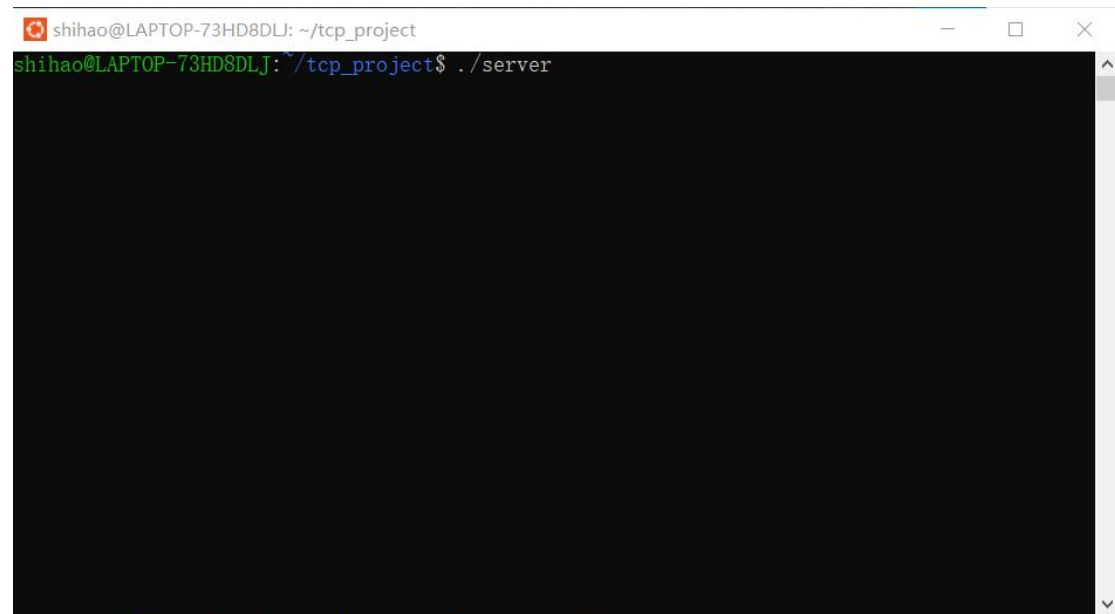


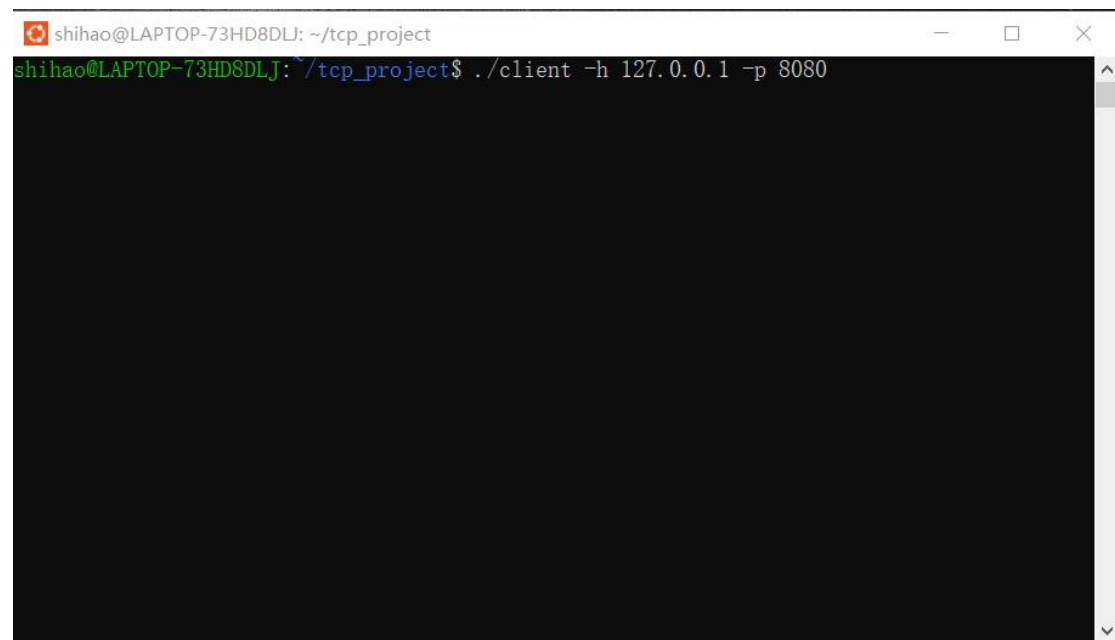
TCP PROJECT REPORT

Run the program:



```
shihao@LAPTOP-73HD8DLJ: ~/tcp_project
shihao@LAPTOP-73HD8DLJ: ~/tcp_project$ ./server
```

A terminal window with a white title bar containing the text "shihao@LAPTOP-73HD8DLJ: ~/tcp_project" and standard window controls. The terminal area has a black background. The prompt "shihao@LAPTOP-73HD8DLJ: ~/tcp_project\$" is shown in green, and the command "./server" is entered in blue. The rest of the terminal area is empty.



```
shihao@LAPTOP-73HD8DLJ: ~/tcp_project
shihao@LAPTOP-73HD8DLJ: ~/tcp_project$ ./client -h 127.0.0.1 -p 8080
```

A terminal window with a white title bar containing the text "shihao@LAPTOP-73HD8DLJ: ~/tcp_project" and standard window controls. The terminal area has a black background. The prompt "shihao@LAPTOP-73HD8DLJ: ~/tcp_project\$" is shown in green, and the command "./client -h 127.0.0.1 -p 8080" is entered in blue. The rest of the terminal area is empty.

RESULT: (3 client)

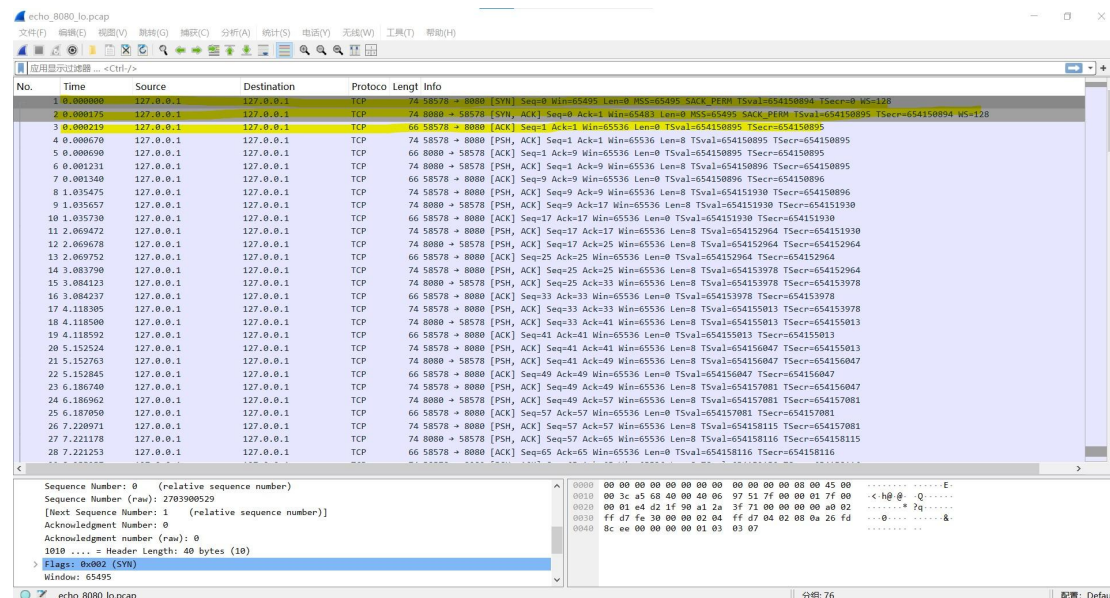
```
shihao@LAPTOP-73HD8DLJ: ~/tcp_project
182      fprintf(stderr, pthread_create: %s\n", strerror(rc));
server.c:182:46: warning: format '%s' expects argument of type 'char *', but argument 3
has type 'int' [-Wformat=] 182      fprintf(stderr, pthread_create: %s\n", strerror
(rc));
                                     |
                                     v
                                     int
                                     |
                                     char *
                                     v
make: *** [Makefile:7: server] Error 1
shihao@LAPTOP-73HD8DLJ: ~/tcp_project$ code .
shihao@LAPTOP-73HD8DLJ: ~/tcp_project$ make
gcc -Wall -Wextra -O2 server.c -o server
shihao@LAPTOP-73HD8DLJ: ~/tcp_project$ ./server
server: waiting for connections...
server: got connection from 127.0.0.1
server: got connection from 127.0.0.1
server: got connection from 127.0.0.1

client: sent: count=31
client: rcv: count=31
client: sent: count=32
client: rcv: count=32
client: sent: count=33
client: rcv: count=33
client: sent: count=34
client: rcv: count=34
client: sent: count=35
client: rcv: count=35
client: sent: count=36
client: rcv: count=36
client: sent: count=37
client: rcv: count=37
client: sent: count=38
client: rcv: count=38
client: sent: count=39
client: rcv: count=39
client: sent: count=40
client: rcv: count=40
client: sent: count=41
client: rcv: count=41

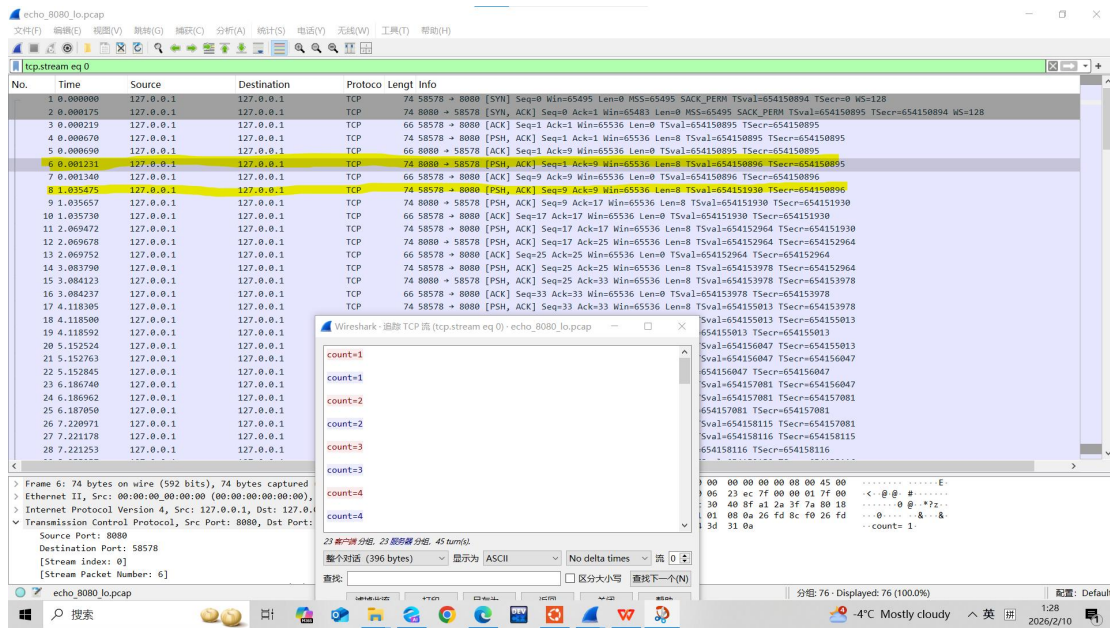
client: rcv: count=196
client: sent: count=197
client: rcv: count=197
client: sent: count=198
client: rcv: count=198
client: sent: count=199
client: rcv: count=199
client: sent: count=200
client: rcv: count=200
client: sent: count=201
client: rcv: count=201
client: sent: count=202
client: rcv: count=202
client: sent: count=203
client: rcv: count=203
client: sent: count=204
client: rcv: count=204
client: sent: count=205
client: rcv: count=205
client: sent: count=206
client: rcv: count=206
client: sent: count=207
client: rcv: count=207

client: sent: count=14
client: rcv: count=15
client: sent: count=15
client: rcv: count=16
client: sent: count=16
client: rcv: count=17
client: sent: count=17
client: rcv: count=18
client: sent: count=18
client: rcv: count=19
client: sent: count=19
client: rcv: count=20
client: sent: count=20
client: rcv: count=21
client: sent: count=21
client: rcv: count=22
client: sent: count=22
client: rcv: count=23
client: sent: count=23
client: rcv: count=24
client: sent: count=24
client: rcv: count=24
```

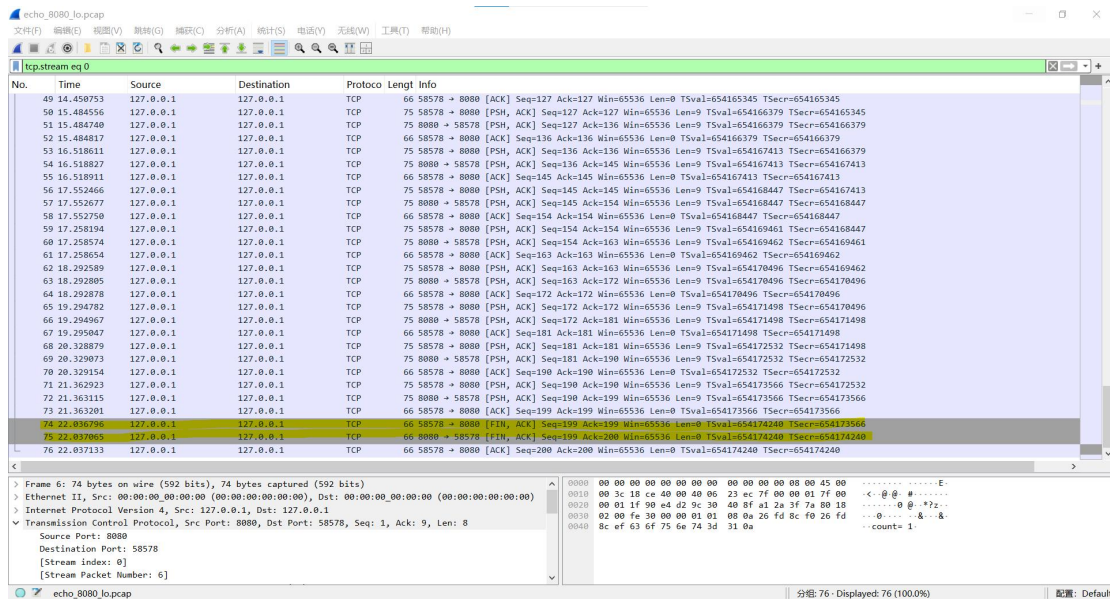
packet capture in Wireshark:



TCP 3-way handshake: the client initiates with SYN, the server replies SYN-ACK, and the client confirms with ACK, establishing the connection.



Echo data: The client sends application payload to the server (PSH, ACK with Len=8), and the server echoes the same payload back to the client (PSH, ACK in the opposite direction).



Graceful close (FIN): the client initiates termination with FIN,ACK, the server responds with FIN,ACK, and the final ACK completes the connection teardown. On the server side, this corresponds to `recv()` returning 0 (peer performed an orderly shutdown).