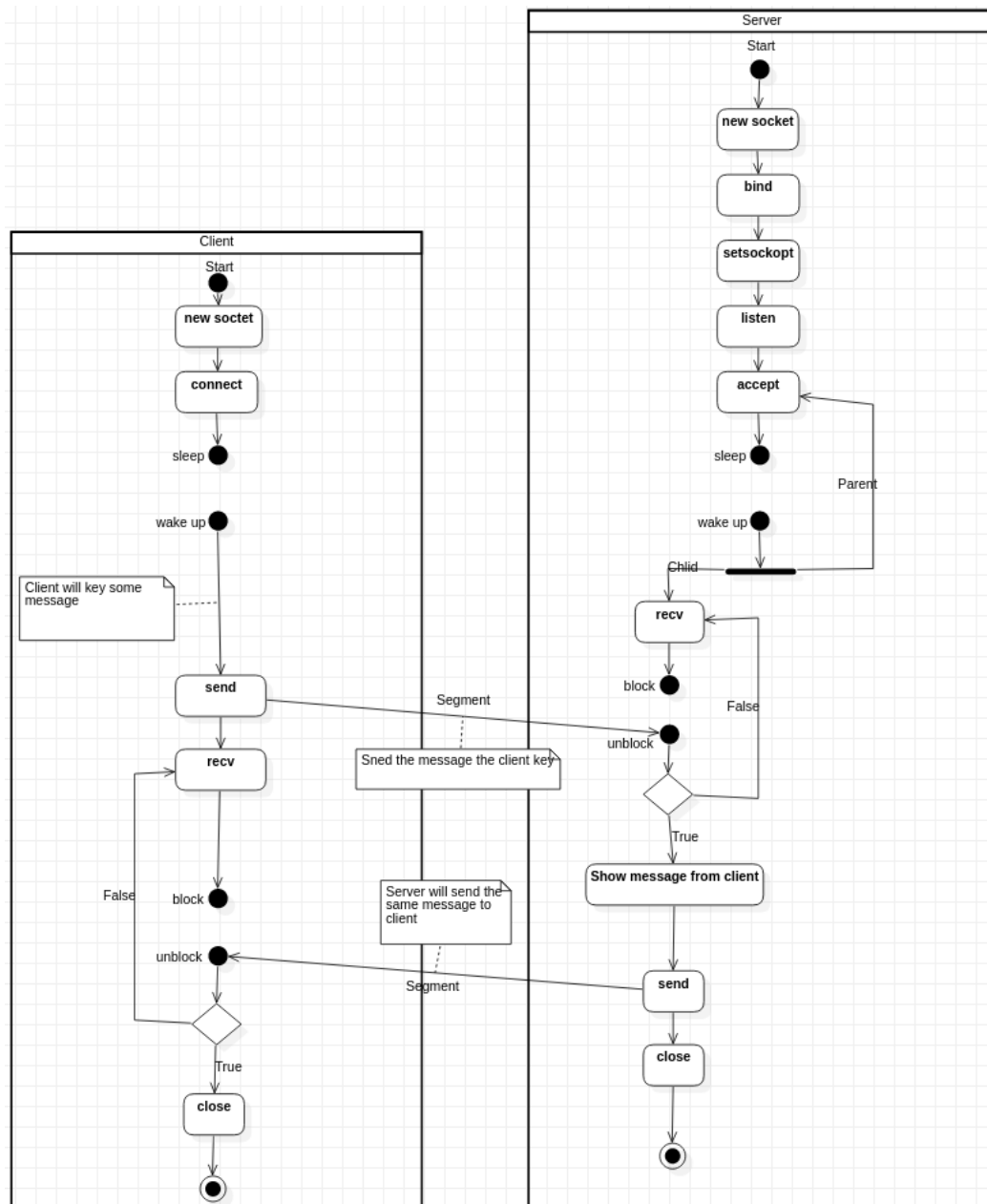


- 程式說明
  - 程式流程圖



- server：先 create socket，setting address 並把他們 bind 在一起，之後開始 listen，等待 client 的連結，等到傳來連結消息後就 accept 並把連到的主機和 port number 列印出來，完成後就利用 threading 創建出 child process 來同時與多個 client 做訊息的交換，最後清除 socket。直到 server 被關閉前，這個程式都不會停下來。
- client：先 create socket，setting address，接著 connect 到 server 並輸出使用者輸入的字串之後從 server 接收相同的字串，最後清除 socket 就完成了。

- 執行結果與說明

- Client: 連接到 server 後，輸入一字串後，傳送到 server 後，在接受 server 相同的字串，最後輸出字串。
- Server: 接受到 client 的訊號後，輸出 client 所傳入的字串，再回傳一個相同的字串給 server。
- 執行 Server.py。

```
timmy@timmy:~/myCode/Socket/Demo4$ python3 Server.py
Server is on
Server is waiting for connection...
```

- 執行 Client.py。

```
timmy@timmy:~/myCode/Socket/Demo4$ python Client.py
Type your message (Type 'exit' to close): _
```

- Client 成功連接 Server。

```
timmy@timmy:~/myCode/Socket/Demo4$ python3 Server.py
Server is on
Server is waiting for connection...
New Connection: ('192.168.3.139', 45460)
Connection from ('192.168.3.139', 45460)
```

- Client 傳入字串 “hello” 給 server，並成功收到 server 相同的字串。

```
timmy@timmy:~/myCode/Socket$ /usr/bin/python3 /home/timmy/myCode/Socket/Demo
4/Client.py
Type your message (Type 'exit' to close): hello
From server: hello
```

- Server 成功收到 Client 的字串。

```
timmy@timmy:~/myCode/Socket/Demo4$ python3 Server.py
Server is on
Server is waiting for connection...
New Connection: ('192.168.3.139', 45522)
Connection from ('192.168.3.139', 45522)
From client ('192.168.3.139', 45522): hello
```

- Client 傳入字串 “exit” 給 server，Client 離開。

```
timmy@timmy:~/myCode/Socket$ /usr/bin/python3 /home/timmy/myCode/Socket/Demo
4/Client.py
Type your message (Type 'exit' to close): hello
From server: hello
-----
Please type new message: (type 'exit' to close)exit
The client leave...
```

- Server 顯示 Client 已離線。

```
timmy@timmy:~/myCode/Socket/Demo4$ python3 Server.py
Server is on
Server is waiting for connection...
New Connection: ('192.168.3.139', 58078)
Connection from ('192.168.3.139', 58078)
From client ('192.168.3.139', 58078): hello
Client ('192.168.3.139', 58078) disconnected.
```

- 遭遇的困難與解決方法

1. 在 windows 環境設計此程式相較於 linux 下我覺得要難很多，首先在 include 的部分就與課本寫的相差很多，因為有的函示庫 windows 並沒有。
2. 處理最久的是 createprocess 這個函式我查網路說是最類似 fork 的東西，但它竟然有十個參數，接著又花好多時間學了它的用法，使用並創建新的 process 時出現了 "bind sock failed"（我用來偵錯的字條）導致沒辦法與多個客戶端連結。

- 解決方法：

1. 查詢一下找到了 <ws2tcpip> 這個函示庫和使用 #pragma comment(lib," ws2\_32.lib") 來處理這個問題。
  2. 後來發現原本的 bind socket 被沿用下來了於是將 createprocess 第五個參數(用來判斷是否要繼承 parent process 的屬性)從 true 改成 false 使 child process 能用他自己的 bind socket 才得以讓程式連結多個 client。
3. 我在執行時遇到：accept() fails with Operation not supported
    - solution:
      - 原本寫法：server = socket.socket(socket.AF\_INET,socket.SOCK\_DGRAM)
      - 更改寫法：server = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)
      - 說明：
        - socket.SOCK\_DGRAM is for UDP
        - socket.SOCKSTREAM is for TCP
  4. 我在執行時遇到：“ConnectionRefusedError: [Errno 111] Connection refused.”
    - Solution:
      - 原本寫法：server.bind((' ', port)) #Bind to the port
      - 更改寫法：server.bind(('server.bind((HOST, PORT))'))
      - 說明：更改後的寫法，會讓 socket 不會這限制。如果只連接到 127.0.0.1，這會讓 server 無法連接到其他不一樣的 host。
  5. 我在執行時遇到：“Network is unreachable”
    - solution: 因為我的 server host 是 192 開頭，是私人網路，無法讓外部的 client 連接到，要用 NAT 轉換成公共網路，才可以成功連接。