

## 【2021 Advanced Computer Networks Homework 6】

### 【簡介】

使用 c 語言建立 2 個 socket，一個傳送 priority data (指令)，另一個傳送 normal data (使用者輸入的資料)。

成功連線後，client 建立一個 thread，此 thread 會不斷的傳送 data 給 server；而 main function 會等待使用者輸入指令，如果 client 有傳送指令的話，那 server 會優先處理。

### 【指令】

編譯

**make**

mininet 建立兩個 hosts

**sudo mn**

**xterm h1 h2**

執行 server

**./hw6\_server**

執行 client

**./hw6\_client**

client 端指令

**start** (開始傳送 data)

**stop** (暫停傳送 data)

**quit** (取消連線)

## 【執行結果】

1. 在 client 端輸入 **server ip**

The screenshot shows a Visual Studio Code editor with a C program named `hw6_server.c` open. The program is a TCP server that listens on port 4461 and handles multiple clients. The code includes headers for `stdio.h`, `stdlib.h`, `unistd.h`, `sys/types.h`, `sys/socket.h`, `netinet/in.h`, and `arpa/inet.h`. It defines a `MAX_CLIENTS` constant and uses `select` for non-blocking I/O. The server socket is created with `AF_INET` and `SOCK_STREAM`. It uses `getsockname` to get the local address and `getpeername` to get the client address. The server sends a "normal socket: connected to the server.." message and receives data from the client, which it then prints to the console.

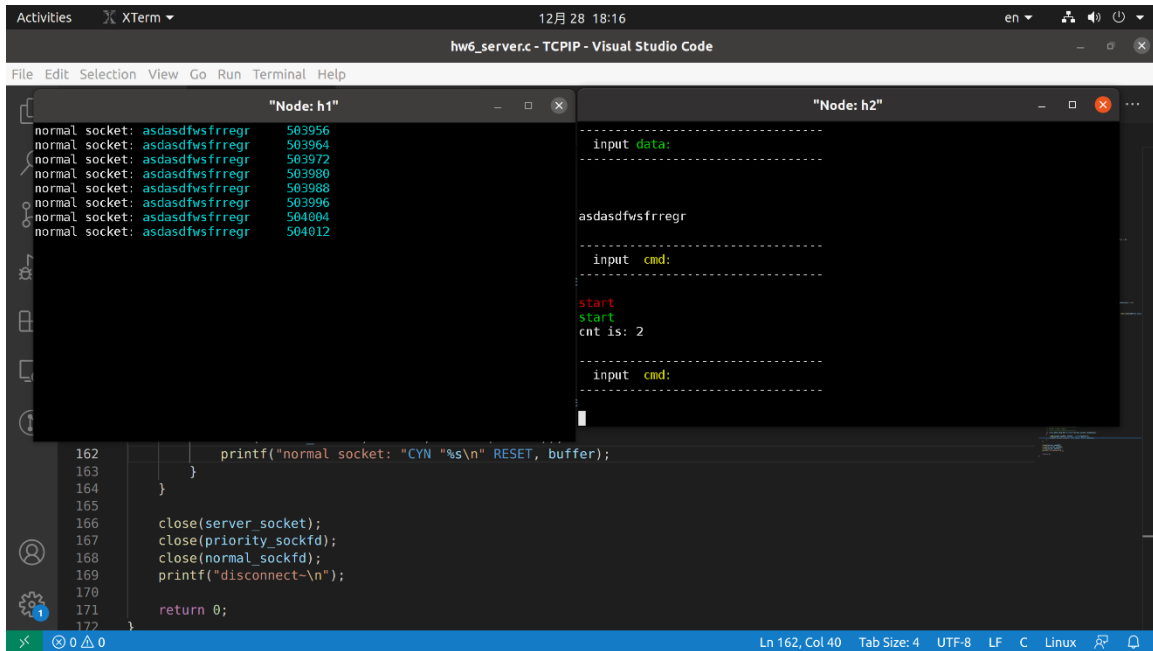
The output window shows the server's execution. It starts by listening on port 4461. When a client connects, it prints the client's IP address (10.0.0.2) and the socket file descriptor (12). It then receives data from the client and prints it to the console. The output shows that the client has connected and is ready to receive data.

## 2. 在 client 端輸入要傳送的 data

### 3. 在 client 端輸入指令

- **start**

client 開始不斷地傳送前面輸入的 data 給 server，並在每個 data 或指令後面都會加上一個 counter，記錄目前送出了幾個資料。



```
normal socket: asdasdfwsfrregr 503956
normal socket: asdasdfwsfrregr 503964
normal socket: asdasdfwsfrregr 503972
normal socket: asdasdfwsfrregr 503980
normal socket: asdasdfwsfrregr 503988
normal socket: asdasdfwsfrregr 503996
normal socket: asdasdfwsfrregr 504004
normal socket: asdasdfwsfrregr 504012

input data:
-----
asdasdfwsfrregr

input cmd:
-----
start
start
cnt is: 2

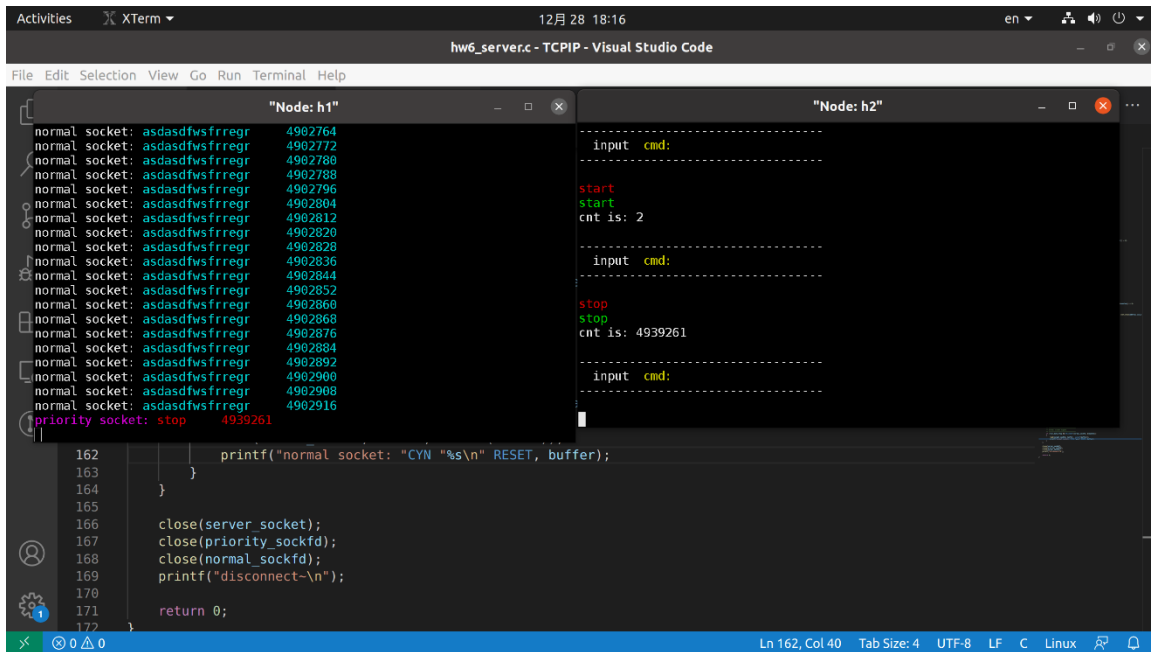
input cmd:
-----
```

```
162     printf("normal socket: \"CYN \"%s\\n\" RESET, buffer);
163 }
164 }
165
166 close(server_socket);
167 close(priority_sockfd);
168 close(normal_sockfd);
169 printf("disconnect-\\n");
170
171 return 0;
172 }
```

## ● stop

client 端暫停傳送 data，並通知 server

從 stop 的 counter 值可看出，在 server 最後一個收到的 data 中還有很多 data 還沒接收，但因為是 priority data，所以會優先處理



```
File Edit Selection View Go Run Terminal Help
hw6_server.c - TCPIP - Visual Studio Code

"Node: h1"
normal socket: asdasdfvsfrregr 4902764
normal socket: asdasdfvsfrregr 4902772
normal socket: asdasdfvsfrregr 4902780
normal socket: asdasdfvsfrregr 4902788
normal socket: asdasdfvsfrregr 4902796
normal socket: asdasdfvsfrregr 4902804
normal socket: asdasdfvsfrregr 4902812
normal socket: asdasdfvsfrregr 4902820
normal socket: asdasdfvsfrregr 4902828
normal socket: asdasdfvsfrregr 4902836
normal socket: asdasdfvsfrregr 4902844
normal socket: asdasdfvsfrregr 4902852
normal socket: asdasdfvsfrregr 4902860
normal socket: asdasdfvsfrregr 4902868
normal socket: asdasdfvsfrregr 4902876
normal socket: asdasdfvsfrregr 4902884
normal socket: asdasdfvsfrregr 4902892
normal socket: asdasdfvsfrregr 4902900
normal socket: asdasdfvsfrregr 4902908
normal socket: asdasdfvsfrregr 4902916
priority socket: stop 4939261

162     printf("normal socket: \"CYN \"%s\\n\" RESET, buffer);
163     }
164 }
165
166 close(server_socket);
167 close(priority_sockfd);
168 close(normal_sockfd);
169 printf("disconnect-\\n");
170
171 return 0;
172 }

"Node: h2"
-----
input cmd:
-----
start
start
cnt is: 2
-----
input cmd:
-----
stop
stop
cnt is: 4939261
-----
input cmd:
-----

Ln 162, Col 40 Tab Size: 4 UTF-8 LF C Linux
```

● quit  
client 送出 quit 後就斷開連線

The screenshot shows the Visual Studio Code editor with a C program named `hw6_server.c` and two terminal windows. The editor window shows the following code:

```
162     }  
163     }  
164     }  
165  
166     close(server_socket);  
167     close(priority_sockfd);  
168     close(normal_sockfd);  
169     printf("disconnect-\n");  
170  
171     return 0;  
172 }
```

The terminal window titled `"Node: h1"` shows the following output:

```
normal socket: asdasdfvsfrregr 5594334  
normal socket: asdasdfvsfrregr 5594342  
normal socket: asdasdfvsfrregr 5594350  
normal socket: asdasdfvsfrregr 5594358  
normal socket: asdasdfvsfrregr 5594366  
normal socket: asdasdfvsfrregr 5594374  
normal socket: asdasdfvsfrregr 5594382  
normal socket: asdasdfvsfrregr 5594390  
normal socket: asdasdfvsfrregr 5594398  
normal socket: asdasdfvsfrregr 5594406  
normal socket: asdasdfvsfrregr 5594414  
normal socket: asdasdfvsfrregr 5594422  
normal socket: asdasdfvsfrregr 5594430  
normal socket: asdasdfvsfrregr 5594438  
normal socket: asdasdfvsfrregr 5594446  
normal socket: asdasdfvsfrregr 5594454  
normal socket: asdasdfvsfrregr 5594462  
priority socket: stop 5637894  
priority socket: quit 5637895  
priority socket: quit 5637895  
disconnect-
```

The terminal window titled `"Node: h2"` shows the following output:

```
start  
start  
cnt is: 4939316  
-----  
input cmd:  
-----  
stop  
stop  
cnt is: 5637094  
-----  
input cmd:  
-----  
quit  
quit  
cnt is: 5637095  
root@miksuki-BM6AE-BM1AE-BP1AE: /home/miksuki/Desktop/TCPIP/hw6#
```