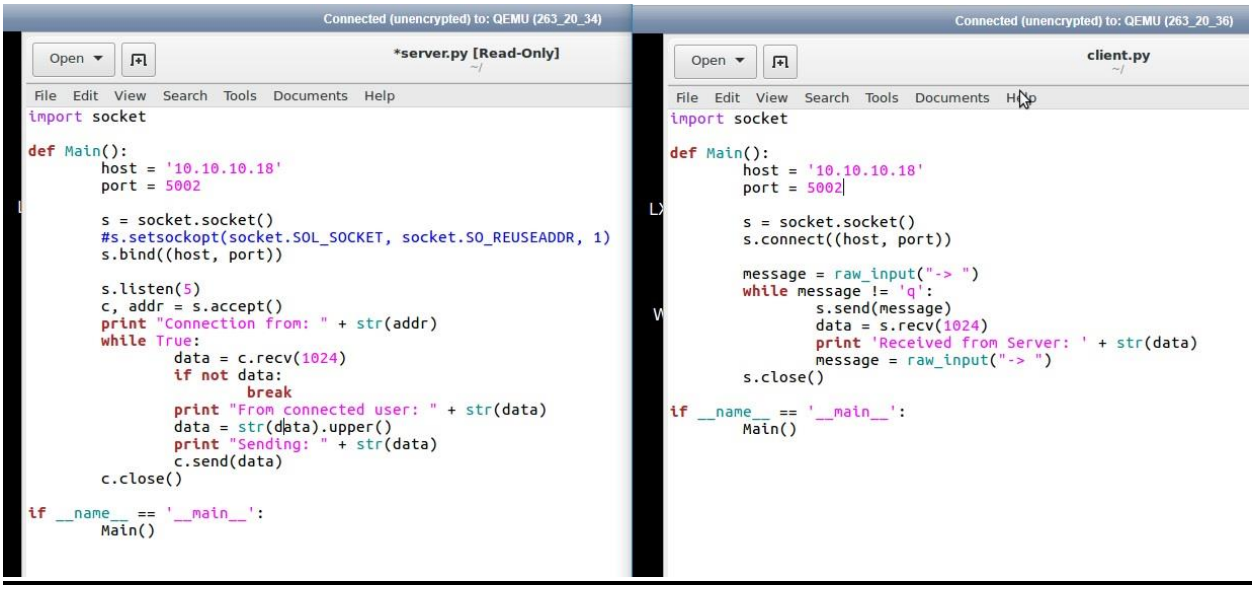
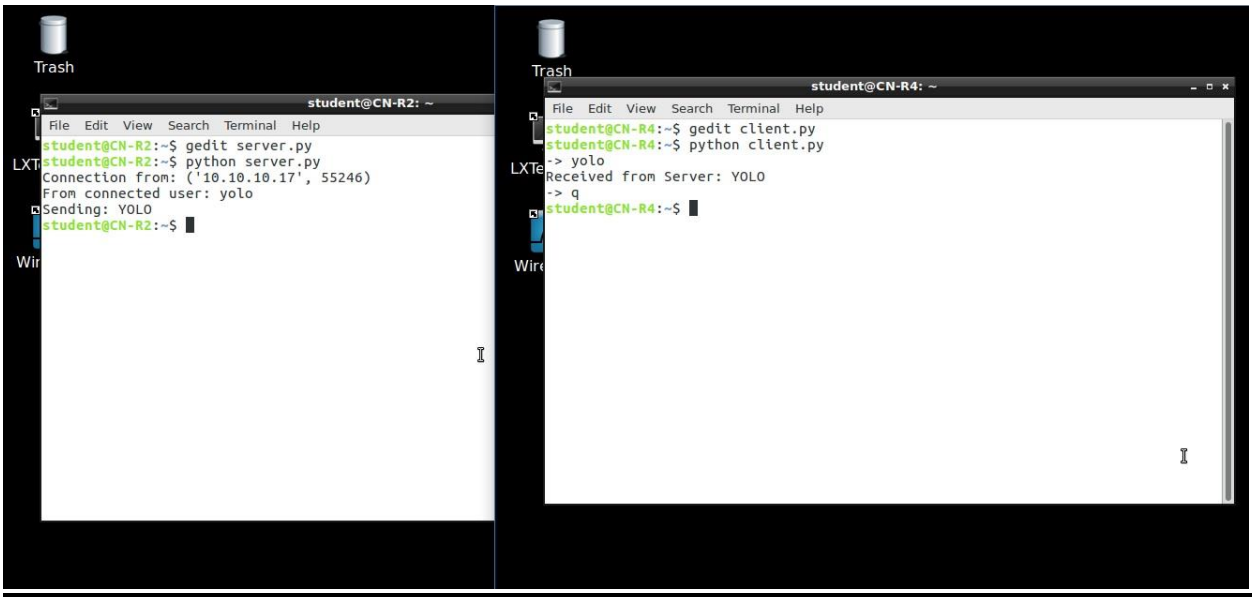
SOCKET PROGRAMMING USING PYTHON

Ajay Shete (abs717)

N19633252

Connections





Write an Echo Server/Client code using socketserver framework
Echo Server:
a) should receive a string from Client
b) If the string contains secret code “SECRET”, then server should return all the digits in the string and count of digits
c) close the connection with client
Echo Client:
a) Should send a string to the server
b) Should receive the output Example: If client sends “socket 123345nfjkw345 SECRET “, then server should reply Digits: 123345345 Count: 9

```
import SocketServer

class MyTCPHandler(SocketServer.BaseRequestHandler):

        def handle(self):
                print "Connection from: " + str(self.client_address[0])

                while True:
                        self.data = self.request.recv(1024)

                        if not self.data:
                                break

                        print "from connected user: " + str(self.data)

                        d1 = str(self.data)
                        d2 = ''
                        count = 0

                        if "SECRET" in d1:

                                for i in range (0,len(d1)):
                                        if d1[i].isdigit():
                                                d2 = d2 + d1[i]
                                                count = count + 1
                        else:
                                d2 = '0'

                        d3 = "Digits: " + d2 + "\nCount: " + str(count)
                        print "Sending: " + d3
                        self.request.sendall
(d3)

if __name__ == '__main__':
        host, port = "10.10.10.18", 5011
        server  = SocketServer.TCPServer((host, port), MyTCPHandler)
        server.serve_forever()
```

Server Side Code

Open

clientd.py
/home/student

File Edit View Search Tools Documents Help

```
import socket
import sys

host, port = "10.10.10.18", 5011

sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

try:
        sock.connect((host,port))

        message = raw_input("\nEnter the String: ")
        while message != 'q':
                sock.sendall(message)
                data = sock.recv(1024)
                print 'Received from server: ' + str(data)
                message = raw_input("\nEnter the String: ")
finally:
        sock.close()
```

Client-side Code

```
root@CN-R2:/home/student# python serverd2.py
Connection from: 10.10.10.17
from connected user: 123qwe987
Sending: Digits: 0
Count: 0
from connected user: 1234qwe89SECRET
Sending: Digits: 123489
Count: 6
```

Server-Output

```
root@CN-R4:/home/student# python clientd.py

Enter the String: 123qwe987
Received from server: Digits: 0
Count: 0

Enter the String: 1234qwe89SECRET
Received from server: Digits: 123489
Count: 6

Enter the String: q
```

Client Output

File Transfer over network using sockets.

Server:

a) should receive data from Client (create a text file)

b) write the received data in a file

c) close the connection

Client:

a) Connect with the server

b) Send the file

c) close the connection

Open | serverf.py /home/student

File Edit View Search Tools Documents Help

```
import socket

def Main():
        host = '10.10.10.18'
        port = 6045

        s = socket.socket()
        s.bind((host,port))
        s.listen(2)

        print 'Server listening....'

        while True:
                conn, addr = s.accept()
                print 'Got connection from ', addr
                data = conn.recv(1024)
                #print('Server received', repr(data))

                with open('received_file', 'wb') as f:
                        print 'file opened'
                        while True:
                                print('receiving data...')
                                f = open(data,'w')
                                data = conn.recv(1024)
                                #print(data = '%s', (data))
                                if not data:
                                        break
                                f.write(data)
                        f.close()
                print('Successfully received file')

        conn.send('Thank you for connecting')
        conn.close()

if __name__ == '__main__':
        Main()
```

Server-side Code

Open | clientf.py /home/student | Save

File Edit View Search Tools Documents Help

```
import socket

def Main():
        s = socket.socket()
        host = '10.10.10.18'
        port = 6045

        s.connect((host, port))
        #s.send("Hello Server!")

        """filename = '2.txt'   f = open(filename, 'rb')        l = f.read(1024)
        while(l):
                s.send(l)       print('Sent', repr(l))          l = f.read(1024)
        f.close()       print('Done Sending')"""

        print '\nEnter the file name:\n'
        message = raw_input("->")
        while message != 'q':
                s.send(message)
                afile = open(message, 'r')
                astring = afile.read()

                afile.close()

                s.send(astring)
                data = s.recv(1024)
                print 'Received from server:' + str(data)
                print '\nEnter the file name:\n'
                message = raw_input("->")

        s.close()
        print('Connection closed!')

if __name__ == '__main__':
        Main()
```

Client-side Code

```
root@CN-R2:/home/student# python serverf.py
Server listening....
Got connection from  ('10.10.10.17', 57992)
file opened
receiving data...
receiving data...
Successfully received file
```

Server (1)

```
root@CN-R4:/home/student# python clientf.py

Enter the file name:

->/home/student/2.txt
```

Client (1)

```
root@CN-R4:/home/student# cat 2.txt
qwertyuiopasdfghjklzxcvbnm
root@CN-R4:/home/student#
```

Client (2)

```
root@CN-R2:/home/student# cat 2.txt
qwertyuiopasdfghjklzxcvbnm
root@CN-R2:/home/student#
```

Server (2)