參與者(抄襲與被抄襲)均以零分計算。
- (4) 可擷取書籍的範例程式，若 Demo 時助教有問，請指出為哪些書籍。
- (5) 請對你的程式碼有深入瞭解，助教會問。
- (6) 對題目有問題可以在討論區上詢問，e-mail 恕不受理也不幫忙 debug。