



LAB 5

Socket Programming

Họ tên và MSSV: Cao Đức An - B1910027

Nhóm học phần: CT293 - 03

- 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.

- Code bài làm:

```
cau1.py > ...
1  from socket import *
2  s = socket(AF_INET, SOCK_STREAM)
3  print("nhập URL của webside mà bạn muốn!!!!")
4  URL = input();
5  s.connect((URL, 80)) # Connect
6  s.send(b'GET / HTTP/1.0\n\n') # Send request(in byte
7  data = s.recv(100000) # Get response
8  print(data.decode())
9  s.close() # Close the socket
10
```

-
- Kết quả thực thi:

```
PS D:\2019-2023\hk2 2021-2022\CT293\thuc_hanh\lab5> py cau1.py
nhập URL của webside mà bạn muốn!!!!
www.cit.ctu.edu.vn
HTTP/1.1 200 OK
Date: Tue, 03 May 2022 02:06:47 GMT
Server: Apache
P3P: CP="NOI ADM DEV PSAi COM NAV OUR OTRo STP IND DEM"
Cache-Control: no-cache
Pragma: no-cache
Vary: Accept-Encoding
Connection: close
Content-Type: text/html; charset=utf-8
Set-Cookie: 16b4a5845789fbf4955d2869fd3a47a5=j3auajltqhoeic2m3l06a88763; path=/; HttpOnly
Set-Cookie: BNES_16b4a5845789fbf4955d2869fd3a47a5=RpH42gbfPTC6t+e7q0a4JlyjDvkY0dkDc6pif0V2Af4

<!DOCTYPE HTML>
<html lang="vi-vn" dir="ltr">

<head>
<meta charset="utf-8" />
<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
  <meta name="author" content="Webmaster" />
  <meta name="generator" content="Joomla! - Open Source Content Management" />
<title>Trang nhất</title>
```

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

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.

- Code bài làm của server:

```
cau2_server.py > convert
1  from socket import *
2  import string
3  from xml.etree.ElementTree import tostring
4  def convert(number):
5      if(number.isdigit()):
6          switcher = {
7              0: 'Không',
8              1: 'Một',
9              2: 'Hai',
10             3: 'Ba',
11             4: 'Bốn',
12             5: 'Năm',
13             6: 'Sáu',
14             7: 'Bảy',
15             8: 'Tám',
16             9: 'Chín',
17         }
18         return switcher.get(int(number))
19     else:
20         return "Không phải là số"
21
22 s = socket(AF_INET, SOCK_STREAM)
23 host = "localhost"
24 port = 8888
25 s.bind((host, port)) # Connect
26 s.listen(96)
27 print("Server on listening on port :", port)
28
29 while True:
30     print('waiting for a connection')
31     connection, client_address = s.accept() # Wait for a connection
```

```
32     try:
33         print('connection from', client_address)
34         while True: # Receive the data in small chunks and retransmit
35             data = connection.recv(64)
36             if (data):
37                 data = convert(data);
38                 print('sending data back to the client', data)
39                 connection.sendall(data.encode())
40             else:
41                 print('no data from', client_address)
42                 break
43         finally: # Clean up the connection
44             connection.close()
45
46
47 s.close() # Close the socket
```

- Code bài làm bên client

```
cau2_client.py > ...
1  from socket import *
2
3  import sys
4
5  from numpy import number
6  # Create a TCP/IP socket
7  s = socket(AF_INET, SOCK_STREAM)
8  # Connect the socket to the server listening port
9  server_address = ('localhost', 8888)
10 s.connect(server_address)
11
12 try:
13     # Send data
14     print("nhập vào kí tự mà bạn muốn gửi qua server!!!!")
15     number = input()
16     s.sendall(number.encode())
17     # Look for the response
18     data = s.recv(64)
19     print('kết quả trả về từ server là:', data.decode())
20 finally:
21     print('closing socket')
22     s.close()
23
```

- Kết quả thực thi Bên client

```
PS D:\2019-2023\hk2 2021-2022\CT293\thuc_hanh\lab5> py .\cau2_client.py
nhập vào kí tự mà bạn muốn gửi qua server!!!!
5
kết quả trả về từ server là: Năm
closing socket
PS D:\2019-2023\hk2 2021-2022\CT293\thuc_hanh\lab5>
```

-
- Kết quả thực thi bên server

```
PS D:\2019-2023\hk2 2021-2022\CT293\thuc_hanh\lab5> py .\cau2_server.py
Server on listening on port : 8888
waiting for a connection
connection from ('127.0.0.1', 62828)
sending data back to the client Bốn
no data from ('127.0.0.1', 62828)
waiting for a connection
```

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'

- Code bài làm bên server

```

cau3_server.py > ...
2  import sys
3  def tinh(op, op1, op2):
4      op1 = int(op1)
5      op2 = int(op2)
6      if(op == '+'):
7          return op1 + op2
8      elif(op == '-'):
9          return op1 - op2
10     elif(op == '*'):
11         return op1 * op2
12     elif(op == '/'):
13         return op1 / op2
14     else:
15         return 0;
16     # Create a UDP socket
17     s = socket(AF_INET, SOCK_DGRAM)
18     # Bind the socket to the port
19     server_addr = ('localhost', 8888)
20     print('starting up on ', server_addr)
21     s.bind(server_addr)
22     kq = 0
23     while True:
24         print("Waiting to receive message")
25         data, addr = s.recvfrom(4096)
26         print("da nhan tu client chuỗi", data)
27         op, op1, op2 = data.decode().split(" ")
28         kq = tinh(op, op1, op2)
29         if(int(kq) > 0):
30             sent = s.sendto(str(kq).encode(), addr)
31         else:
32             sent = s.sendto("không thể tính ra kết quả", addr)
33

```

- Code bài làm bên client

```
cau3_client.py > ...
1  from socket import *
2  import sys
3  # Create a UDP socket
4  s = socket(AF_INET, SOCK_DGRAM)
5
6  print("nhập các phép tính, vd: 100 + 200")
7  data = input();
8  op1, op, op2 = data.split(" ")
9  data = ' '.join([op, op1, op2])
10 server_addr = ('localhost', 8888)
11 try:
12     # Send data
13     print('sending "%s"' % data)
14     sent = s.sendto(data.encode(), server_addr)
15     # Receive response
16     print('waiting to receive ...')
17     data, server = s.recvfrom(4096)
18     print('received "%s"' % data.decode())
19 finally:
20     print('closing socket')
21     s.close()
22
```

- Kết quả thực thi bên client

```
PS D:\2019-2023\hk2 2021-2022\CT293\thuc_hanh\lab5> py .\cau3_client.py
nhập các phép tính, vd: 100 + 200
22 - 3333
sending "- 22 3333"
waiting to receive ...
received "-3311"
closing socket
PS D:\2019-2023\hk2 2021-2022\CT293\thuc_hanh\lab5>
```


- Kết quả thực thi bên server

```
TypeError: a bytes-like object is required, not 'str'
PS D:\2019-2023\hk2 2021-2022\CT293\thuc_hanh\lab5> py .\cau3_server.py
starting up on ('localhost', 8888)
Waiting to receive message
da nhan tu client chuỗi b'- 22 3333'
Waiting to receive message
```

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

4. Bài 4

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

Trong đó,

- + Server hoạt động ở 2 cổng 800 và 8001:
 - 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'
- Code bên server


```

33     if not data:
34         fi.close()
35         c2.close()
36         break;
37     finally:
38         c.send("ERROR\n".encode())
39     # delete
40     elif(method == "DELETE"):
41         file_path = './' + name
42         try:
43             os.remove(file_path)
44             c.send("OK\n".encode())
45             print("da xoa file ", name, " thanh cong!!")
46         except OSError as e:
47             c.send("ERROR\n".encode())
48             print("Error: %s : %s" % (file_path, e.strerror))
49     # list
50     elif(method == "LIST"):
51         file_path = './' + name
52         try:
53             list = os.listdir(file_path)
54             c.send("OK\n".encode())
55             c2, addr2 = s2.accept()
56             myString = "-".join(list)
57             c2.send(myString.encode())
58             print("hien ds trong thu muc ", name, " thanh cong!!")
59             c2.close();
60         finally:
61             c.send("ERROR\n".encode())
62     c.close()

```

```
63         list = os.listdir(file_path)
64         s2.bind((host, port2))          # Bind to the port
65         s2.listen(5)
66         c.send("OK\n".encode())
67         c2, addr2 = s2.accept()
68         myString = "-".join(list)
69         c2.send(myString.encode())
70         print("hien ds trong thu muc ", name, " thanh cong!!")
71     finally:
72         c.send("ERROR\n".encode())
73         c2.close()
74         s2.close()
75     else:
76         s.close();
```

- Code bên client

```

1  ✓ import socket                # Import socket module
2  import time
3  s = socket.socket()            # Create a socket object
4  s2 = socket.socket()
5  host = socket.gethostname()    # Get Local machine name
6  port = 8000
7  port2 = 8001                   # Reserve a port for your service.
8  buffer_size = 1024
9  s.connect((host, port))
10 method = input("nhập vào phương thức mà bạn muốn: ")
11 filename = input("Nhập vào tên tập tin: ")
12 # method = "get"
13 # filename = "data1.txt"
14 method = method.upper()
15 dataSend = method + " " + filename
16 print(dataSend)
17 s.send(dataSend.encode() )
18 res = s.recv(buffer_size)
19 # Get
20 ✓ if(method == "GET" and res.decode() == "OK\n" ):
21     print("ok")
22     newfilename = str(time.time()).split('.')[0] + '_' + filename
23     fo = open(newfilename,'wb')
24     s2.connect((host, port2))
25     ✓ while True:
26         print('receiving data...')
27         data = s2.recv(buffer_size)
28         print('data=%s', (data))
29     ✓ if not data:
30         fo.close()
31         print('file close()')
32         break

```

```
33         # write data to a file
34         fo.write(data)
35     fo.close()
36     s.close()
37     s2.close()
38 elif(method == "GET" and res.decode() == "ERROR\n" ):
39     print("File không tồn tại")
40     s.close()
41
42 if(method == "DELETE" and res.decode() == "OK\n"):
43     print("ok đã xóa thành công")
44     s.close()
45 elif( method == "DELETE" and res.decode() == "ERROR\n"):
46     print("File không tồn tại")
47     s.close()
48 if(method == "LIST" and res.decode() == "OK\n"):
49     print("ok")
50     s2.connect((host, port2))
51     data = s2.recv(buffer_size)
52     print(data.decode().split('-'))
53     s2.close()
54     s.close()
55 elif( method == "LIST" and res.decode() == "ERROR\n"):
56     print("File không tồn tại")
57     s2.close()
58     s.close()
59
60
```

-
- Kết quả thực thi bên client khi sử dụng GET data1.txt

```
PS D:\2019-2023\hk2 2021-2022\CT293\thuc_hanh\lab5> py .\Cau4Client.py
nhập vào phương thức mà bạn muốn: get
Nhập vào tên tập tin: data1.txt
GET data1.txt
ok
receiving data...
data=%s b'day l\xc3\xa0 data thu 1'
receiving data...
data=%s b''
file close()
PS D:\2019-2023\hk2 2021-2022\CT293\thuc_hanh\lab5>
```

-
- 1 file mới tên 1651636769_data1.txt đã được tạo
- Kết quả của client khi sử dụng lệnh delete 1651636769_data1.txt

```
PS D:\2019-2023\hk2 2021-2022\CT293\thuc_hanh\lab5> py .\Cau4Client.py
nhập vào phương thức mà bạn muốn: delete
Nhập vào tên tập tin: 1651636769_data1.txt
DELETE 1651636769_data1.txt
ok đã xóa thành công
PS D:\2019-2023\hk2 2021-2022\CT293\thuc_hanh\lab5>
```

-
- 1651636769_data1.txt đã bị xóa
- Kết quả thực hiện bên client khi sử dụng LIST cau4

```
PS D:\2019-2023\hk2 2021-2022\CT293\thuc_hanh\lab5> py .\Cau4Client.py
nhập vào phương thức mà bạn muốn: list
Nhập vào tên tập tin: cau4
LIST cau4
ok
['cau4_1.txt', 'cau4_2.txt']
PS D:\2019-2023\hk2 2021-2022\CT293\thuc_hanh\lab5>
```

-
- Lúc này server sẽ trả về các file con của thư mục client yêu cầu nằm trong thư mục đang hiện hành của server

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

--- Hết ---