COMP2322 Computer Networking

Lab 4 Report: Socket Programming

Author
Wang Yuqi
[redacted]

Lecturer
Dr. LOU Wei

BsC in Computer Science
Academic year: 2024/2025 Sem 2

Questions

Question 1

Q: What is the output when running this python program? Screen capture the executing result.

322_lab_3.typ comp2322_lab_4.typ GoogleClient.py q1.png

lab4 > GoogleClient.py > port

```python
try:
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    print ("socket successfully created")
except socket.error as err:
    print ("socket creation failed with error %s" %(err))

# default port for socket
port = 80

try:
    # find the IP of the Google's web server
    host_ip = socket.gethostbyname('www.google.com')
except socket.gaierror:
    # this means could not resolve the host
    print ("there was an error resolving the host")
    sys.exit()

# connecting to the server
s.connect((host_ip, port))
print ("the socket has successfully connected to google")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

```
(def-311) (base) → labs /opt/homebrew/anaconda3/envs/def-311/bin/python "/Users/tony
socket successfully created
the socket has successfully connected to google
(def-311) (base) → labs
```

Question 2

lab4 > TCPClient.py > ...

```python
# import the socket library
import socket
# create a socket object
clientSocket = socket.socket(socket.AF_INET, socket.SOCK_STR
# define the server's name and port on which you want to con
serverName = '127.0.0.1'
serverPort = 12345
# connect to the server
clientSocket.connect((serverName, serverPort))
# receive data from the server and decode to get the string.
sentence = clientSocket.recv(1024).decode()
print ("from server:", sentence)
# close the connection
clientSocket.close()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS COMMENTS

```
/Users/tonywang/.zprofile:15: no such file or directory: /usr/local/bin/brew
/Users/tonywang/.zshrc:source:120: no such file or directory: /opt/homebrew/opt/powerle
(def-311) (base) → labs python ./lab4/TCPClient.py
from server: thank you for connecting
(def-311) (base) → labs
```

ogleClient.py | TCPServer.py | TCPClient.py | q1.png

lab4 > TCPServer.py > ...

```python
# import the socket library
import socket
# create a socket object
serverSocket = socket.socket(socket.AF_INET, socket.SOCK_STR
print ("socket successfully created")
# reserve a port=12345 on your computer
serverPort = 12345
# bind to the port
# we have not typed any ip in the ip field, instead we have
# this makes the server listen to requests coming from othe
serverSocket.bind(('', serverPort))
print ("socket binded to %s" %(serverPort))
# put the socket into listening mode
serverSocket.listen(5)
print ("socket is listening")

# a forever loop until we interrupt it or an error occurs
while True:
    # establish connection with client.
    connectionSocket, addr = serverSocket.accept()
    print ('got connection from', addr )
    # send a message to the client, using encode() to send b
    sentence='thank you for connecting'
    connectionSocket.send(sentence.encode())
    # close the connection with the client
    connectionSocket.close()
    break
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS COMMENTS

```
(def-311) (base) → labs python ./lab4/TCPServer.py
socket successfully created
socket binded to 12345
socket is listening
got connection from ('127.0.0.1', 60994)
(def-311) (base) → labs
```

(See Next Page)

Question 3

TCPServer.py | q3-server.png | TCPClient.py | _socket.py

lab4 > TCPClient.py > ...

```python
# create a socket object
clientSocket = socket.socket(socket.AF_INET, socket.SOCK_STR
clientSocket.bind(('', 40134))
# define the server's name and port on which you want to co
serverName = '127.0.0.1'
serverPort = 12345
# connect to the server
clientSocket.connect((serverName, serverPort))

srcIP, srcPort = clientSocket.getsockname()
destIP, destPort = clientSocket.getpeername()

print(f'Tuple: (SrcIP: {srcIP}, SrcPort: {srcPort}, DestIP:

# receive data from the server and decode to get the string.
sentence = clientSocket.recv(1024).decode()
print ("from server:", sentence)
# close the connection
clientSocket.close()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS COMMENTS

```
(def-311) (base) → labs python ./lab4/TCPClient.py
Tuple: (SrcIP: 127.0.0.1, SrcPort: 40134, DestIP: 127.0.0.1, DestPort: 12345)
from server: thank you for connecting
(def-311) (base) → labs
```

TCPServer.py × TCPClient.py _socket.pyi q2-client.png

lab4 > TCPServer.py > ...

~/Code/prac/polyu_comp_

```python
import socket
# create a socket object
serverSocket = socket.socket(socket.AF_INET, socket.SOCK_STR
print ("socket successfully created")
# reserve a port=12345 on your computer
serverPort = 12345
# bind to the port
# we have not typed any ip in the ip field, instead we have
# this makes the server listen to requests coming from other
serverSocket.bind(('', serverPort))
print ("socket binded to %s" %(serverPort))
# put the socket into listening mode
serverSocket.listen(5)
print ("socket is listening")

# a forever loop until we interrupt it or an error occurs
while True:
    # establish connection with client.
    connectionSocket, addr = serverSocket.accept()
    print ('got connection from', addr )
    # send a message to the client, using encode() to send b
    sentence='thank you for connecting'
    connectionSocket.send(sentence.encode())
    # close the connection with the client
    connectionSocket.close()
    break
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS COMMENTS

```
(def-311) (base) → labs python ./lab4/TCPServer.py
socket successfully created
socket binded to 12345
socket is listening
got connection from ('127.0.0.1', 40134)
(def-311) (base) → labs
```

(See Next Page)

Modified Code for `TCPCLient.py`

```python
# import the socket library
import socket
# create a socket object
clientSocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
clientSocket.bind(('', 40134))
# define the server's name and port on which you want to connect
serverName = '127.0.0.1'
serverPort = 12345
# connect to the server
clientSocket.connect((serverName, serverPort))

srcIP, srcPort = clientSocket.getsockname()
destIP, destPort = clientSocket.getpeername()

print(f'Tuple: (SrcIP: {srcIP}, SrcPort: {srcPort}, DestIP: {destIP},
DestPort: {destPort})')

# receive data from the server and decode to get the string.
sentence = clientSocket.recv(1024).decode()
print ("from server:", sentence)
# close the connection

clientSocket.close()
```

Question 4

TCPServer.py × TCPClient.py q3-server.png comp2322_lab_4.typ q3-

lab4 > TCPServer.py > ...

```python
serverSocket.listen(5)
print ("socket is listening")

# a forever loop until we interrupt it or an error occurs
while True:
    # establish connection with client.
    connectionSocket, addr = serverSocket.accept()
    print ('got connection from', addr)
    # send a message to the client, using encode() to send byte type
    sentence='thank you for connecting'
    connectionSocket.send(sentence.encode())
    # close the connection with the client
    passwd = connectionSocket.recv(1024).decode()
    connectionSocket.send(("Your password is correct!" if passwd == '0134'

    connectionSocket.close()
    break
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS COMMENTS

```
(def-311) (base) → labs python ./lab4/TCPServer.py
socket successfully created
socket binded to 12345
socket is listening
got connection from ('127.0.0.1', 40134)
(def-311) (base) → labs python ./lab4/TCPServer.py
socket successfully created
socket binded to 12345
socket is listening
got connection from ('127.0.0.1', 40134)
(def-311) (base) → labs
```

TCPServer.py | q4-server.png | TCPClient.py | q3-server.png | comp232

lab4 > TCPClient.py > ...

```python
import socket
# create a socket object
clientSocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
clientSocket.bind(('', 40134))
# define the server's name and port on which you want to connect
serverName = '127.0.0.1'
serverPort = 12345
# connect to the server
clientSocket.connect((serverName, serverPort))

srcIP, srcPort = clientSocket.getsockname()
destIP, destPort = clientSocket.getpeername()

print(f'Tuple: (SrcIP: {srcIP}, SrcPort: {srcPort}, DestIP: {destIP}, DestP

# receive data from the server and decode to get the string.
sentence = clientSocket.recv(1024).decode()
print ("from server:", sentence)
# close the connection

passwd = input('Input 4 digit password: ')
clientSocket.send(passwd.encode())

auth_result = clientSocket.recv(1034).decode()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS COMMENTS

```
(def-311) (base) → labs python ./lab4/TCPClient.py
Tuple: (SrcIP: 127.0.0.1, SrcPort: 40134, DestIP: 127.0.0.1, DestPort: 12345)
from server: thank you for connecting
Input 4 digit password: 0134
from server:  Your password is correct!
(def-311) (base) → labs python ./lab4/TCPClient.py
Tuple: (SrcIP: 127.0.0.1, SrcPort: 40134, DestIP: 127.0.0.1, DestPort: 12345)
from server: thank you for connecting
Input 4 digit password: 1441
from server:  Your password is incorrect!
(def-311) (base) → labs
```

(See Next Page)

Modified Code for `TCPServer.py`

```
# import the socket library
import socket
# create a socket object
serverSocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
serverSocket.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)

print ("socket successfully created")
# reserve a port=12345 on your computer
serverPort = 12345
# bind to the port
# we have not typed any ip in the ip field, instead we have inputted an
empty string.
# this makes the server listen to requests coming from other computers
on the network
serverSocket.bind(('', serverPort))
print ("socket binded to %s" %(serverPort))
# put the socket into listening mode
serverSocket.listen(5)
print ("socket is listening")

# a forever loop until we interrupt it or an error occurs
while True:
    # establish connection with client.
    connectionSocket, addr = serverSocket.accept()
    print ('got connection from', addr)
    # send a message to the client, using encode() to send byte type
    sentence='thank you for connecting'
    connectionSocket.send(sentence.encode())
    # close the connection with the client
    passwd = connectionSocket.recv(1024).decode()
    connectionSocket.send(("Your password is correct!" if passwd ==
'0134' else "Your password is incorrect!").encode())

    connectionSocket.close()
    break
```

(See Next Page)

Modified Code for `TCPClient.py`

```
# import the socket library
import socket
# create a socket object
clientSocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
clientSocket.bind(('', 40134))
# define the server's name and port on which you want to connect
serverName = '127.0.0.1'
serverPort = 12345
# connect to the server
clientSocket.connect((serverName, serverPort))

srcIP, srcPort = clientSocket.getsockname()
destIP, destPort = clientSocket.getpeername()

print(f'Tuple: (SrcIP: {srcIP}, SrcPort: {srcPort}, DestIP: {destIP},
DestPort: {destPort})')

# receive data from the server and decode to get the string.
sentence = clientSocket.recv(1024).decode()
print ("from server:", sentence)
# close the connection

passwd = input('Input 4 digit password: ')
clientSocket.send(passwd.encode())

auth_result = clientSocket.recv(1034).decode()
print("from server: ", auth_result)

clientSocket.close()
```