

## SP HW 1 report

B07902094 廖榮運

1. 用 select 實現, 創建一個 master\_set 還有 read\_set, 每一次迴圈都會先複製 master\_set 到 read\_set 再把 read\_set 丟去 select, select 回傳 read\_set 後會把 read\_set 掃過去, 發現哪些 FD\_ISSET 就往下進行.
  - 如果 fd==svr.listen\_fd 就代表是新連線, 將 master\_set 設置成 on 之後就 continue,
  - 否則就直接往下做, 用 handle read 把 socket 的資料讀出, 所有事情處理完或是將 client 的連線關掉時, 就會把該 master\_set 的 fd 位置關掉。
2. 並沒有 busy waiting, 每次 while loop 只是跑過 read\_set 一次, 並不會一直做無謂的等待和詢問, 讀取完 read\_set 後, 直到下次 while loop 前都不會等待或詢問, 所以能避免 busy waiting。
3. 同個 process 的 lock, 我是在 process 內建立 write\_lock array 和 read\_lock array 來得知哪個 id 有被上鎖, 以及被上的鎖屬於 read\_lock 還是 write\_lock。若要執行 read 的時候要 check write\_lock[id] 是不是 true, 執行 write 的時候就要 check read\_lock[id] 和 write\_lock[id], 有任何一個被上鎖就要反回 locked 並關閉連線。並且讀寫完都要把鎖的 array 設回 false。
4. 在不同的 process 我就是利用 fcntl, 每次 write 或 read 前都呼叫 fcntl 檢查檔案的該區域有沒有被 lock, 回傳-1 就回傳 locked 並把連線關掉, 否則就進行 read 或 write, 完成 read 或 write 之後, 再利用 fcntl 進行 unlock。