Report CN HW3

B07902144 彭約博 資工三

Server運作方式:

使用master_sock(代表現有的已連線sockets) 和
command_sock(代表現在有指令需要執行的sockets)，
並使用select監聽所有有東西可讀的Sockets。

首先將localSocket加入master_sock以便於監聽是否有新socket連線進來，

進入While迴圈:

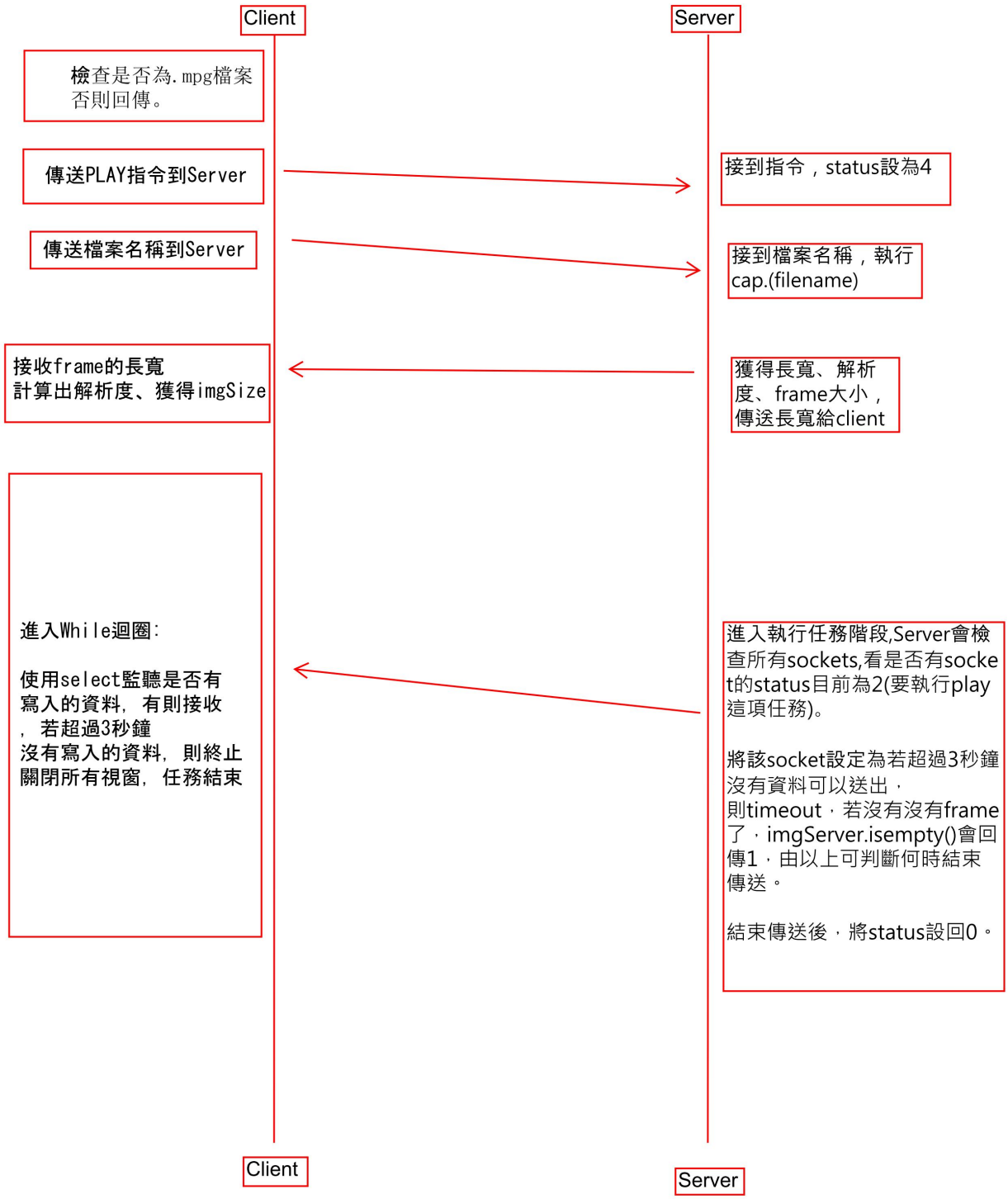
從最小的檔案描述符，檢查到最大的檔案描述符(fdmax)，看看是否有人目前有指令需要執行:

若是localSocket有指令，代表有新的連線，將accept到的socket加進master_socket。

若是其他Socket有指令，代表有client傳送想要執行的指令，檢查收到的是哪個指令，並且將status[i]設置為1~4其中之一個數字(預設為0，代表沒有指令需要做)，代表ls、play、put、get四個指令，意思為第i個socket目前為需要執行某一個指令的狀態。

再來便可檢查在master_sock中，有指令需要執行的socket(status不為0)，根據每個Socket不同的status去執行各Socket之任務，並在執行完成後將status設回0。

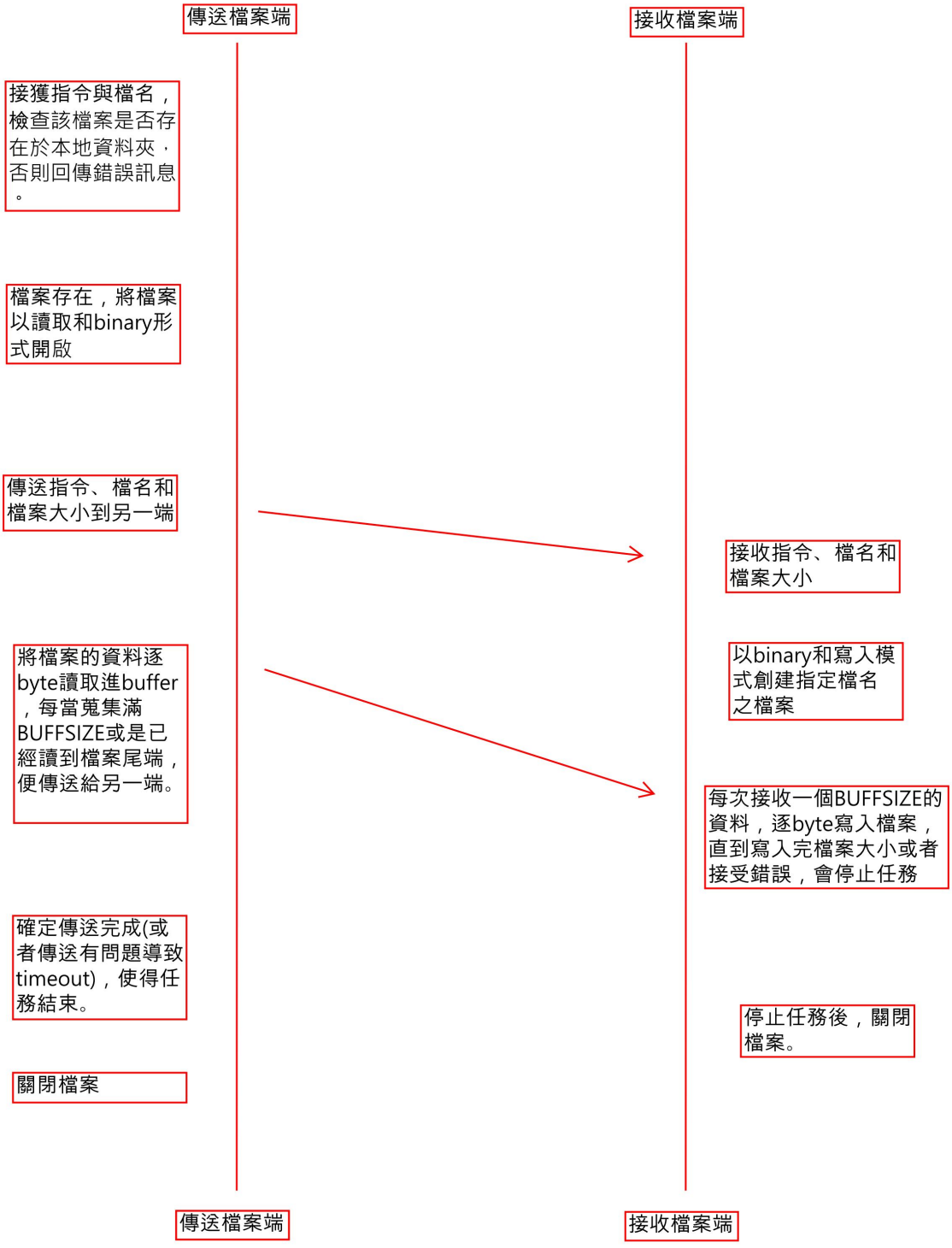
1


Client
Server
檢查是否為.mpg檔案
否則回傳。
傳送PLAY指令到Server
接到指令，status設為4
傳送檔案名稱到Server
接到檔案名稱，執行
cap.(filename)
接收frame的長寬
計算出解析度、獲得imgSize
獲得長寬、解析
度、frame大小，
傳送長寬給client
進入While迴圈:
使用select監聽是否有
寫入的資料，有則接收
，若超過3秒鐘
沒有寫入的資料，則終止
關閉所有視窗，任務結束
進入執行任務階段,Server會檢
查所有sockets,看是否有socke
t的status目前為2(要執行play
這項任務)。
將該socket設定為若超過3秒鐘
沒有資料可以送出，
則timeout，若沒有沒有frame
了，imgServer.isempty()會回
傳1，由以上可判斷何時結束
傳送。
結束傳送後，將status設回0。
Client
Server

Server端在接受到play指令、檔案名稱後，將解析度傳送給Client，將如前所述，不斷檢查目前是否有需要執行play任務的socket，並且每次傳送一個frame到對應的Client，預設了cap[100]，理論上可供100個client同時串流(但實際上應該會lag)，我沒有做影片之buffer的部分。

Client端則在檢查完檔名、獲得解析度後便進入while不斷接收frame並且播放，並在最後由於Server端已無檔案送過來，Client會在3秒後，中斷接收任務，並且關閉所有視窗。

2


傳送檔案端
接收檔案端
接獲指令與檔名，檢查該檔案是否存在於本地資料夾，否則回傳錯誤訊息。
檔案存在，將檔案以讀取和binary形式開啟
傳送指令、檔名和檔案大小到另一端
接收指令、檔名和檔案大小
將檔案的資料逐byte讀取進buffer，每當蒐集滿BUFFSIZE或是已經讀到檔案尾端，便傳送給另一端。
以binary和寫入模式創建指定檔名之檔案
每次接收一個BUFFSIZE的資料，逐byte寫入檔案，直到寫入完檔案大小或者接受錯誤，會停止任務
確定傳送完成(或者傳送有問題導致timeout)，使得任務結束。
停止任務後，關閉檔案。
關閉檔案
傳送檔案端
接收檔案端

put:

Client在收到put指令後，會檢查要put的檔案是否存在於Client的資料夾，若否則回傳錯誤訊息，若存在，則將檔案已binary和讀取mode開啟，以seekg、tellg獲得檔案之大小，並將put指令、檔案名稱、檔案大小，傳送給server。然後便開始逐個byte讀取檔案並且放到buffer(1024 bytes)中，若是已經讀了檔案大小個bytes，或者buffer已滿，就將資料傳送給Server，前者狀況傳送完資料後，會結束讀取檔案，並且關閉檔案，完成任務。

Server在收到put指令、檔名和檔案大小後，會以寫入mode在Client的資料夾創建該檔案，接下來便不斷地從Client端接收資料到buffer(1024 bytes)，並且逐個byte寫入檔案，若寫入了檔案大小個bytes，則結束上傳任務，並且關閉檔案。

get:

Client在收到get指令後，會執行ls指令，列出Server資料夾中，目前有的檔案，並且檢查我們所要求的檔案是否在其中，若否則回傳錯誤訊息，若存在，傳送get指令和檔名給Server，並以binary和寫入mode在Client的資料夾創建該檔案，接下來便不斷地從Server端接收資料到buffer(1024 bytes)，並且逐個byte寫入檔案，若寫入了檔案大小個bytes，則結束下載任務，並且關閉檔案。

Server收到ls指令後，會執行其要求，將Server端的資料夾內的檔名傳送給Client端，並且在其確認要下載之檔案存在後，獲得get指令和檔名、將檔案已binary和讀取mode開啟，以seekg、tellg獲得檔案之大小。然後便開始逐個byte讀取檔案並且放到buffer(1024 bytes)中，若是已經讀了檔案大小個bytes，或者buffer已滿，就將資料傳送給Server，前者狀況傳送完資料後，會結束讀取檔案，並且關閉檔案，完成任務。

3

當Server close一個連線時，若client端接著發資料。根據TCP協議的規定，會收到一個RST回應，Client再往這個Server傳送資料時，系統會發出一個SIGPIPE訊號給Client，告訴程序連線已經斷開了，不要再寫了。

由於我只會在Client端終止時才關閉Server端的socket，故不會發生此狀況。

4

Synchronous I/O 在主thread發出I/O請求時，要等待結果返回，但依照process的行為可以分為blocking 或者non-blocking，範例如下:

範例中，使用Synchronous I/O時，服務生說要查一下，但是我可以選擇守在電腦前等待服務生傳訊息告訴我結果，也可以選擇去做其他事情，隔一段時間再回來看有沒有訊息告訴我訂位結果。

故Blocking I/O是Synchronous I/O的一種，但Synchronous I/O不等於Blocking I/O。

Reference:

https://kaka-lin.github.io/2020/07/io_models/

https://medium.com/@clu1022/淺談i-o-model-32da09c619e6