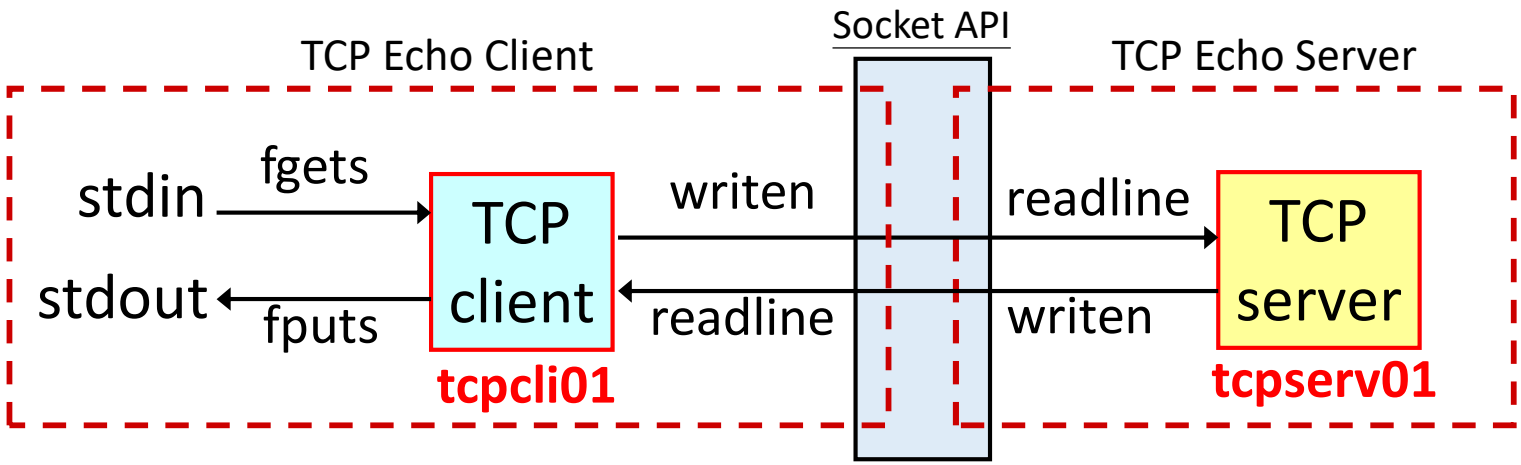
網路程式設計

動手操作 2

TCP State 的觀察實驗

TCP Echo Server and Client (Chap. 5)



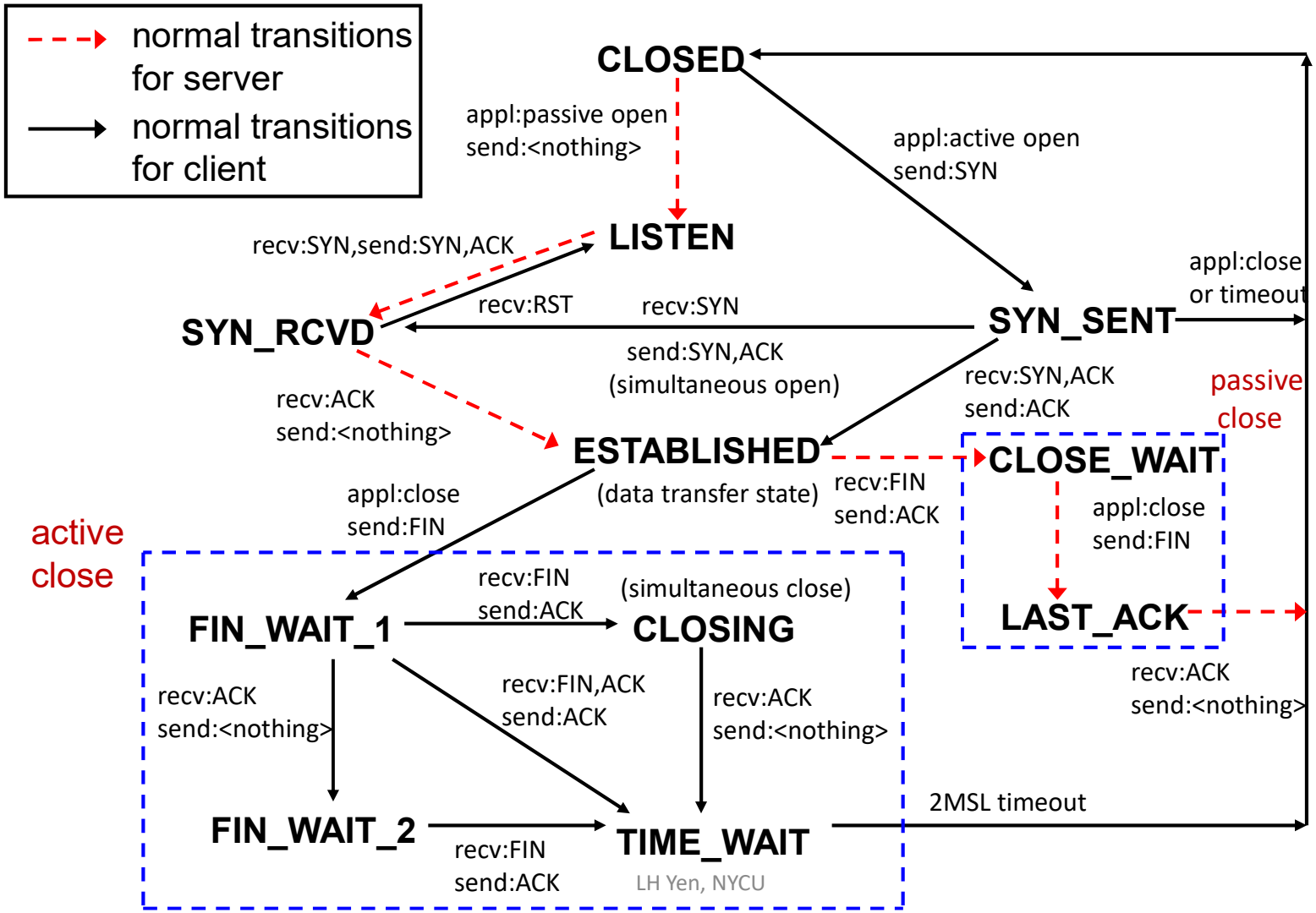
TCP State (1/4): Without Server

- 先開一終端機鍵入 **watch 'netstat | grep 9877'** 按下 

- 再開一終端機進入~/unpv13e/tcpcliserv
- 如未執行過 make 先執行 make (忽略錯誤訊息)
- 執行echo client: **./tcpcli01 140.113.1.1** 

- 藉由第一個終端機 觀察此client 的 TCP state 為何?
  Ans:
- 參照下一頁的 TCP state 變換圖，說明是否合理?
- client程式最終傳回何訊息?

TCP State Transition Diagram



TCP State (2/4): Server State

- 在先前觀察用的終端機按下Ctrl+C. 再執行 **watch 'netstat –at'** 

- 使用之前執行client的終端機，進入~/unpv13e/tcpcliserv
- 確定client未在執行
- 背景執行echo server (port 9877): **./tcpserv01 &** 

- 在觀察用的終端機觀察server 的 TCP state 為何?
  Ans:
- 參照前一頁的 TCP state 變換圖說明，此結果是否合理?

TCP State (3/4): Active

- 在觀察用的終端機按下Ctrl+C. 再執行**watch 'netstat | grep 9877'** 

- 在另一終端機執行echo client: **./tcpcli01 127.0.0.1** 

- 在觀察用的終端機，指出何者為server，何者為client，目前的TCP state為何?
  Ans:

TCP State (4/4): Closing

- 承前。回到執行client的終端機，隨意鍵入文字按 

，會有甚麼輸出?
- 按Ctrl+D 結束 client 程式執行
- 在觀察用的終端機觀察此時client的TCP state為何?
  Ans:

- (持續執行**netstat**直到有變化) 此 state 何時會改變? Ans:

Process State 的觀察實驗

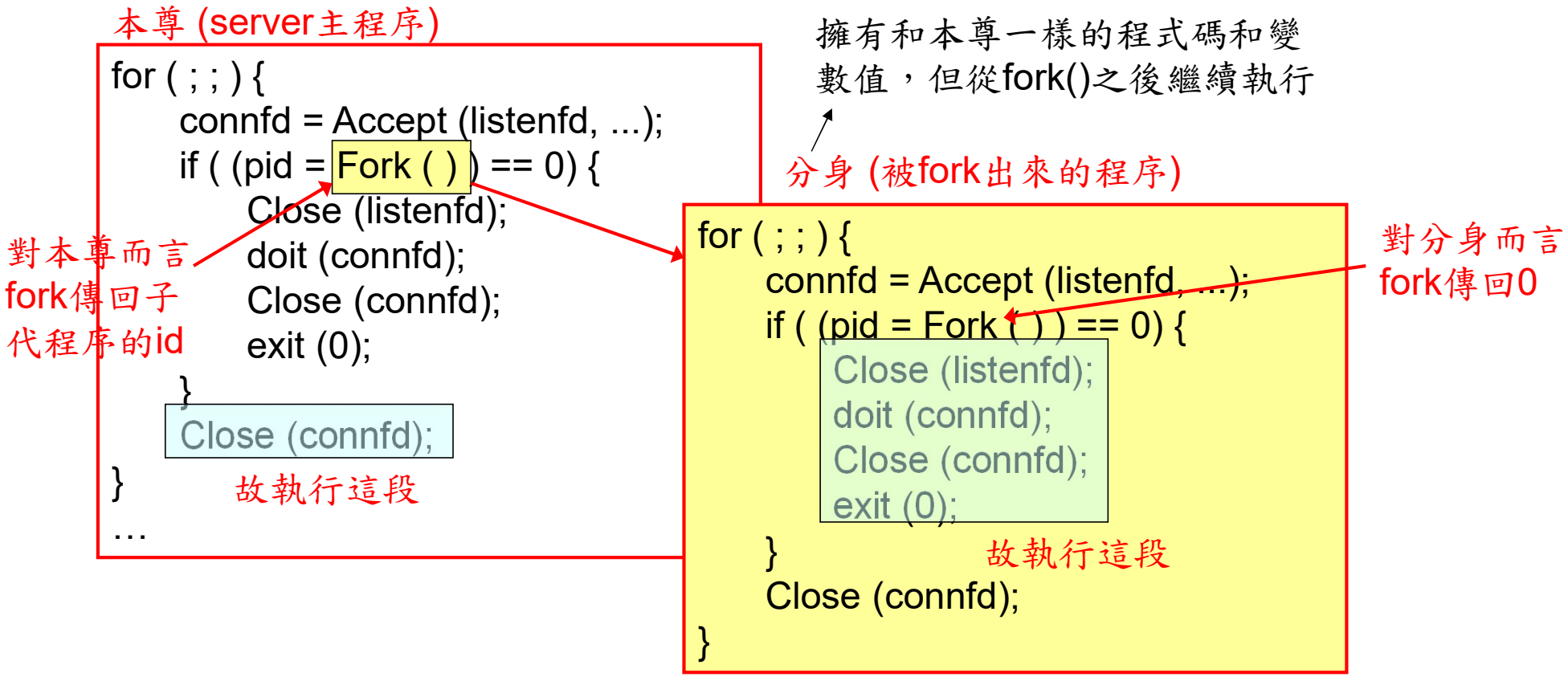
Process State

- 使用任一終端機進入~/unpv13e/tcpcliserv
- 觀看tcpserv01.c，哪一行碼建立child process? (呼叫 fork())
- (如尚未執行則)背景執行echo server: **./tcpserv01 &** 

- 執行 **ps –l** 

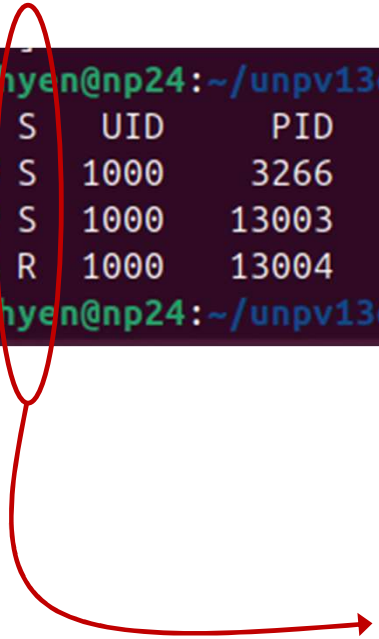
觀察server process資訊 (PID, state)

fork 的本尊與分身 (Chap. 4)



Process State Codes

```
lhyen@np24:~/unpv13e/unpv13e/tcpcliserv$ ps -l
F S   UID     PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S  1000    3266    3246  0  80   0 -  2231 do_wai pts/0    00:00:00 bash
0 S  1000   13003    3266  0  80   0 -   639 inet_c pts/0    00:00:00 tcpserv01
0 R  1000   13004    3266 99  80   0 -  2762 -      pts/0    00:00:00 ps
lhyen@np24:~/unpv13e/unpv13e/tcpcliserv$
```



- D Uninterruptible sleep (usually IO)
- R Running or runnable (on run queue)
- S Interruptible sleep (waiting for an event to complete)
- T Stopped, either by a job control signal or because it is being traced.
- W paging (not valid since the 2.6.xx kernel)
- X dead (should never be seen)
- Z Defunct ("zombie") process, terminated but not reaped by its parent.

Child Process

- 續前。執行echo client: **./tcpcli01 127.0.0.1** 

- 在另一終端機執行 **ps -ejf | grep tcpserv01** 

。你是否能指認何者為 child process? PID 為何?

- 輸入訊息以測試client程式執行是否正確

Child Process Termination

- 續前。在client程式執行的終端機按 Ctrl+D 結束 client 程式執行
- client 程式結束後，在client程式執行的終端機執行 **ps –l** 

，觀察child process 的狀態為何? (參照p.12的code)

D	Uninterruptible sleep (usually IO)
R	Running or runnable (on run queue)
S	Interruptible sleep (waiting for an event to complete)
T	Stopped, either by a job control signal or because it is being traced.
W	paging (not valid since the 2.6.xx kernel)
X	dead (should never be seen)
Z	Defunct ("zombie") process, terminated but not reaped by its parent.

Appendix – Why Cannot Use netstat?

- 如果安裝的作業系統沒有辦法使用netstat的指令
  - 請執行 sudo apt install net-tools 指令安裝功能

Appendix – Why Cannot Build the Project?

- 關於在linux作業系統(UBUNTU)運行unpv13e可能有錯誤需要排除
- 關於編譯libfree時出現error: argument 'size' doesn't match prototype
  - 請參考 https://people.cs.nycu.edu.tw/~lhyen/np/unpv13e_install.html
- 關於編譯tcpcliserv/cpservpoll01時可能出現error: ‘OPEN_MAX’ undeclared (first use in this function)
  - 這是因為linux系統定義的系統MACRO變數不一樣所致
  - 可以在 unp.h 裡面新加入一行自定義的MACRO變數
    - #define OPEN_MAX 1024