



LAB 5

Socket Programming

Họ tên và MSSV: Trần Quốc Phú B2205899

Nhóm học phần: CT29301

- Các sinh viên bị phát hiện sao chép bài của nhau sẽ nhận 0đ cho tất cả bài thực hành của môn này.
- Bài nộp phải ở dạng PDF, hình minh họa phải rõ ràng chi tiết. Hình minh họa chỉ cần chụp ở nội dung thực hiện, không chụp toàn màn hình.

1. Bài 1

Viết chương trình nhận đối số là một URL (Ví dụ: www.cit.ctu.edu.vn). Sử dụng TCP socket nối kết đến web server trong URL để lấy file HTML về, và hiển thị nội dung file HTML đó ra màn hình.

(Chụp hình minh họa code bài làm và kết quả thực thi; đính kèm tập tin code khi nộp bài.)

Code bài làm:

```
1  import socket
2  import ssl
3  import sys
4  import webbrowser
5  import os
```

```
1 def fetch_html_https(url): #Hàm chính để lấy HTML từ trang web
2     if not url.startswith("http"):
3         host = url
4     else:
5         host = url.split("//")[1]
6     port = 443 # Cổng HTTPS mặc định
7     #Kiểm tra và kết nối
8     try:
9         # Tạo socket TCP
10        raw_socket = socket.create_connection((host, port))
11        context = ssl.create_default_context()
12
13        # Bọc socket với SSL
14        secure_socket = context.wrap_socket(raw_socket, server_hostname=host)
15
16        # Gửi HTTP GET request
17        request = f"GET / HTTP/1.1\r\nHost: {host}\r\nConnection: close\r\n\r\n"
18        # Gửi yêu cầu
19        secure_socket.send(request.encode())
20
21        # Nhận phản hồi
22        response = b""
23        while True:
24            data = secure_socket.recv(4096)
25            if not data:
26                break
27            response += data
28
29        secure_socket.close()
30
31        # Tách header và nội dung HTML
32        response_str = response.decode(errors="ignore")
33        #Chuyển đổi bytes thành chuỗi (errors="ignore"): nếu có bất kì bytes nào không chuyển đổi đc
34        #=> bỏ qua chứ không để lỗi
35        header_end = response_str.find("\r\n\r\n")
36        html_content = response_str[header_end + 4:]
37        return(html_content)
38
39    except socket.gaierror: #Xử lý ngoại lệ
40        print(" Không thể phân giải tên miền.")
41    except Exception as e:
42        print(f" Lỗi: {e}")
```

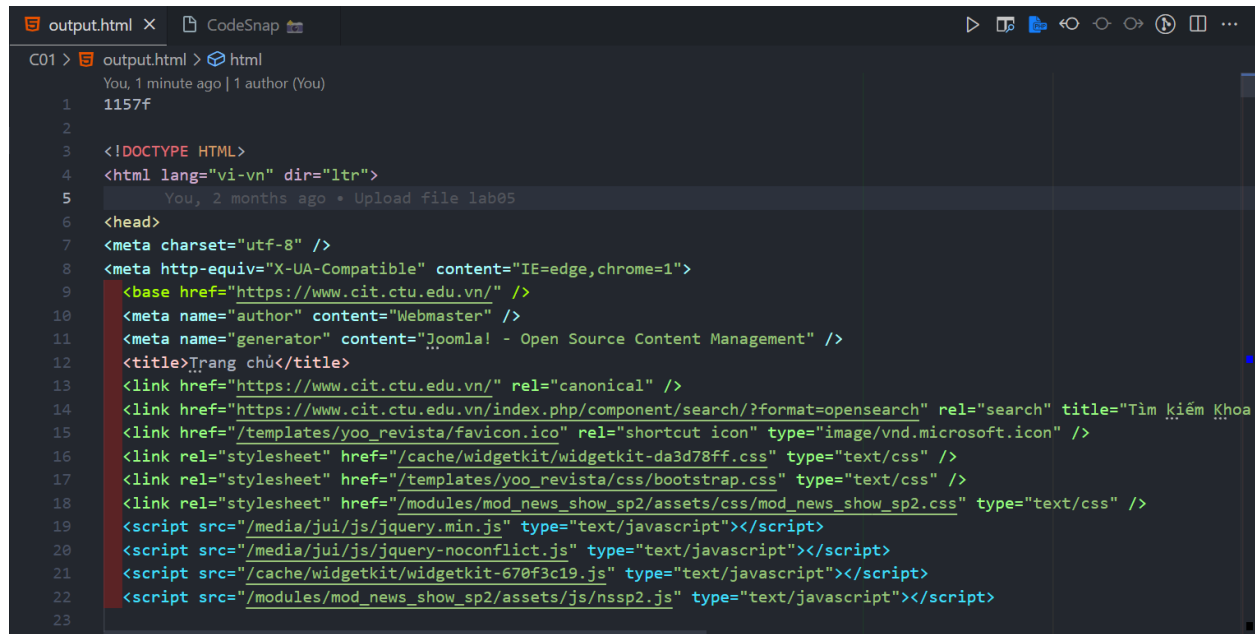
```
1  if __name__ == "__main__":
2      if len(sys.argv) != 2:
3          print("Cách sử dụng: python tcp_fetch_html.py <url>")
4          print("Ví dụ: python tcp_fetch_html.py www.cit.ctu.edu.vn")
5      else:
6          #fetch_html(sys.argv[1])
7          html = fetch_html_https(sys.argv[1])
8
9          if html.startswith("Lỗi:") or html.startswith("Không"):
10             print(html)
11          else:
12             # Lưu HTML vào file
13             with open("output.html", "w", encoding="utf-8") as f:
14                 f.write(html)
15
16             # Mở file HTML trong trình duyệt
17             webbrowser.open(f"file://{os.path.abspath('output.html')}")
```

Khi chạy chương trình kết quả (mã html của trang web) sẽ được lưu vào file output.html

```
1  %%bash
2  python tcp_fetch_html.py www.cit.ctu.edu.vn
```

Lệnh chạy chương trình (url truyền vào www.cit.ctu.edu.vn)

Kết quả:



```
output.html x CodeSnap
C01 > output.html > html
You, 1 minute ago | 1 author (You)
1157f
1
2
3 <!DOCTYPE HTML>
4 <html lang="vi-vn" dir="ltr">
5   You, 2 months ago • Upload file lab05
6 <head>
7 <meta charset="utf-8" />
8 <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
9 <base href="https://www.cit.ctu.edu.vn/" />
10 <meta name="author" content="Webmaster" />
11 <meta name="generator" content="Joomla! - Open Source Content Management" />
12 <title>Trang chủ</title>
13 <link href="https://www.cit.ctu.edu.vn/" rel="canonical" />
14 <link href="https://www.cit.ctu.edu.vn/index.php/component/search/?format=opensearch" rel="search" title="Tìm kiếm Khoa
15 <link href="/templates/yoo_revista/favicon.ico" rel="shortcut icon" type="image/vnd.microsoft.icon" />
16 <link rel="stylesheet" href="/cache/widgetkit/widgetkit-da3d78ff.css" type="text/css" />
17 <link rel="stylesheet" href="/templates/yoo_revista/css/bootstrap.css" type="text/css" />
18 <link rel="stylesheet" href="/modules/mod_news_show_sp2/assets/css/mod_news_show_sp2.css" type="text/css" />
19 <script src="/media/jui/js/jquery.min.js" type="text/javascript"></script>
20 <script src="/media/jui/js/jquery-noconflict.js" type="text/javascript"></script>
21 <script src="/cache/widgetkit/widgetkit-670f3c19.js" type="text/javascript"></script>
22 <script src="/modules/mod_news_show_sp2/assets/js/nssp2.js" type="text/javascript"></script>
23
```

Sau khi chạy lệnh trên thì kết quả là hiện lên file output.html như trên hình.

2. Bài 2

Viết chương trình theo mô hình Client-Server sử dụng TCP Socket.

Trong đó:

- + Server lắng nghe ở cổng 8888, làm nhiệm vụ đọc một ký tự số từ '0' đến '9'.
(Ví dụ: nhận số 0, trả về "khong"; 1, trả về "mot" ; 9, trả về "chin"; nhận ký tự khác số thì trả về "Không phải số nguyên").
- + Client sẽ nhập vào 1 ký tự, gửi qua Server, nhận kết quả trả về từ Server và thể hiện lên màn hình.

(Chụp hình minh họa code bài làm và kết quả thực thi; đính kèm tập tin code khi nộp bài.)

Code bài làm:

```
1  import socket
2
3  HOST = 'localhost' # hoặc IP của server
4  PORT = 8888
5
6  def main():
7      # Nhập 1 ký tự
8      ch = input("Nhập một ký tự số (0-9): ").strip()
9
10     # Tạo socket TCP
11     client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
12     client_socket.connect((HOST, PORT))
13
14     client_socket.send(ch.encode())
15     response = client_socket.recv(1024).decode()
16
17     print("Phản hồi từ Server:", response)
18
19     client_socket.close()
20
21 if __name__ == "__main__":
22     main()
```

tcp_client.py

```
1  import socket
2
3  # Bảng ánh xạ số sang chữ
4  num_to_word = {
5      '0': "khong",
6      '1': "mot",
7      '2': "hai",
8      '3': "ba",
9      '4': "bon",
10     '5': "nam",
11     '6': "sau",
12     '7': "bay",
13     '8': "tam",
14     '9': "chin"
15 }
16
17 HOST = '0.0.0.0' # Lắng nghe trên tất cả các địa chỉ IP
18 PORT = 8888
19
20
21 def main():
22     # Tạo socket TCP
23     server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
24     server_socket.bind((HOST, PORT))
25     server_socket.listen(5)
26     print(f"Server đang lắng nghe tại cổng {PORT}...")
27
28     while True:
29         conn, addr = server_socket.accept()
30         print(f"Client kết nối từ {addr}")
31
32         data = conn.recv(1024).decode().strip()
33         print(f"Nhận từ client: '{data}'")
34
35         if data in num_to_word:
36             response = num_to_word[data]
37         else:
38             response = "Không phải số nguyên"
39
40         conn.send(response.encode())
41         conn.close()
42
43 if __name__ == "__main__":
44     main()
```

tcp_server.py

Kết quả thực thi

Phía client:

```
D:\MangTTDL_CT293\NopLab\Lab5\C02>python tcp_client.py
Nhập một ký tự số (0-9): 9
Phản hồi từ Server: chín

D:\MangTTDL_CT293\NopLab\Lab5\C02>
```

Phía Server:

```
D:\MangTTDL_CT293\NopLab\Lab5\C02>python tcp_server.py
Server đang lắng nghe tại cổng 8888...
Client kết nối từ ('127.0.0.1', 58559)
Nhận từ client: '9'
█
```

Hình ảnh cả 2 terminal:

```
D:\MangTTDL_CT293\NopLab\Lab5\C02>python tcp_client.py
Nhập một ký tự số (0-9): 6
Phản hồi từ Server: sáu

D:\MangTTDL_CT293\NopLab\Lab5\C02>
```

```
D:\MangTTDL_CT293\NopLab\Lab5\C02>python tcp_server.py
Server đang lắng nghe tại cổng 8888...
Client kết nối từ ('127.0.0.1', 59358)
Nhận từ client: '6'
█
```

3. Bài 3

Viết chương trình theo mô hình Client-Server sử dụng UDP Socket.

Trong đó:

- + Server hoạt động ở cổng 8888, sẽ nhận từ Client các yêu cầu là một chuỗi có khuôn dạng như sau:

"OP Operant1 Operant2\n" Ví dụ: "+ 100 200"

Trong đó:

- OP là một ký tự chỉ phép toán muốn thực hiện: '+', '-', '*', '/'
- Operant1, Operant2 là đối số của phép toán.
- Các thành phần trên cách nhau bởi 1 ký tự trắng ' '
- Mỗi khi server nhận được một thông điệp nó sẽ thực hiện phép toán "Operant1 OP Operant2" để cho ra kết quả sau đó đổi kết quả thành chuỗi và gửi về Client.
- + Client cho phép người dùng nhập các phép toán muốn tính theo cách thức thông thường, ví dụ: 100 + 200. Client tạo ra thông điệp yêu cầu theo đúng dạng do Server quy định rồi gửi sang Server, chờ nhận kết quả trả về và in ra màn hình.
- + Kết thúc yêu cầu bằng ký tự xuống dòng '\n'

(Chụp hình minh họa code bài làm và kết quả thực thi; đính kèm tập tin code khi nộp bài.)

Code:

Phía client:

```
1 import socket
2
3 SERVER_IP = 'localhost'
4 # hoặc thay bằng địa chỉ IP của server nếu chạy khác máy
5 PORT = 8888
6
7 def parse_input(expr):
8     # Tách biểu thức kiểu: "100 + 200"
9     try:
10         parts = expr.strip().split()
11         if len(parts) != 3:
12             return None
13         operand1, operator, operand2 = parts
14         return f"{operator} {operand1} {operand2}\n"
15     except:
16         return None
17
18 def main():
19     client_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
20
21     while True:
22         expr = input("Nhập phép toán (ví dụ: 100 + 200), hoặc 'exit' để thoát: ")
23         if expr.lower() == 'exit':
24             break
25
26         message = parse_input(expr)
27         if not message:
28             print("Biểu thức không hợp lệ. Vui lòng thử lại.")
29             continue
30
31         client_socket.sendto(message.encode(), (SERVER_IP, PORT))
32         result, _ = client_socket.recvfrom(1024)
33         print("Kết quả:", result.decode())
34
35     client_socket.close()
36
37 if __name__ == "__main__":
38     main()
```

Phía Server:


```
1  import socket
2
3  HOST = '0.0.0.0'
4  PORT = 8888
5
6  def calculate(operation, op1, op2):
7      try:
8          op1 = float(op1)
9          op2 = float(op2)
10         if operation == '+':
11             return str(op1 + op2)
12         elif operation == '-':
13             return str(op1 - op2)
14         elif operation == '*':
15             return str(op1 * op2)
16         elif operation == '/':
17             if op2 == 0:
18                 return "Lỗi: chia cho 0"
19             return str(op1 / op2)
20         else:
21             return "Phép toán không hợp lệ"
22     except ValueError:
23         return "Đối số không hợp lệ"
24
25 def main():
26     server_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
27     server_socket.bind((HOST, PORT))
28     print(f"UDP Server đang chạy tại cổng {PORT}...")
29
30     while True:
31         message, client_addr = server_socket.recvfrom(1024)
32         msg = message.decode().strip()
33         print(f"Nhận từ {client_addr}: '{msg}'")
34
35         parts = msg.split()
36         if len(parts) != 3:
37             response = "Định dạng không hợp lệ"
38         else:
39             op, operand1, operand2 = parts
40             response = calculate(op, operand1, operand2)
41
42         server_socket.sendto(response.encode(), client_addr)
43
44 if __name__ == "__main__":
45     main()
```

Kết quả thực thi:

Phía client:

```
D:\MangTTDL_CT293\NopLab\Lab5\C03>python udp_client.py
Nhập phép toán (ví dụ: 100 + 200), hoặc 'exit' để thoát: 200 + 63
Kết quả: 263.0
Nhập phép toán (ví dụ: 100 + 200), hoặc 'exit' để thoát: █
```

Phía server:

```
D:\MangTTDL_CT293\NopLab\Lab5\C03>python udp_server.py
UDP Server đang chạy tại cổng 8888...
Nhận từ ('127.0.0.1', 63238): '+ 200 63'
█
```

Hình cả 2 terminal:

<pre>D:\MangTTDL_CT293\NopLab\Lab5\C03>python udp_server.py UDP Server đang chạy tại cổng 8888... Nhận từ ('127.0.0.1', 63238): '+ 200 63' █</pre>	<pre>D:\MangTTDL_CT293\NopLab\Lab5\C03>python udp_client.py Nhập phép toán (ví dụ: 100 + 200), hoặc 'exit' để thoát: 200 + 63 Kết quả: 263.0 Nhập phép toán (ví dụ: 100 + 200), hoặc 'exit' để thoát: █</pre>
---	--

Khi client gửi exit:

```
Nhập phép toán (ví dụ: 100 + 200), hoặc 'exit' để thoát: exit

D:\MangTTDL_CT293\NopLab\Lab5\C03> █
```

4. Bài 4

Viết chương trình theo mô hình Client-Server sử dụng TCP socket.

Trong đó,

- + Server hoạt động ở 2 cổng (2 TCP socket) 8001 và 8002:
 - Cổng 8000: sử dụng để nhận các lệnh từ Client và gửi phản hồi cho Client.
 - Cổng 8001: sử dụng để gửi dữ liệu tập tin/thư mục tới Client.
- + Ở cổng 8000, Server sẽ nhận từ Client các câu lệnh có dạng như sau:
 - "GET <Tên file>".
 - Nếu file yêu cầu tồn tại: Server sẽ gửi trả cho Client chuỗi "OK\n". Sau đó là nội dung file sẽ được gửi tới Client qua cổng 8001.
 - Nếu file không tồn tại: Server sẽ gửi trả cho Client chuỗi "ERROR\n".
 - Client nhận nội dung tập tin từ Server và lưu lại.
 - "DELETE <Tên file>".
 - Nếu file yêu cầu tồn tại: Server sẽ gửi trả cho Client chuỗi "OK\n". Xóa file đó trên thư mục hiện hành của Server.
 - Nếu file không tồn tại: Server sẽ gửi trả cho Client chuỗi "ERROR\n".

- "LIST <Tên thư mục>".
 - Nếu thư mục yêu cầu tồn tại: Server sẽ gửi trả cho Client chuỗi "OK\n". Sau đó là nội dung thư mục sẽ được gửi tới Client qua cổng 8001.
 - Nếu thư mục không tồn tại: Server sẽ gửi trả cho Client chuỗi "ERROR\n".
 - Client nhận nội dung thư mục từ Server và in ra màn hình.
- + Client sẽ cho phép người dùng gửi các lệnh qua Server và nhận phản hồi ở cổng 8000.
- + Kết thúc yêu cầu bằng ký tự xuống dòng '\n'

(Chụp hình minh họa code bài làm và kết quả thực thi; đính kèm tập tin code khi nộp bài.)

Code:

Phía client:

```
1 import socket
2 import os
3
4 SERVER_IP = 'localhost'
5 CMD_PORT = 8000
6 DATA_PORT = 8001
7
8 def receive_data_to_file(filename):
9     with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
10         s.connect((SERVER_IP, DATA_PORT))
11         # with open("client_" + filename, "wb") as f:
12         with open("client_" + os.path.basename(filename), "wb") as f:
13             while True:
14                 data = s.recv(1024)
15                 if not data:
16                     break
17                 f.write(data)
18     print(f"[CLIENT] Đã lưu vào client_{filename}")
19
20 def receive_data_to_screen():
21     with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
22         s.connect((SERVER_IP, DATA_PORT))
23         data = s.recv(4096)
24         print("[CLIENT] Nội dung thư mục:")
25         print(data.decode())
26
27 def main():
28     with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as cmd_socket:
29         cmd_socket.connect((SERVER_IP, CMD_PORT))
30         while True:
31             command = input("Nhập lệnh (GET/DELETE/LIST <tên file hoặc thư mục>) hoặc EXIT: ").strip()
32             if command.upper() == "EXIT":
33                 break
34
35             cmd_socket.sendall((command + "\n").encode())
36             response = cmd_socket.recv(1024).decode().strip()
37             print(f"[CLIENT] Phản hồi: {response}")
38
39             if response == "OK":
40                 if command.startswith("GET"):
41                     filename = command.split()[1]
42                     receive_data_to_file(filename)
43                 elif command.startswith("LIST"):
44                     receive_data_to_screen()
45
46 if __name__ == "__main__":
47     main()
```

Phía Server:



```
1  import socket
2  import threading
3  import os
4
5  HOST = '0.0.0.0'
6  CMD_PORT = 8000
7  DATA_PORT = 8001
```

```
1 def handle_command(conn_cmd, addr):
2     try:
3         while True:
4             data = conn_cmd.recv(1024).decode().strip()
5             if not data:
6                 break
7             print(f"[COMMAND] Nhận từ {addr}: {data}")
8
9             parts = data.split()
10            if len(parts) != 2:
11                conn_cmd.sendall("ERROR\n".encode())
12                continue
13
14            cmd, target = parts
15            if cmd == "GET":
16                if os.path.isfile(target):
17                    conn_cmd.sendall("OK\n".encode())
18                    # Gửi file qua cổng dữ liệu
19                    with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as data_socket:
20                        data_socket.bind((HOST, DATA_PORT))
21                        data_socket.listen(1)
22                        conn_data, _ = data_socket.accept()
23                        with open(target, "rb") as f:
24                            while True:
25                                chunk = f.read(1024)
26                                if not chunk:
27                                    break
28                                conn_data.sendall(chunk)
29                                conn_data.close()
30                else:
31                    conn_cmd.sendall("ERROR\n".encode())
32
33            elif cmd == "DELETE":
34                if os.path.isfile(target):
35                    os.remove(target)
36                    conn_cmd.sendall("OK\n".encode())
37                else:
38                    conn_cmd.sendall("ERROR\n".encode())
39
40            elif cmd == "LIST":
41                if os.path.isdir(target):
42                    conn_cmd.sendall("OK\n".encode())
43                    # Gửi nội dung thư mục qua cổng dữ liệu
44                    with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as data_socket:
45                        data_socket.bind((HOST, DATA_PORT))
46                        data_socket.listen(1)
47                        conn_data, _ = data_socket.accept()
48                        files = "\n".join(os.listdir(target))
49                        conn_data.sendall(files.encode())
50                        conn_data.close()
51                else:
52                    conn_cmd.sendall("ERROR\n".encode())
53
54            else:
55                conn_cmd.sendall("ERROR\n".encode())
56        except Exception as e:
57            print(f"Lỗi xử lý lệnh: {e}")
58        finally:
59            conn_cmd.close()
```

```
1 def main():
2     with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as server_cmd_socket:
3         server_cmd_socket.bind((HOST, CMD_PORT))
4         server_cmd_socket.listen(5)
5         print(f"[SERVER] Đang lắng nghe lệnh tại cổng {CMD_PORT}...")
6
7     while True:
8         conn_cmd, addr = server_cmd_socket.accept()
9         print(f"[SERVER] Kết nối mới từ {addr}")
10        threading.Thread(target=handle_command, args=(conn_cmd, addr)).start()
11
12 if __name__ == "__main__":
13     main()
```

Kết quả thực thi:

Phía client:

GET

```
D:\MangTTDL_CT293\NopLab\Lab5\C04>python tcp_client_dual.py
Nhập lệnh (GET/DELETE/LIST <tên file hoặc thư mục>) hoặc EXIT: GET Biba/abc.txt
[CLIENT] Phản hồi: OK
[CLIENT] Đã lưu vào client_Biba/abc.txt
Nhập lệnh (GET/DELETE/LIST <tên file hoặc thư mục>) hoặc EXIT: █
```

DELETE

```
Nhập lệnh (GET/DELETE/LIST <tên file hoặc thư mục>) hoặc EXIT: DELETE Biba/xyz.txt
[CLIENT] Phản hồi: OK
Nhập lệnh (GET/DELETE/LIST <tên file hoặc thư mục>) hoặc EXIT: █
```

LIST

```
Nhập lệnh (GET/DELETE/LIST <tên file hoặc thư mục>) hoặc EXIT: LIST Biba
[CLIENT] Phản hồi: OK
[CLIENT] Nội dung thư mục:
abc.txt
phutv.txt
Nhập lệnh (GET/DELETE/LIST <tên file hoặc thư mục>) hoặc EXIT: █
```

Phía server:

GET

```
D:\MangTTDL_CT293\NopLab\Lab5\C04>python tcp_server_dual.py
[SERVER] Đang lắng nghe lệnh tại cổng 8000...
[SERVER] Kết nối mới từ ('127.0.0.1', 57280)
[COMMAND] Nhận từ ('127.0.0.1', 57280): GET Biba/abc.txt
█
```

DELETE

```
[COMMAND] Nhận từ ('127.0.0.1', 57280): DELETE Biba/xyz.txt
█
```

LIST

```
[COMMAND] Nhận từ ('127.0.0.1', 57280): LIST Biba
█
```

--- Hết ---