

【高等電腦網路作業四】

Deadline : 2011.11.09

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，使用UDP協定，程式練習的部份包含 BootP 和 TFTP 概念，程式實作方法以 stop-wait 為基礎，另外還有 checksum和 resend 機制來作 error control 的練習。

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可能會有許多張網卡，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()`

```
network interface = eth0
network port = 44003
```

- 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!!
```

註一：

一個資料大小需要重送兩次DATA的流程

| Client | | Server | |
|----------------------|----|-----------------------|------------------|
| Sendto | -> | Recvfrom | //Client找尋Server |
| Recvfrom | <- | Sendto | //Server回傳 |
| 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=0) | -> | Recvfrom | //收完檔案 |

註二：

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

註三：

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一的方式，否則Part2將無法順利執行。

Part2

1. 請讓FTP程式能傳送大型檔案，自行測試請使用32M以上

Part3

1. 跟助教的Server可以進行連線。

※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44003，檔案名稱預設為testfile > 32 M

※testfile大小為35597898

※如要查詢Server的訊息，請ssh/sftp登入140.117.170.9，使用帳密為：ftpTest/123，針對不同登入的IP有不同的紀錄檔，以IP為檔名。

※由於容量有限，所以只記錄當次執行的訊息，如有需要請用自己主機備份

※記錄檔格式為<time> : <IP> <R or S or :> <message>

<Wed Oct 26 18:25:04 2011> : 140.117.170.95 : file transmission start

<Wed Oct 26 18:25:05 2011> : 140.117.170.95 R client send a FRQ

<Wed Oct 26 18:25:05 2011> : 140.117.170.95 S send first DATA(512) to client, block = 1

: 代表一般訊息，R代表有來自Client的動作，S代表Server即將送封包前的訊息

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。