

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)

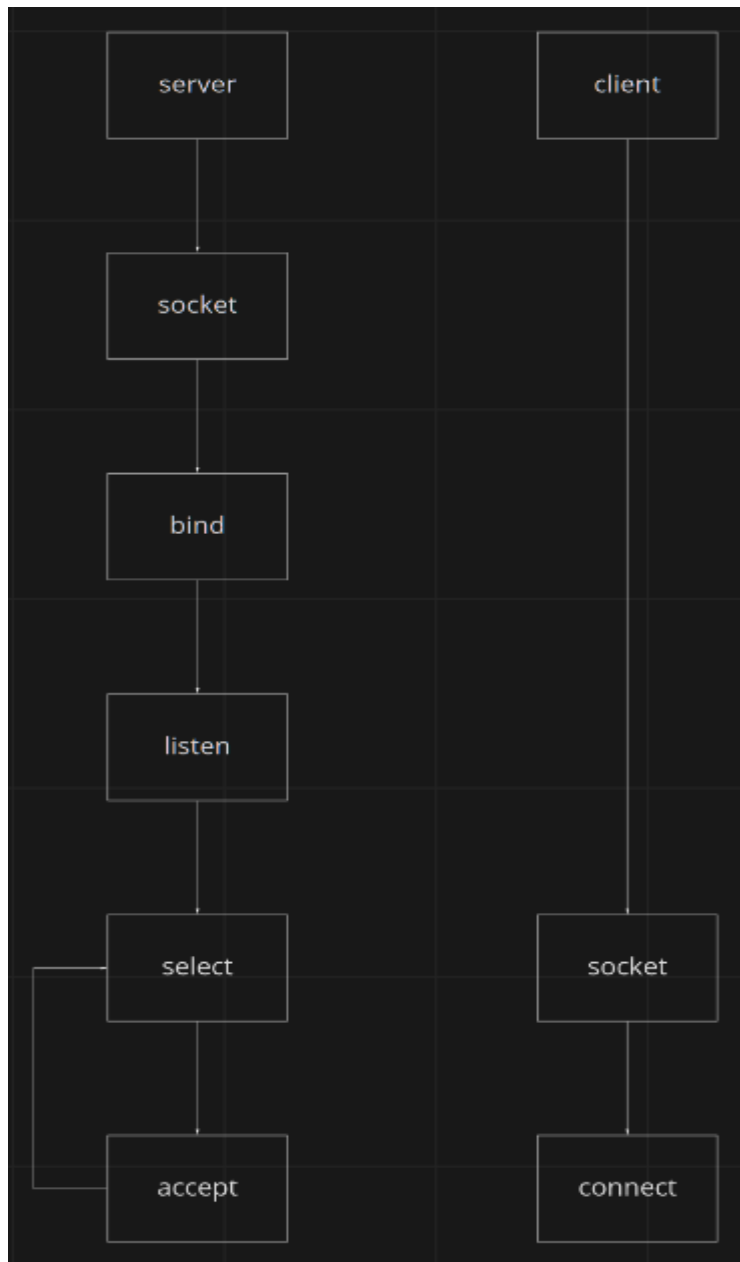
接著開啟兩個 fd_set 從上述的陣列中抓出連線的 fd 並丟去 select，再按照當前 fd 有沒有被 select 到來決定要不要對這個 fd 做事。如此就能利用 I/O multiplexing 達到讓多人同時傳輸檔案的目的，而每次被 select 到時只會傳輸 1kB，因此所耗時間不長不會讓有的 user 等太久。

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

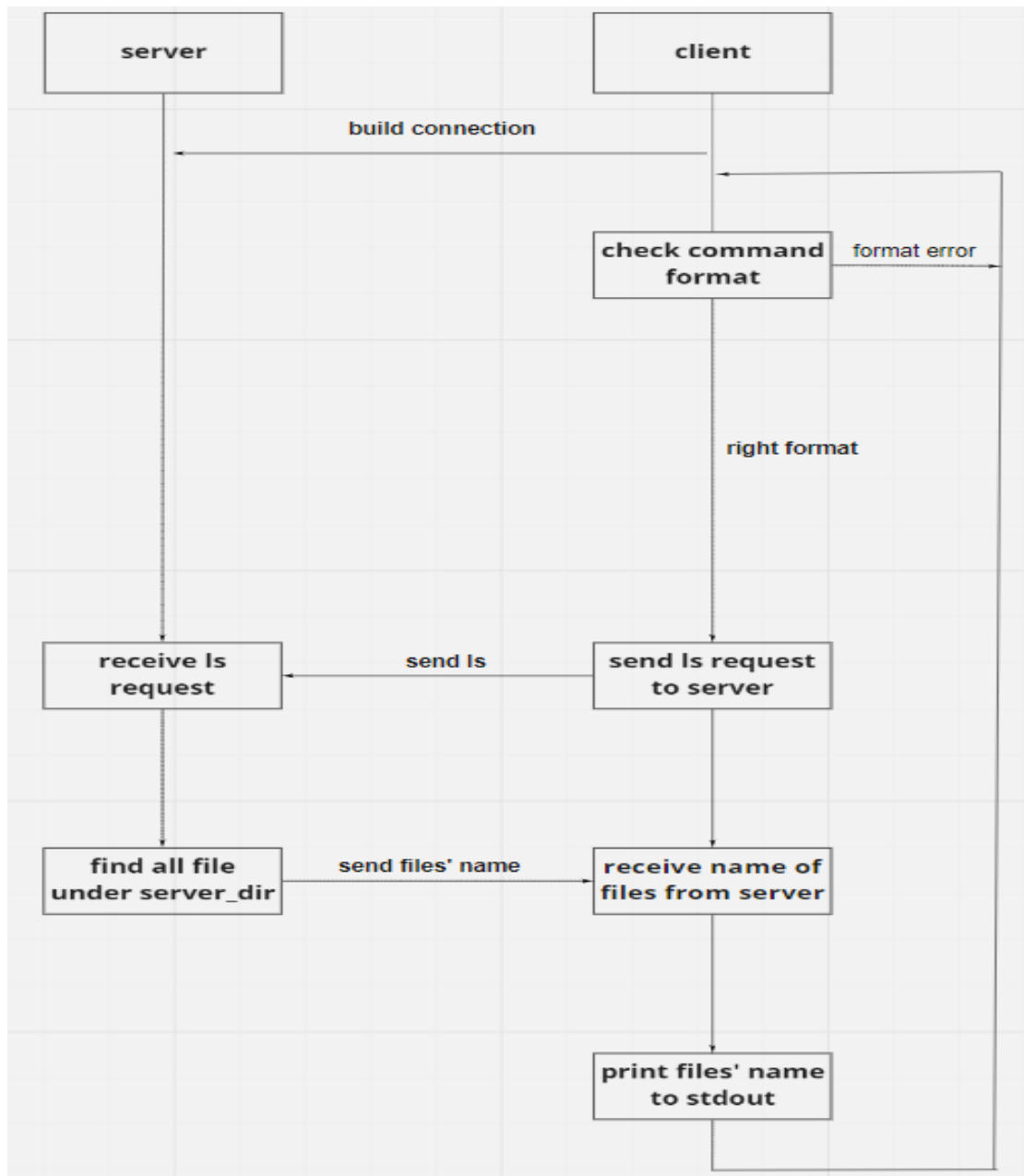
SIGPIPE 是”broken pipe”的 signal，當有程式要對一個沒有人在讀取的 pipe 或 socket 寫入時，就會產生 SIGPIPE 給這支程式，預設動作是 terminate 因此程式就會被結束掉

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

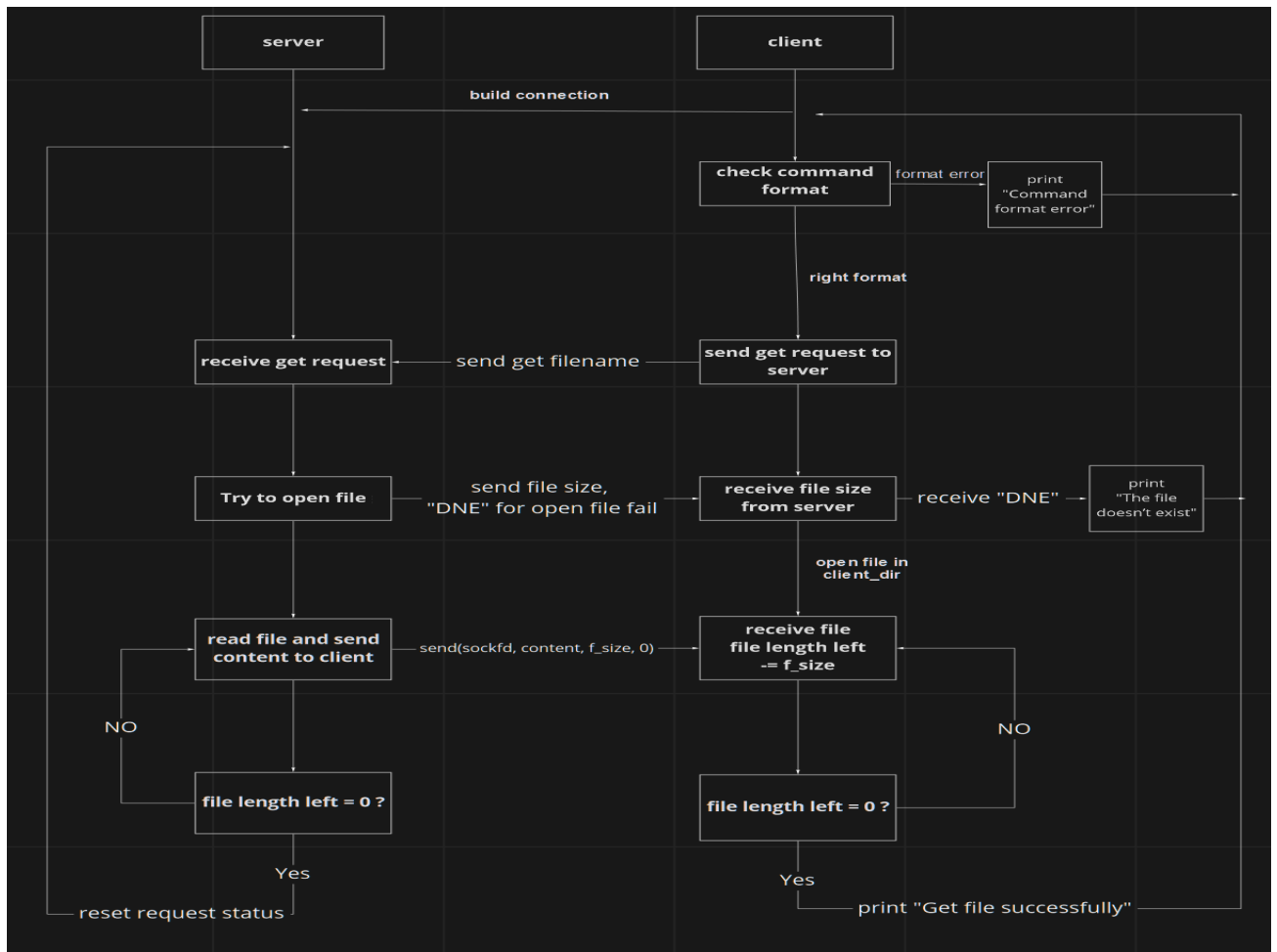


ls



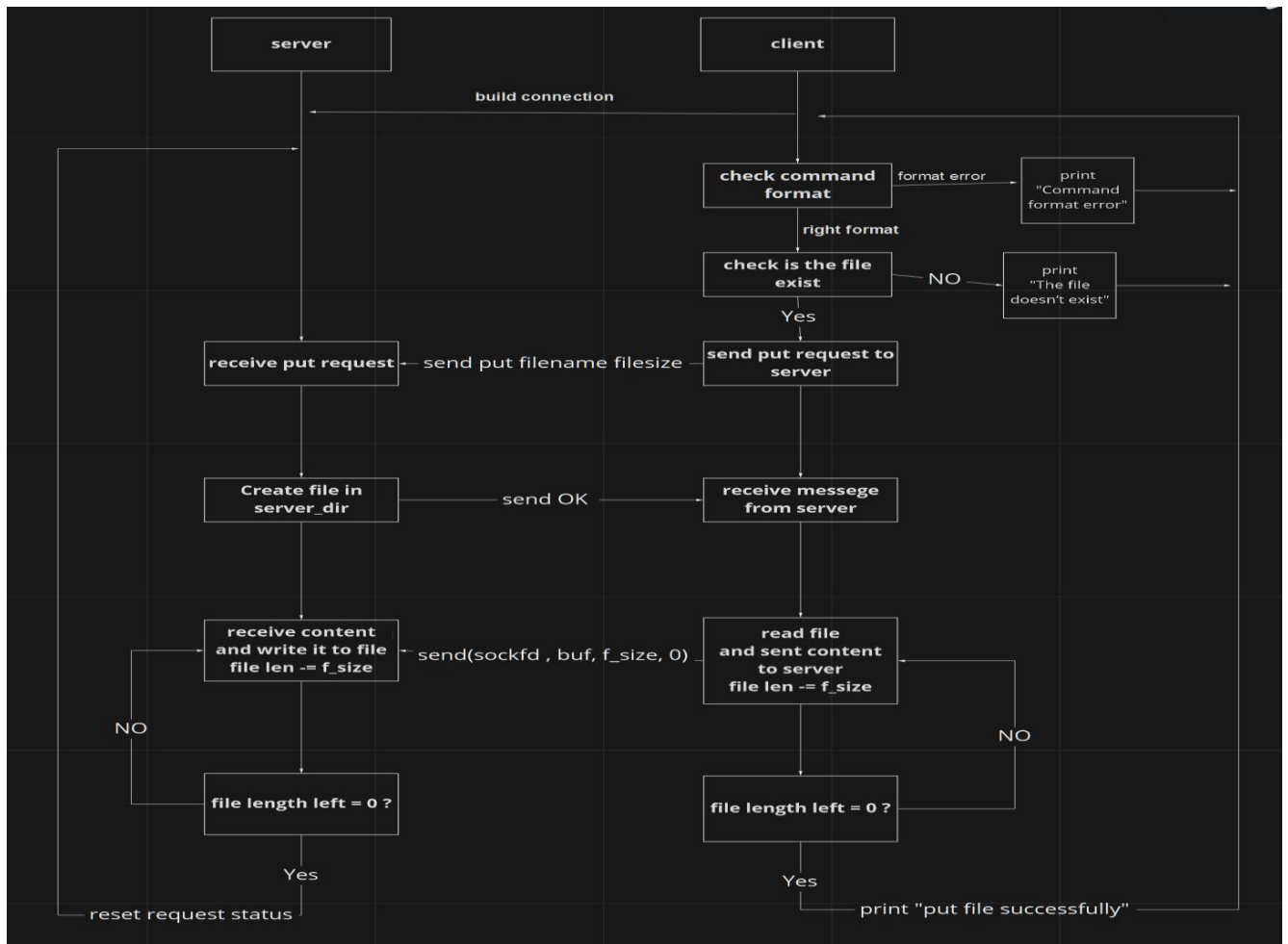
1. 先和 server 建立連線，接著讀取到 ls 指令時傳送給 server
2. Server 讀到 ls 指令後開始用 scandir 遍歷 server_dir 找出檔名，然後加上排列方式保證會是按字典序排列
3. 最後回傳結果給 client 輸出到 stdout

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
3. 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
4. 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
5. 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