

網路程式設計 HW-傳送接收檔案作業

一、作業

本次作業採 Pair programming 方式進行，我的夥伴是 108xxxx XXX。我們採取邊觀看影片邊實作的方式來撰寫這次的作業，前四題的部分，主要由她撰寫，而我負責檢查，後四題的部分主要由我進行撰寫，而她負責檢查。

二、執行畫面

以下所有執行畫面左方視窗皆為 Server 端的執行畫面，右方視窗皆為 Client 端的執行畫面。

練習 1：

問題：請寫一組程式來測試若兩方都先送後收，可以運作嗎？

回答：可以！

```

C:\data\Code\Cpp\network_programming\cmake-build-debug (main -> origin)
λ .\hm8_exercise1_1_server.exe
sd=360
bind(): 0
listen(): 0
send:I love Algorithm!! (0)
recv:I love NP!! (11)
Press any key to continue . . .

C:\data\Code\Cpp\network_programming\cmake-build-debug (main -> origin)
λ

C:\data\Code\Cpp\network_programming\cmake-build-debug (main -> origin)
λ .\hm8_exercise1_1_client.exe
sd=188
connect():188
send:I love NP!! (11)
recv:I love Algorithm!! (18)
Press any key to continue . . .

C:\data\Code\Cpp\network_programming\cmake-build-debug (main -> origin)
λ

```

問題：請寫一組程式來測試若兩方都先收後送，可以運作嗎？

回答：不行，因為兩方都在等待別人先傳送訊息！

```

C:\data\Code\Cpp\network_programming\cmake-build-debug (main -> origin)
λ .\hm8_exercise1_2_server.exe
sd=388
bind(): 0
listen(): 0

C:\data\Code\Cpp\network_programming\cmake-build-debug (main -> origin)
λ .\hm8_exercise1_2_client.exe
sd=372
connect():372

```

練習 2：

完成「送一次，收多次」的實驗。

```

C:\data\Code\Cpp\network_programming\cmake-build-debug (main -> origin)
λ .\hm8_exercise2_1_server.exe
Server waits.
Server receive:[NCYU], byte:6
Server receive:[NCYU], byte:6
Server receive:[NCYU], byte:6
Server receive:[NCYU], byte:6
Server receive:[NCYU], byte:6
Press any key to continue . . .

C:\data\Code\Cpp\network_programming\cmake-build-debug (main -> origin)
λ

C:\data\Code\Cpp\network_programming\cmake-build-debug (main -> origin)
λ .\hm8_exercise2_1_client.exe
Press any key to continue . . .

C:\data\Code\Cpp\network_programming\cmake-build-debug (main -> origin)
λ

```

完成「送多次，收一次」的實驗。

```

C:\data\Code\Cpp\network_programming\cmake-build-debug (main -> origin)
λ .\hw8_exercise2_2_server.exe
Server waits.
Server receive:[NCYU][NCYU][NCYU][NCYU], byte:30
Press any key to continue . . .
C:\data\Code\Cpp\network_programming\cmake-build-debug (main -> origin)
λ

C:\data\Code\Cpp\network_programming\cmake-build-debug (main -> origin)
λ .\hw8_exercise2_2_client.exe
Press any key to continue . . .
C:\data\Code\Cpp\network_programming\cmake-build-debug (main -> origin)
λ |
  
```

練習 3：

完成 UDP 成對收送的實驗。

```

C:\data\Code\Cpp\network_programming\cmake-build-debug (main -> origin)
λ .\hw8_exercise3_server.exe
Server waits.
Server receive:[NCYU], byte:6
Server receive:[NCYU], byte:6
Server receive:[NCYU], byte:6
Server receive:[NCYU], byte:6
Server receive:[NCYU], byte:6
C:\data\Code\Cpp\network_programming\cmake-build-debug (main -> origin)
λ

C:\data\Code\Cpp\network_programming\cmake-build-debug (main -> origin)
λ .\hw8_exercise3_client.exe
Press any key to continue . . .
C:\data\Code\Cpp\network_programming\cmake-build-debug (main -> origin)
λ |
  
```

練習 4：

完成單次傳送文字檔的實驗。

```

C:\data\Code\Cpp\network_programming\cmake-build-debug (main -> origin)
λ .\hw8_exercise4_3_server.exe
sd=304
bind(): 0
listen(): 0
recv:123456
(7)
Press any key to continue . . .
C:\data\Code\Cpp\network_programming\cmake-build-debug (main -> origin)
λ |

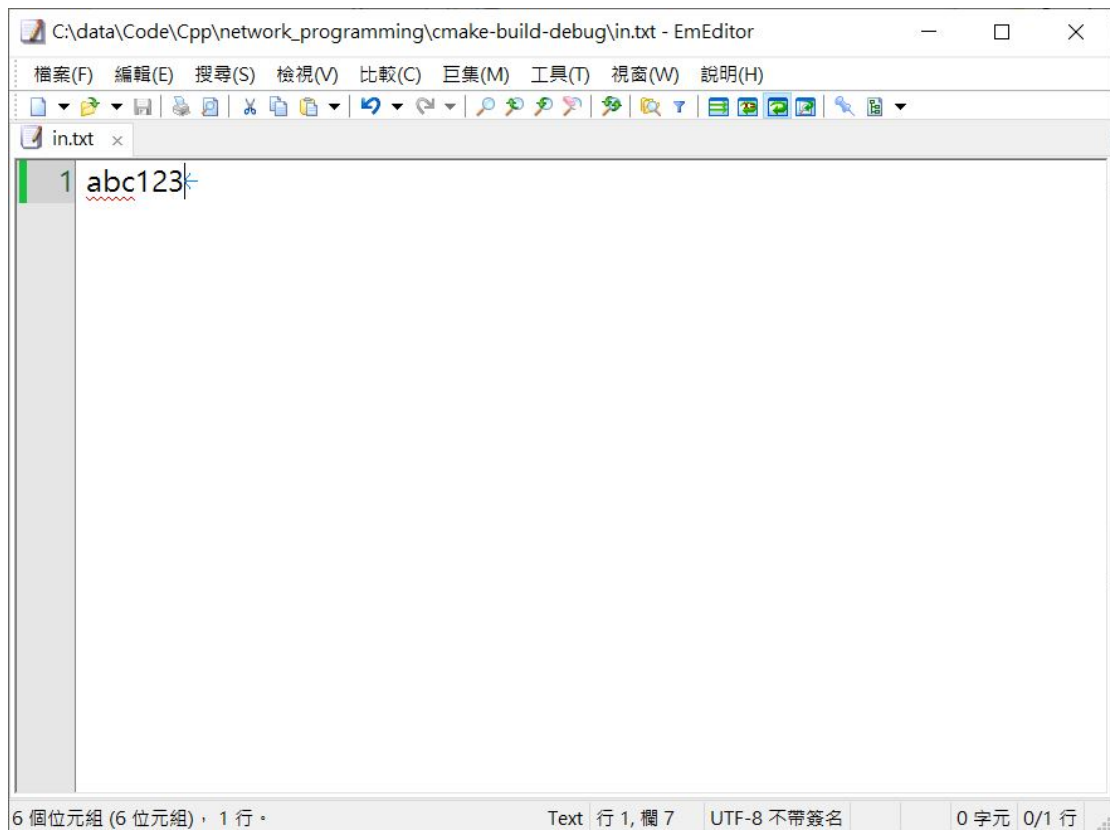
C:\data\Code\Cpp\network_programming\cmake-build-debug (main -> origin)
λ .\hw8_exercise4_3_client.exe
len:7
123456
sd=132
connect():0
send:123456
(7)
Press any key to continue . . .
C:\data\Code\Cpp\network_programming\cmake-build-debug (main -> origin)
λ
  
```

檢驗傳送的文字檔 hash 值與接收到的文字檔 hash 值相同。

左方為傳送的文字檔 in.txt，右方為接收到的文字檔 out.txt，兩者 hash 值皆相同。

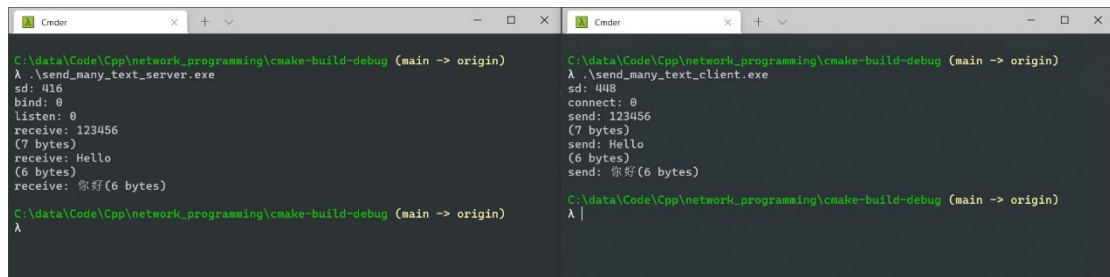
名稱	校驗值
CRC32	CF02BB5C
MD5	E99A18C428CB38D5F260853678922E03
SHA-1	6367C48DD193D56EA7B0BAAD25B19455E529F5EE
SHA-256	6CA13D52CA70C883E0F0BB101E425A89E8624DE51DB2D2

文字檔內容如下圖。



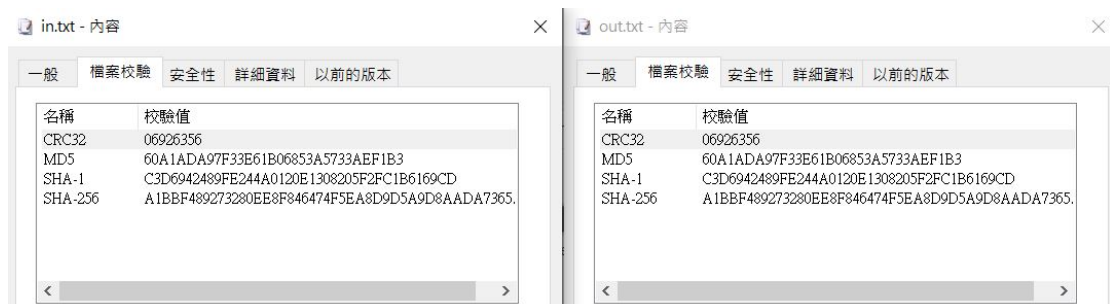
練習 5：

請參考連續讀寫範例，完成連續傳送大量文字檔的實驗。

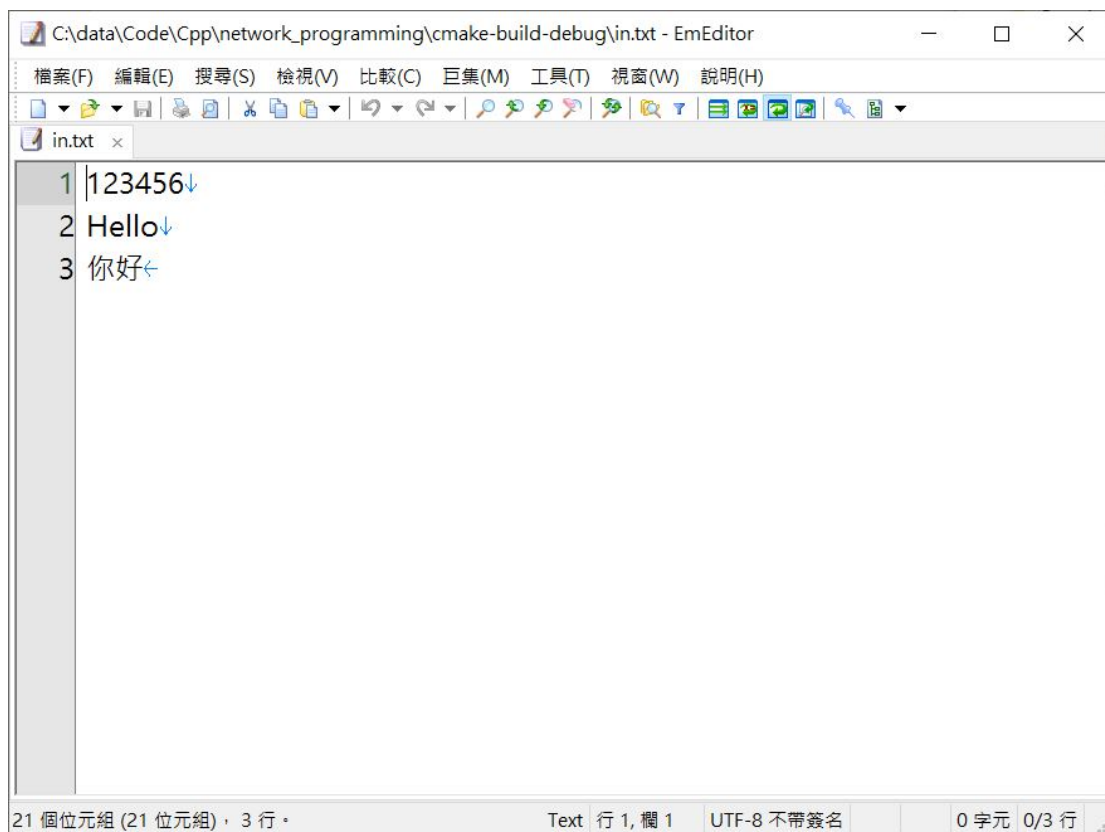


檢驗傳送的文字檔 hash 值與接收到的文字檔 hash 值相同。

左方為傳送的文字檔 in.txt，右方為接收到的文字檔 out.txt，兩者 hash 值皆相同。

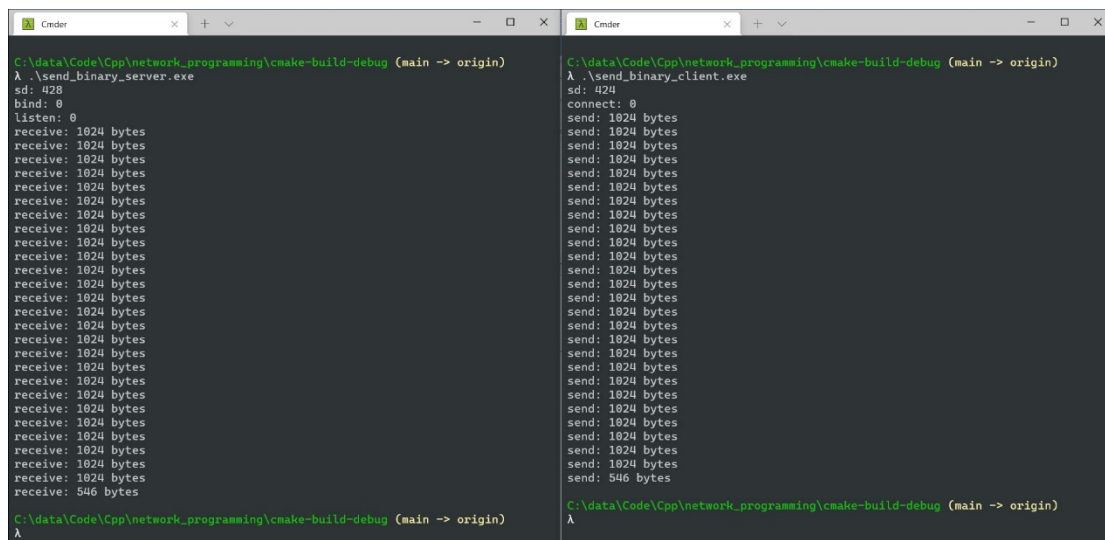


文字檔內容如下圖。



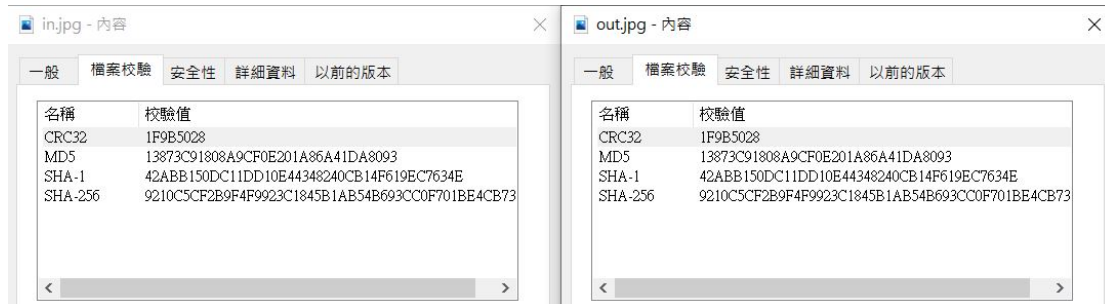
練習 6：

請參考連續讀寫二位元檔範例，完成連續傳送大量二位元檔的實驗。



檢驗傳送的文字檔 hash 值與接收到的文字檔 hash 值相同。

左方為傳送的二位元檔(圖片檔) in.jpg，右方為接收到的二位元檔(圖片檔) out.jpg，兩者 hash 值皆相同。



二位元檔(圖片檔)內容如下圖。



練習 7：

請完成 UDP 傳檔的實驗。

```

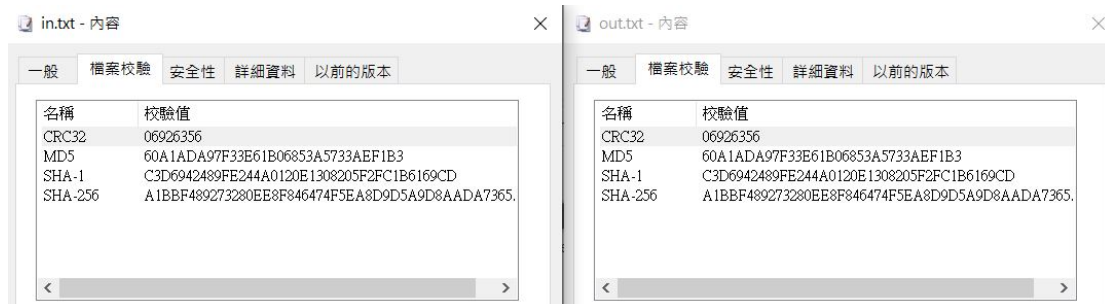
C:\data\Code\Cpp\network_programming\cmake-build-debug (main -> origin)
λ .\udp_send_many_text_server.exe
sd: 428
bind: 0
receive: 7 bytes
receive: 6 bytes
receive: 6 bytes
+C
C:\data\Code\Cpp\network_programming\cmake-build-debug (main -> origin)
λ |

C:\data\Code\Cpp\network_programming\cmake-build-debug (main -> origin)
λ .\udp_send_many_text_client.exe
sd: 444
connect: 0
send: 123456
(7 bytes)
send: Hello
(6 bytes)
send: 你好(6 bytes)
C:\data\Code\Cpp\network_programming\cmake-build-debug (main -> origin)
λ

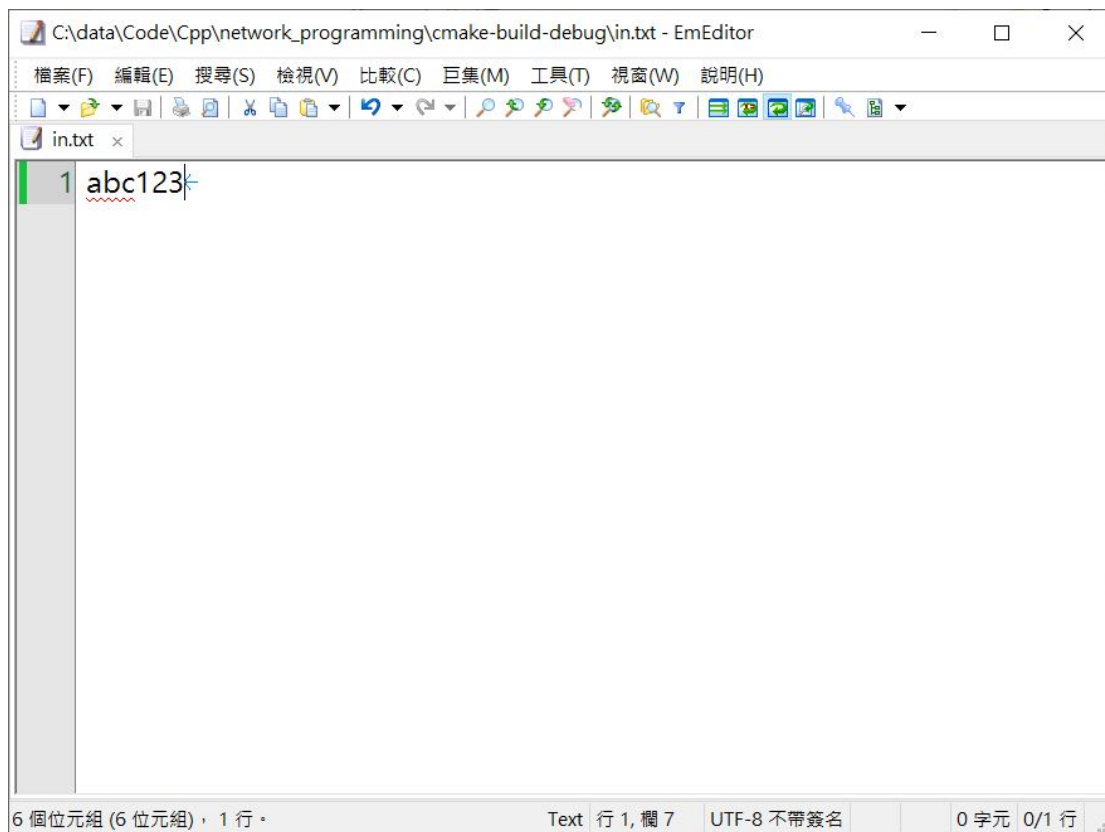
```

檢驗傳送的文字檔 hash 值與接收到的文字檔 hash 值相同。

左方為傳送的文字檔 in.txt，右方為接收到的文字檔 out.txt，兩者 hash 值皆相同。



文字檔內容如下圖。



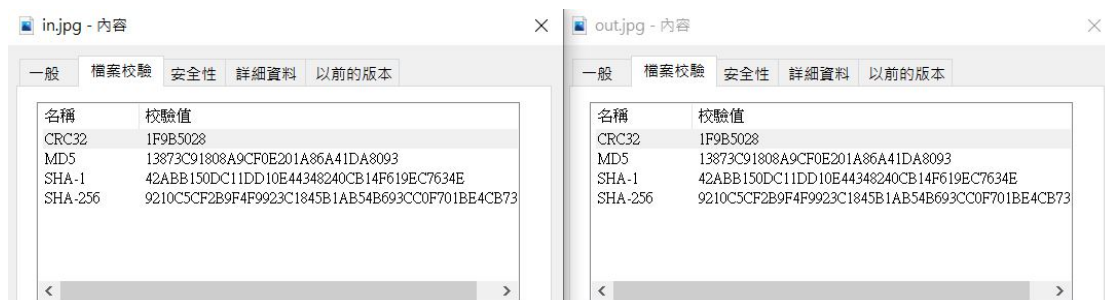
練習 8：

請完成 UDP 傳檔改良的實驗。

[illegible]

檢驗傳送的文字檔 hash 值與接收到的文字檔 hash 值相同。

左方為傳送的二位元檔(圖片檔) in.jpg，右方為接收到的二位元檔(圖片檔) out.jpg，兩者 hash 值皆相同。



二位元檔(圖片檔)內容如下圖。



三、心得

這次作業的內容非常多，老師讓全班抽籤分組，以 Pair programming 的方式來撰寫程式，這次遇到的問題就是當把 `n = recv` 函式放在 `while` 迴圈接收判斷時，會導致只回傳 0 或 1 的情況，須將其安排至迴圈外才可進行運作，除此之外，似乎沒有其他問題了。