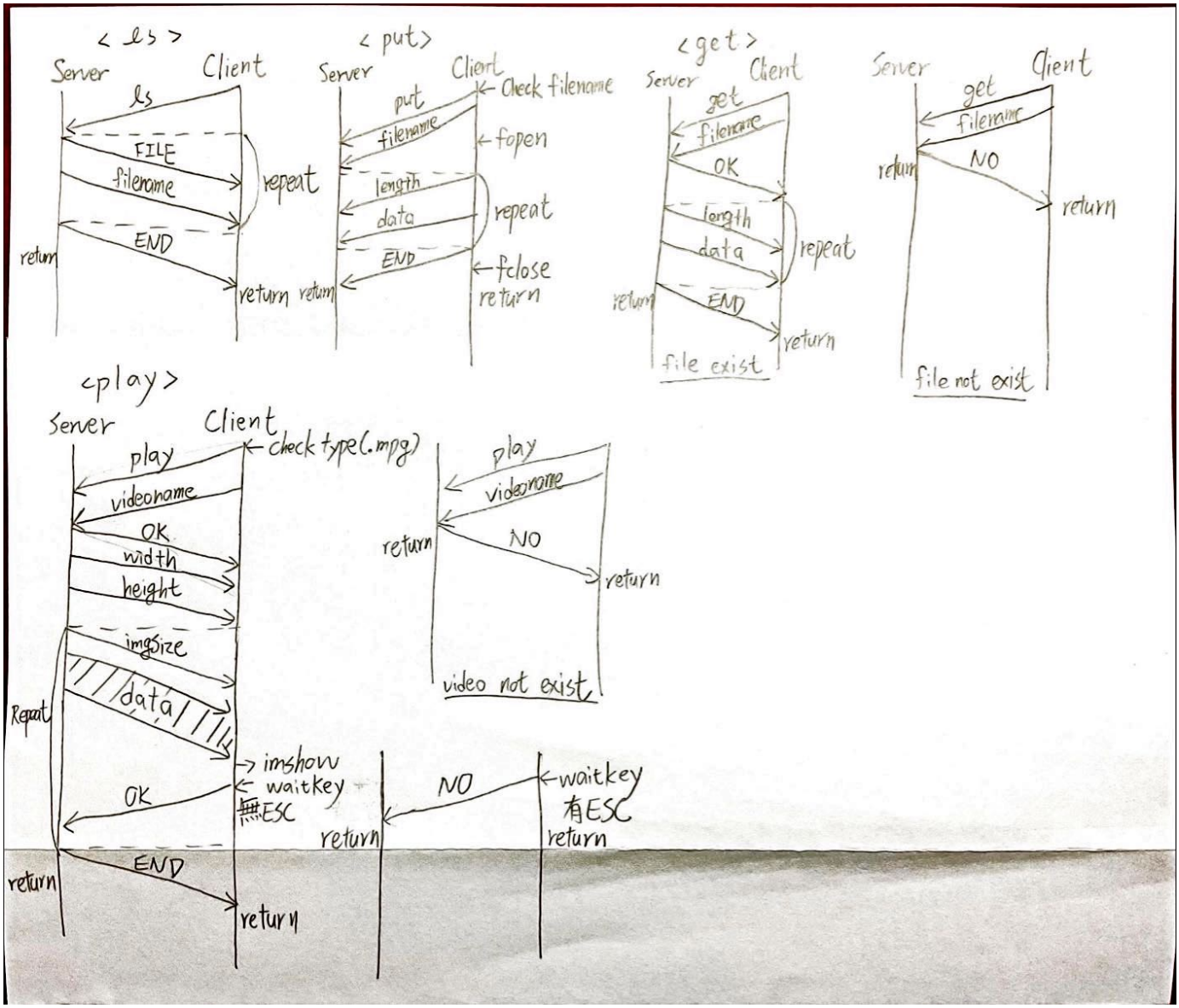
計算機網路HW2 Report

all flowchart(手繪)



file transferring(傳輸多個檔案拆分成一次傳輸一個檔，重複處理至結束)

Client簡稱C，Server簡稱S

- ls: 首先C傳送ls給S，表示要進行ls操作。接著S每次傳送標籤FILE以及實際上的filename這兩個buffer給C(不包含.以及..)，C則將recv到的filename輸出，最後S讀完所有file則傳送標籤END給C表示結束並return，C接收到也是return等待下個指令。
- put(每次put單一個檔案): 首先C先確定本地有filename此檔案，若沒有則輸出不存在；有的話則傳送put給S，表示要進行put操作。C傳送filename給S，S則開啟此檔案以寫入。接著C每次從file讀取一個BUFF_SIZE的data，傳送讀到的length以及實際上的data這兩個buffer給S，S則根據此length，將收到的data buffer中的length內的資料寫入file。C讀完檔案後傳送END給S，S接受到END則結束寫入並關閉file。兩者都return等待下個指令。

- get(每次get單一個檔案): 首先C傳送get給S，表示要進行get操作。接著C傳送filename給S，S則確認是否存在，若否回傳NO，有則回傳OK。C若接收到NO，則輸出此檔案不存在並return；若收到OK，則開啟此檔案並接下去以下操作: S每次從file讀取一個BUFF_SIZE的data，傳送讀到的length以及實際上的data這兩個buffer給C，C則根據此length，將收到的data buffer中的length內的資料寫入file。S讀完檔案後傳送END給C，C接受到END則結束寫入並關閉file。兩者都return等待下個指令。

video streaming

Client簡稱C，Server簡稱S

- play: 首先C先確認檔名的類型是否為.mpg，否則輸出錯誤訊息並return。若正確則傳送play給S，表示要進行play操作。接著C傳送videoname給S，S確認是否存在，是則回傳OK，否回傳NO。C接收到NO則輸出檔案不存在並return，若收到OK則接下去正式streaming: S先傳送image的長寬給C，兩者都初始化好Mat後，S每次取得一個frame，先傳送此image的大小imgSize給C，接著迴圈傳送總共此imgSize大小的image data給C，而C在收到imgSize後，也是迴圈讀取此imgSize大小的image data，完整讀取後使用imshow播放image，接著以waitKey查看是否有按下esc，若有C傳送NO給S表示影片不播放了，S接收到NO也會因此結束播放並return。若是沒等到esc，C傳送OK給S，S則繼續下一輪傳送。最後若是S播放完影片，S會傳送END給C，C接收到後也會因此結束播放，最後兩者都return等待下個指令。

SIGPIPE

SIGPIPE是OS傳給process的一種信號。當process傳送訊息至已關閉的socket第二次，也就是說socket的另一端接收者已經關閉了socket，此時就會產生SIGPIPE，default action是OS會把process關閉。而在我的process中會有機會收到SIGPIPE，例如在streaming的過程，client若意外的斷線或關閉，導致socket另一端關閉，server process就會收到SIGPIPE，而我的作法是catch SIGPIPE並直接return，讓server繼續運作。而若server recv得到0或是負值，則判定client斷線，返回不繼續執行當下指令。

Blocking I/O vs Synchronized I/O

They are not totally same.
事實上$Blocking\ I/O \subset Synchronized\ I/O$
For example:
Non-blocking I/O與Blocking I/O都屬於synchronized I/O，因為根據synchronized I/O的定義，呼叫I/O的process若被blocked直到I/O完成，那麼便屬於synchronized I/O。而反面的asynchronized I/O是呼叫I/O的process不會被blocked並馬上返回繼續執行，待kernel處理好data後才會以信號通知呼叫者取用。
Non-blocking I/O雖然在要求的資料還沒準備好時會馬上返回，但在要求的資料可取用時，便會被blocked住去獲取data並再完成後才返回，因此也算是synchronized I/O。
區分synchronized I/O與asynchronized I/O在於是否是主執行緒親自去進行I/O。