B09902063 資工二 董瑋

Explains how the server deal with multiple users and keep the file transfer working in detail

我利用兩個陣列紀錄現在有哪些user的connect fd正在被讀/寫

而每個connect fd後都有一個struct request紀錄這個連線的相關資料

包含: connect fd, 正在讀取的file\_fd, 正在寫入的file\_fd, user name, 傳輸中的檔案大小以及當前request的狀態 (這個連線正在做甚麼)

接著開啟兩個fd\_set從上述的陣列中抓出連線的fd並丟去select，再按照當前fd有沒有被select到來決定要不要對這個fd做事。如此就能利用I/O multiplexing 達到讓多人同時傳輸檔案的目的，而每次被select到時只會傳輸1kB，因此所耗時間不長不會讓有的user等太久。

What is SIGPIPE? It is possible to happen to your code? If so, how do you handle it?

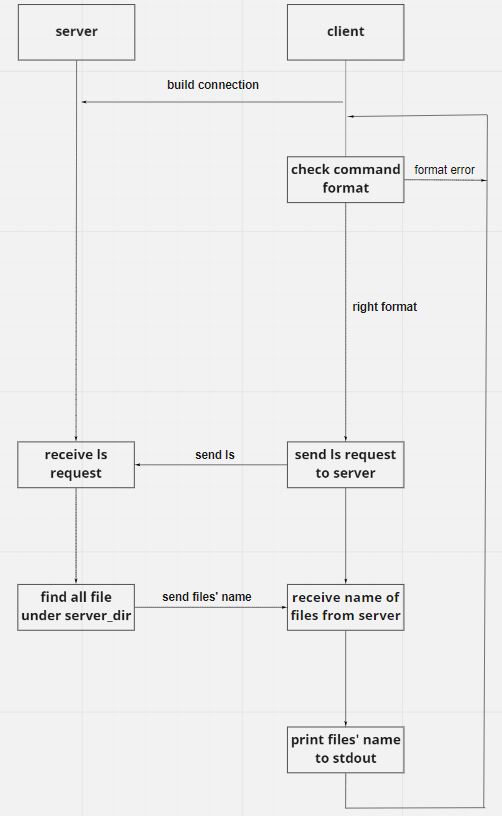
SIGPIPE是”broken pipe”的signal，當有程式要對一個沒有人在讀取的pipe或socket寫入時，就會產生SIGPIPE給這支程式，預設動作是terminate 因此程式就會被結束掉

我的server有可能會產生SIGPIPE，因此我把它預設的動作改成ignore掉他。接著當server讀取時若讀到EOF則代表client已經關掉連線或是send/write時回傳-1也可能是連線被關閉，因此我設定成只要讀寫socket時回傳值<=0就視為client關閉連線，server就把他的username從紀錄裡丟掉並重設request狀態

Connection

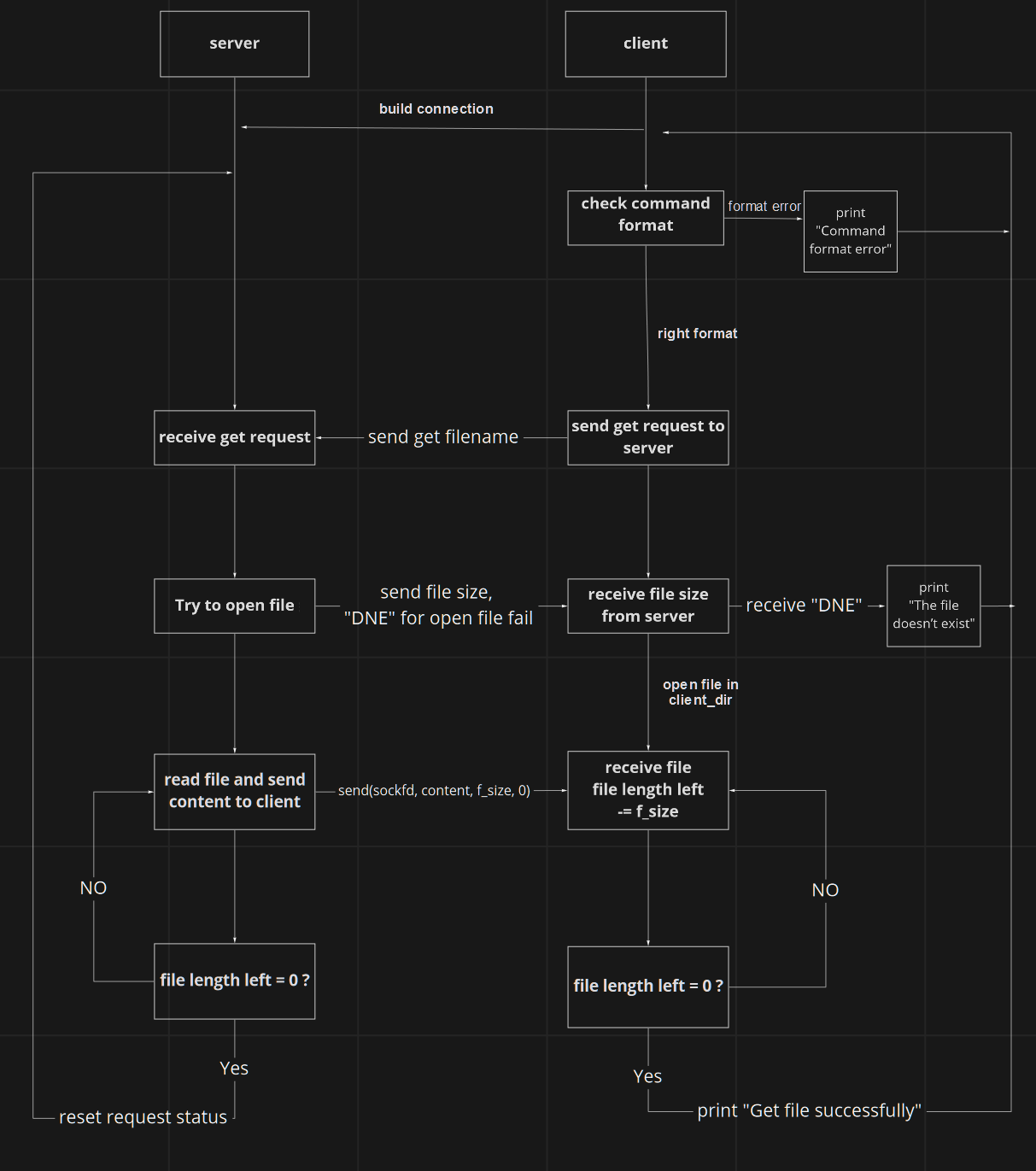


ls



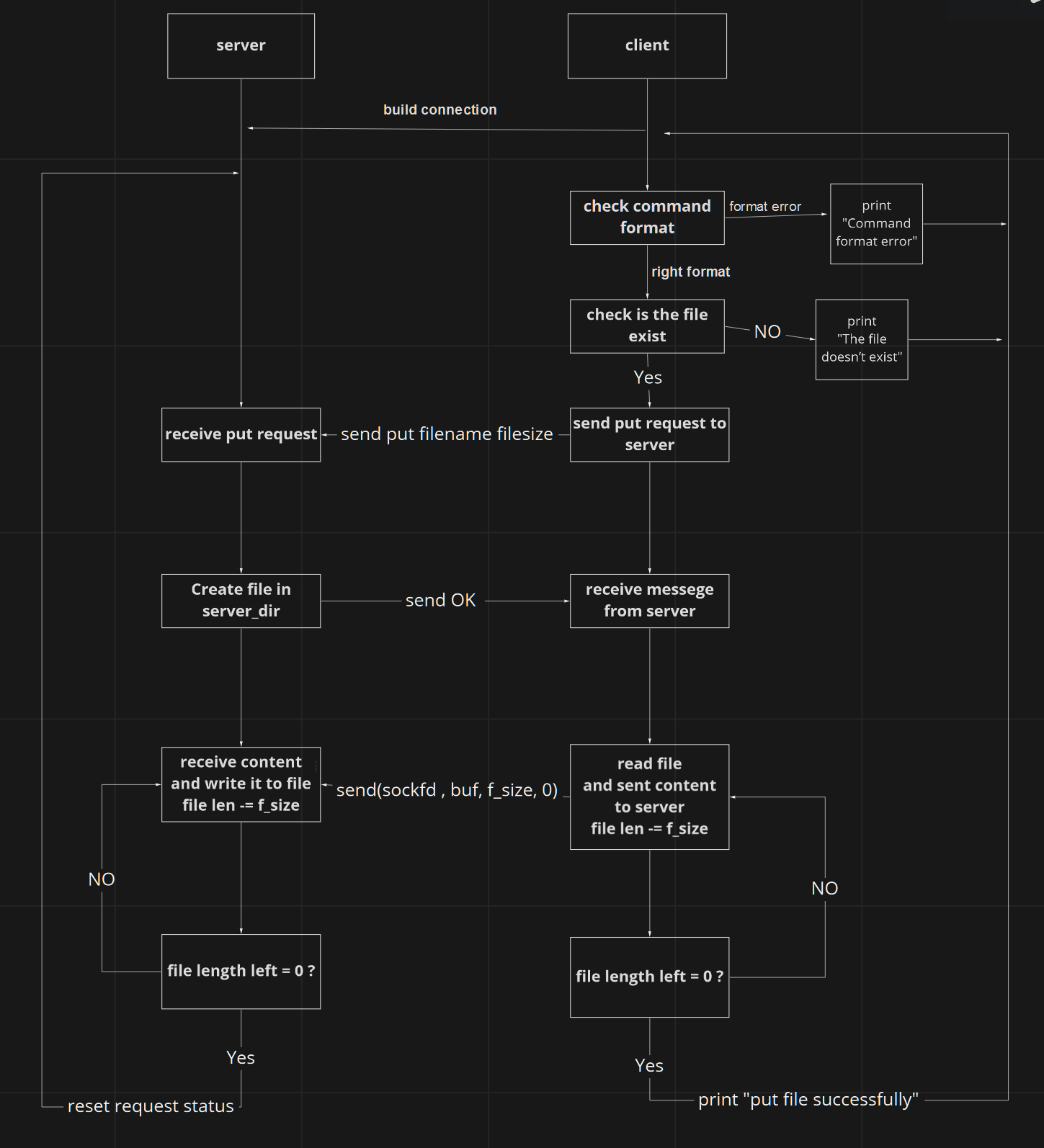
1. 先和server建立連線，接著讀取到ls指令時傳送給server
2. Server讀到ls指令後開始用scandir遍歷server\_dir找出檔名，然後加上排列方式保證會是按字典序排列
3. 最後回傳結果給client輸出到stdout

get



1. 同樣先建立連線，接著確認command format是否正確
2. 傳送get指令給server，還要把filename傳給server確認檔案是否存在
3. Server\_dir中若檔案不存在則回傳DNE，否則回傳檔案大小
4. 如果client收到DNE則print “The filename doesn’t exist”，否則紀錄檔案大小然後一直傳輸並更新剩餘多少還沒收到直到接收到全部檔案
5. Server端則會記錄client的connect fd和剩下多少還沒傳，每次select到這格connect fd時傳輸min(1kB, filesize)並更新剩下多少還沒傳。傳完以後便初始化request狀態，繼續等待client下個指令

Put



1. 同樣先建立連線，接著確認command format是否正確
2. 接著檢查client\_dir是否有filename存在，若沒有要

print“The filename doesn’t exist”

否則把put指令、filename和檔案大小傳給server

1. Server收到要求後在server\_dir下建立檔案，然後更新request狀態並記錄filesize準備開始寫檔
2. Client 檢查檔案剩多少沒傳，每次傳1KB並更新filesize

當filesize==0時回到之前狀態重新等user輸入指令

1. Server則紀錄這個connect fd正在寫檔，每次select到他時接收1KB並寫進server\_dir底下的檔案，然後更新filesize。當filesize==0時初始化request狀態並等待下個指令