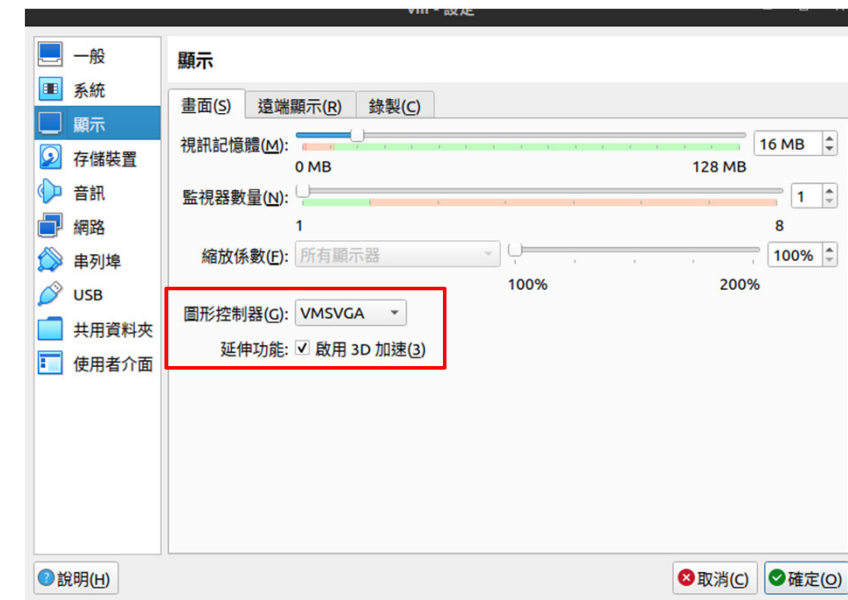
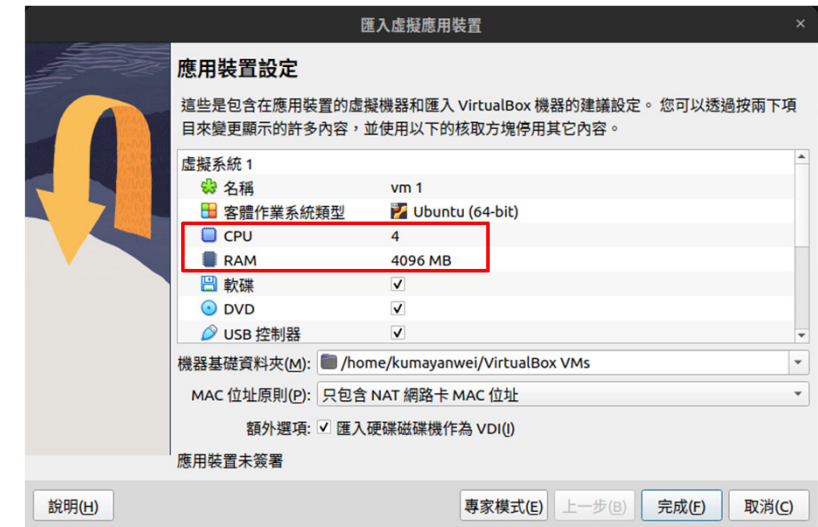


網路程式設計

動手操作 2

Setting

- Virtualbox setting:
 - 匯入(D:ubuntu-np) (圖1)
 1. Core改為 2core Memory改為2048MB
 - 匯入後: 顯示-畫面-圖形控制器 (圖2)
 1. 改為 VMSVGA
 2. 選 啟用3D加速
- Ubuntu密碼: nycunp2023



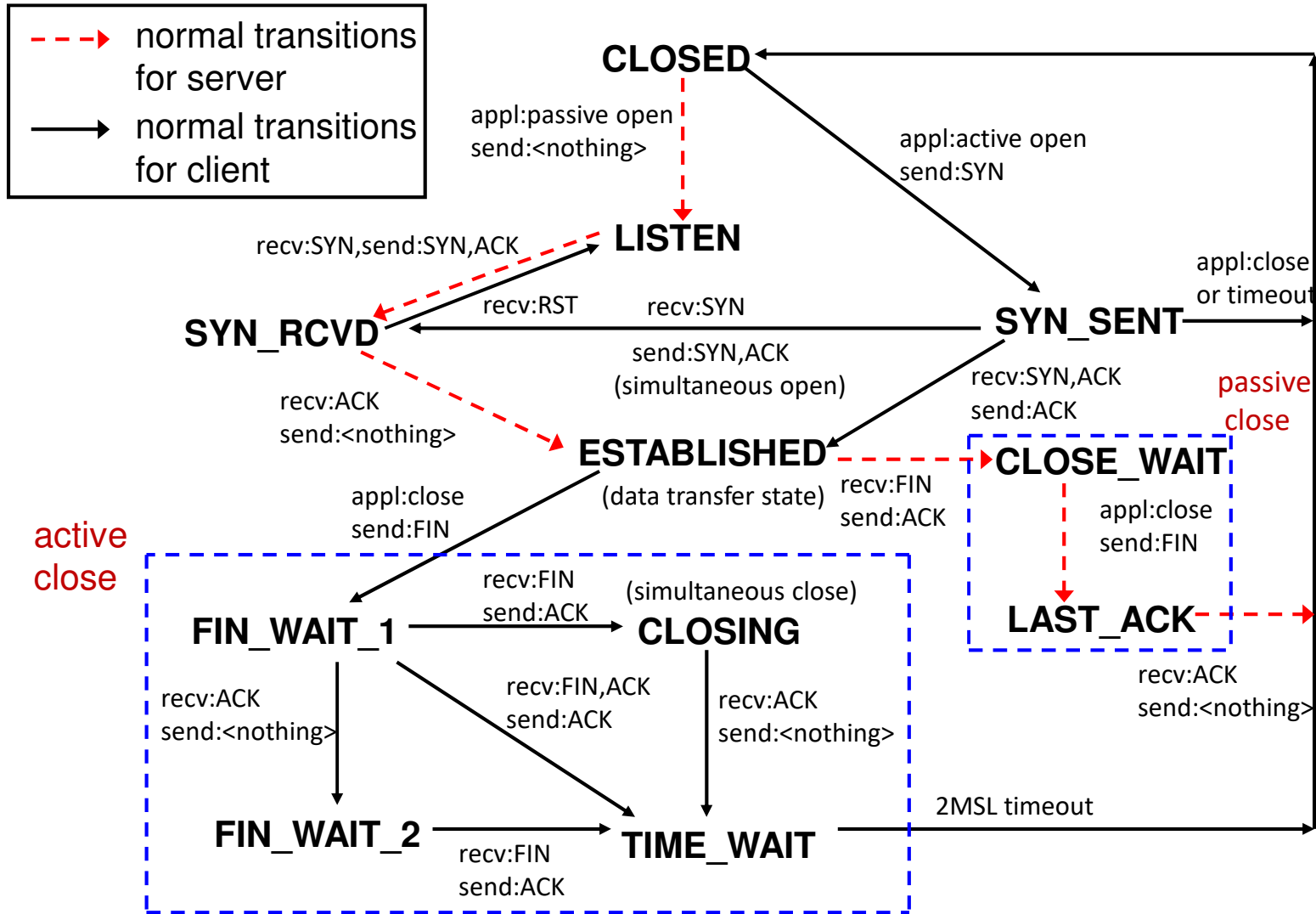
TCP State (1/3)

- 使用任一終端機進入unpv13e/tcpcliserv
- 如未執行過 make 先執行 make
- 執行echo client: `./tcpcli01 140.113.1.1`
- 另開一終端機執行 `netstat -l | grep 9877`，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?
- 按 Ctrl+D 結束 client 程式執行

TCP State Transition Diagram



TCP State (2/3)

- 使用之前執行client的終端機，進入unpv13e/tcpcliserv
- 背景執行echo server: **./tcpserv01 &**
- 執行 **netstat -l | grep 9877**，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 (3/3)

- 承前。在同一終端機執行echo client: `./tcpcli01 127.0.0.1`
- 保持echo client在執行狀態，在另一終端機執行 `netstat | grep 9877`，指出何者為server，何者為client，目前的TCP state為何? Ans:
- 按 Ctrl+D 結束 client 程式執行
- 執行 `netstat | grep 9877`，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:

Child Process

- 使用任一終端機進入unpv13e/tcpcliserv
- 觀看tcpser01.c，哪一行碼建立child process? (呼叫 `fork()`)
- (如尚未執行則)背景執行echo server: `./tcpser01 &`
- 執行echo client: `./tcpcli01 127.0.0.1`
- 在另一終端機執行 `ps -ejf | grep tcpser01`。你是否能指認何者為 child process?
- 輸入訊息以測試client程式執行是否正確
- 按 `Ctrl+D` 結束 client 程式執行

觀察Zombie

- 承前。client 程式結束後執行
ps -ejf | grep tcpserver01
你是否能指認出 zombie process?
- 使用 **kill** 指令強迫中止 server 程式。再執行
ps -ejf | grep tcpserver01
zombie process 是否仍然存在?

Zombie的處理

- 使用任一終端機進入unpv13e/tcpcliserv
- 觀看tcpser04.c，哪一行碼在註冊signal handler? 哪一行碼在處理accept 呼叫被中斷時所產生的錯誤?
- 觀看sigchldwait.c，哪一行碼處理zombie?
- 執行 **./tcpser04 &** 並重覆前次的client程式執行測試。
- 測試結束後執行 **ps -ejf | grep tcpser04** 查看是否還有 zombie process?

Half-Closed Connection (1/2)

- 打開終端機程式，進入unpv13e/tcpcliserv
- 執行 **./tcpserver04**
- 執行echo client: **./tcpcli01 127.0.0.1**
- 另開一終端機程式執行 **ps -ejf | grep tcpserver04** 找出 **child** process id 並將其 kill 掉
- 執行 **netstat | grep 9877**。指認 server parent 與 client 的 TCP states 各為何?

Half-Closed Connection (2/2)

- 承前。在client程式輸入訊息並按enter(或讓client程式閒置一段時間)。觀察client程式的輸出訊息
- 執行 **netstat | grep 9877**。指認 server/client 的 TCP state。

Orphan (1/2)

- 承前。確認 `./tcpserver04` 已在執行。
- 執行 echo client: `./tcpcli01 127.0.0.1`
- 另開一終端機程式執行 `ps -ejf | grep tcpserver04` 找出 `parent` process id 並將其 kill 掉
- 執行 `netstat | grep 9877`。指認 server child 與 client 的 TCP states 各為何?
- 測試 client 程式是否能順利執行?

Orphan (2/2)

- 承前。在client 程式按 Ctrl+D 結束執行。觀察系統給的訊息。
- 執行 **netstat | grep 9877**。指認 server child 與 client 的 TCP states 各為何?

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 unpv.h 里面新加入一行自定義的MACRO變數
 - #define OPEN_MAX 1024