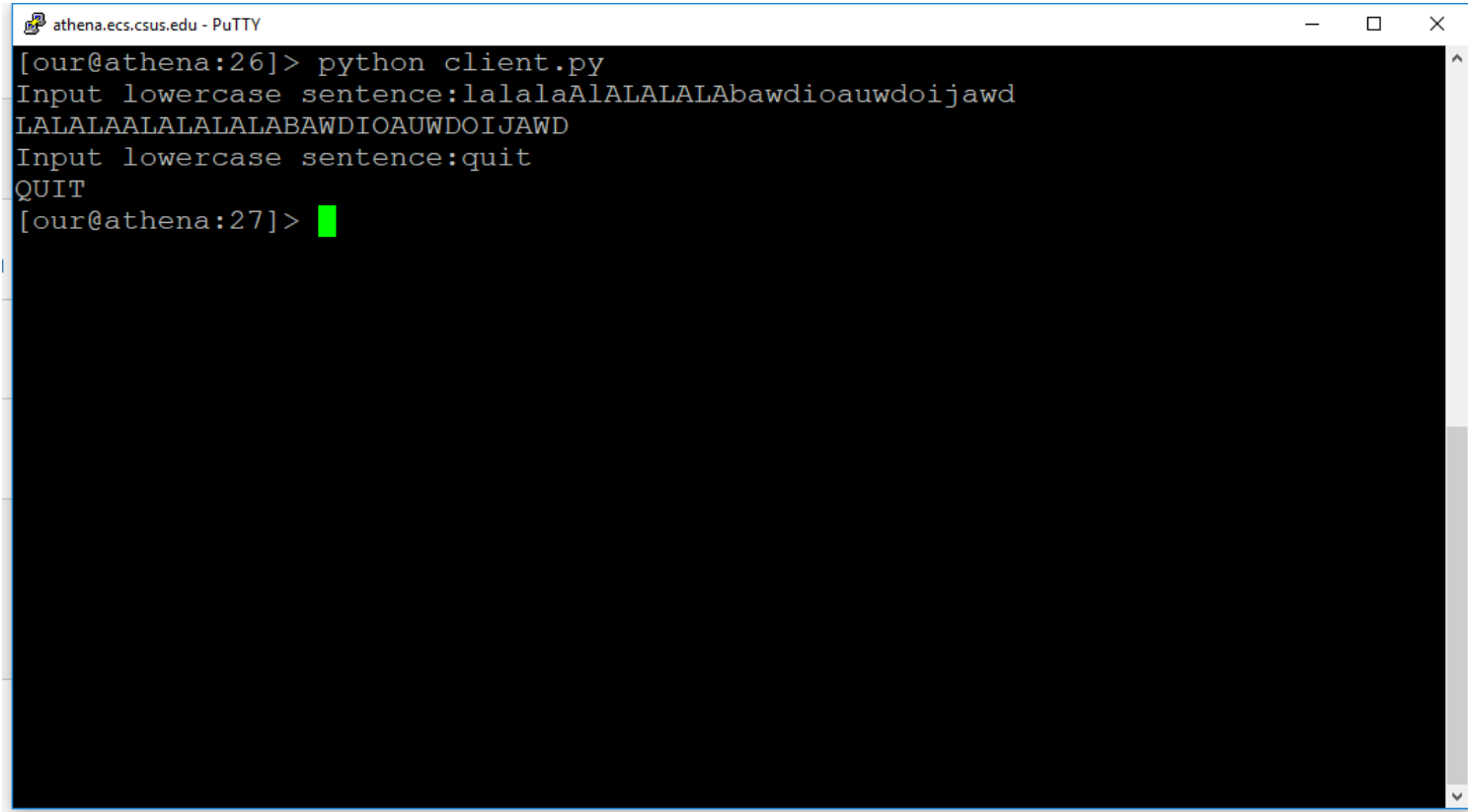


SOCKET PROGRAMMING 1 - RONGGUANG OU

Wednesday, February 20, 2019 8:06 PM

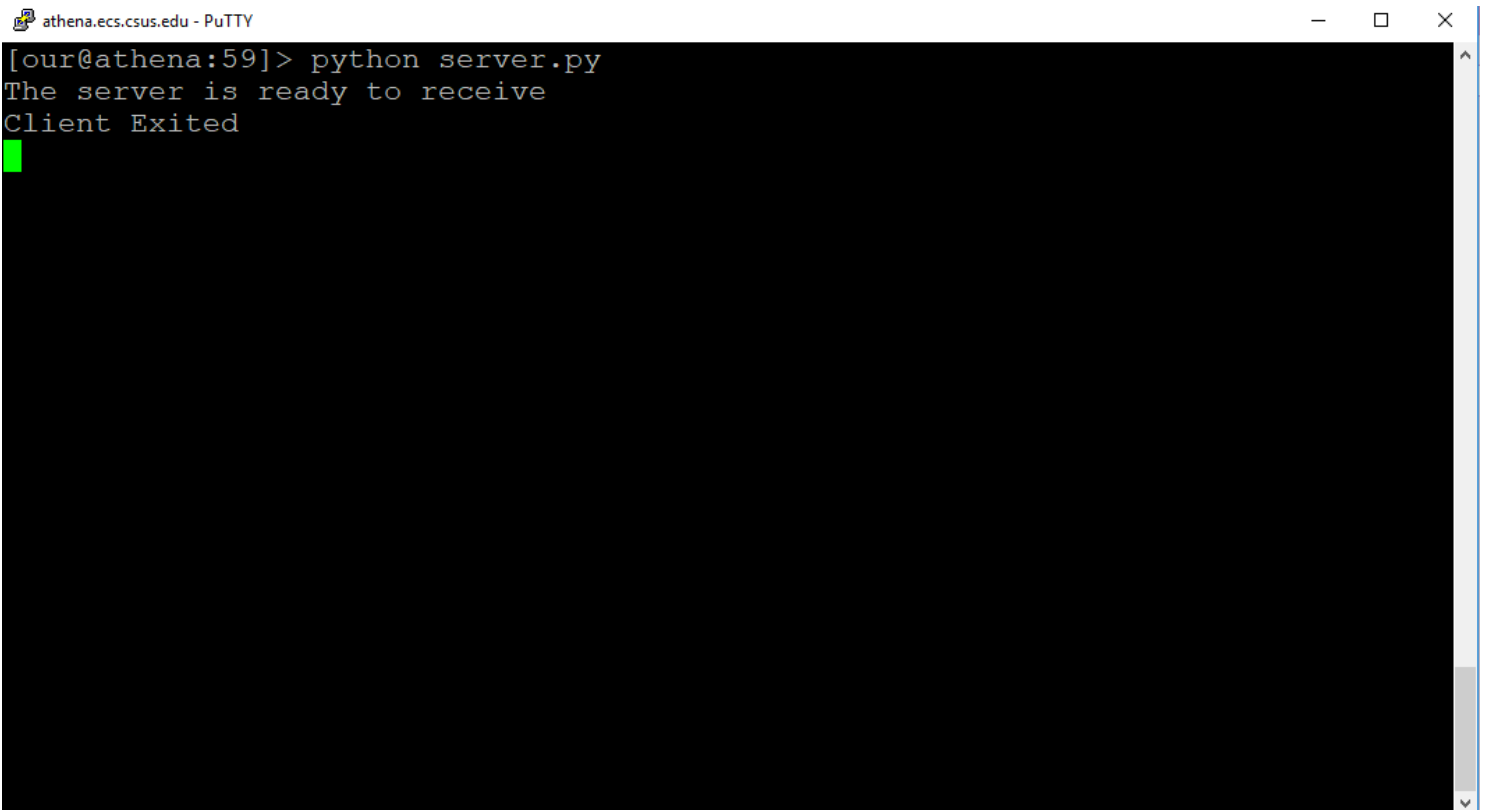
UDP Client SIDE



The screenshot shows a PuTTY terminal window titled 'athena.ecs.csus.edu - PuTTY'. The terminal output is as follows:

```
[our@athena:26]> python client.py
Input lowercase sentence:lalalaAlALALALAbawdioauwdoijawd
LALALAALALALALABAWDIOAUWDOIJAWD
Input lowercase sentence:quit
QUIT
[our@athena:27]> █
```

UDP SERVER SIDE



The screenshot shows a PuTTY terminal window titled 'athena.ecs.csus.edu - PuTTY'. The terminal output is as follows:

```
[our@athena:59]> python server.py
The server is ready to receive
Client Exited
█
```

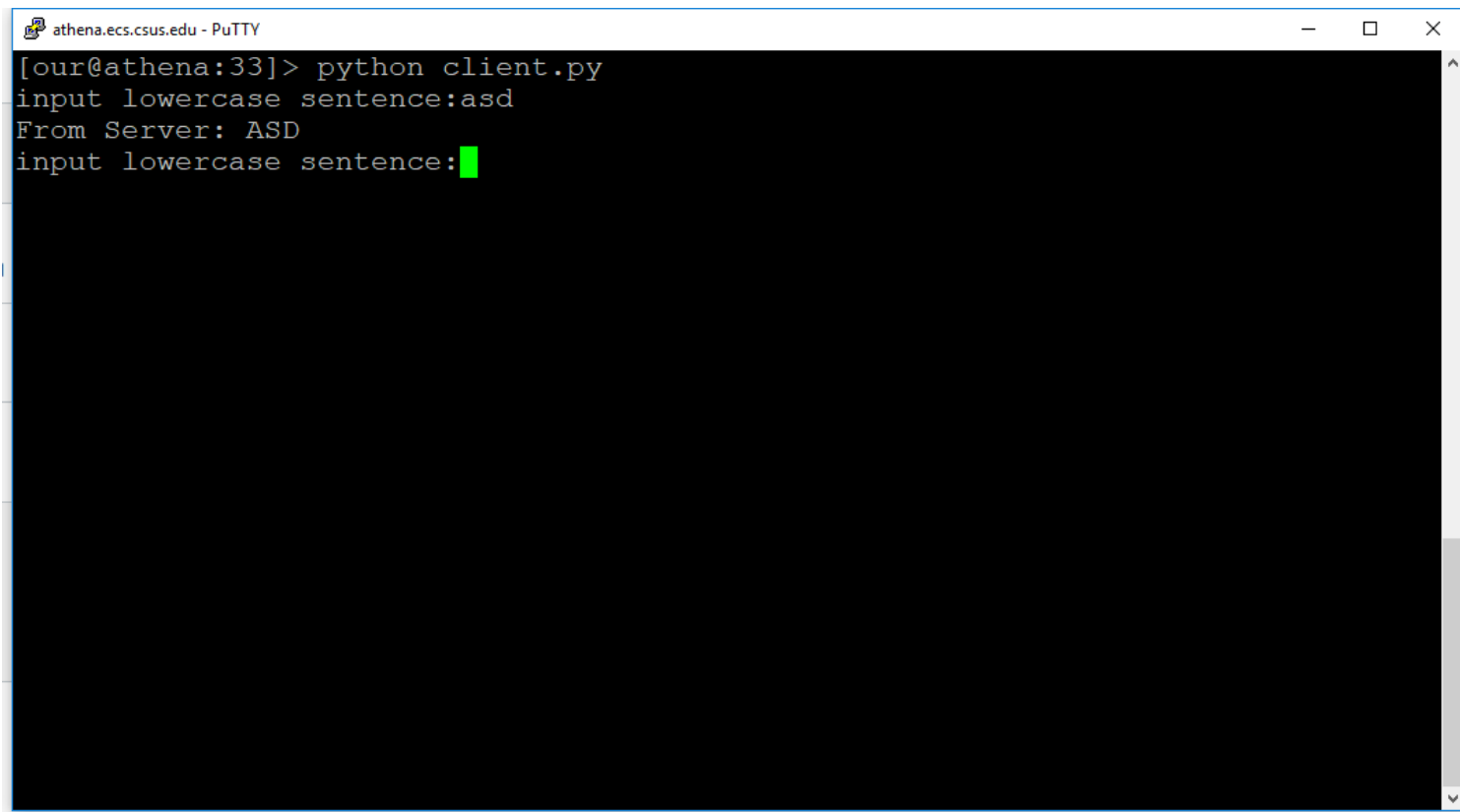
UDP Client Code

```
client.py
1  from socket import*
2
3  #server setup
4  serverName = '127.0.0.1'
5  serverPort = 1603
6
7  #clientSocket object
8  clientSocket = socket(AF_INET,SOCK_DGRAM)
9
10 while True:
11     message = raw_input('Input lowercase sentence:')
12     clientSocket.sendto(message.encode(),(serverName,serverPort))
13     modifiedMessage,serverAddress = clientSocket.recvfrom(2048)
14     print modifiedMessage.decode()
15     if message == 'quit':
16         clientSocket.close()
17         break;
18
19 #clientSocket.close()
20
21
```

UDP Server Code

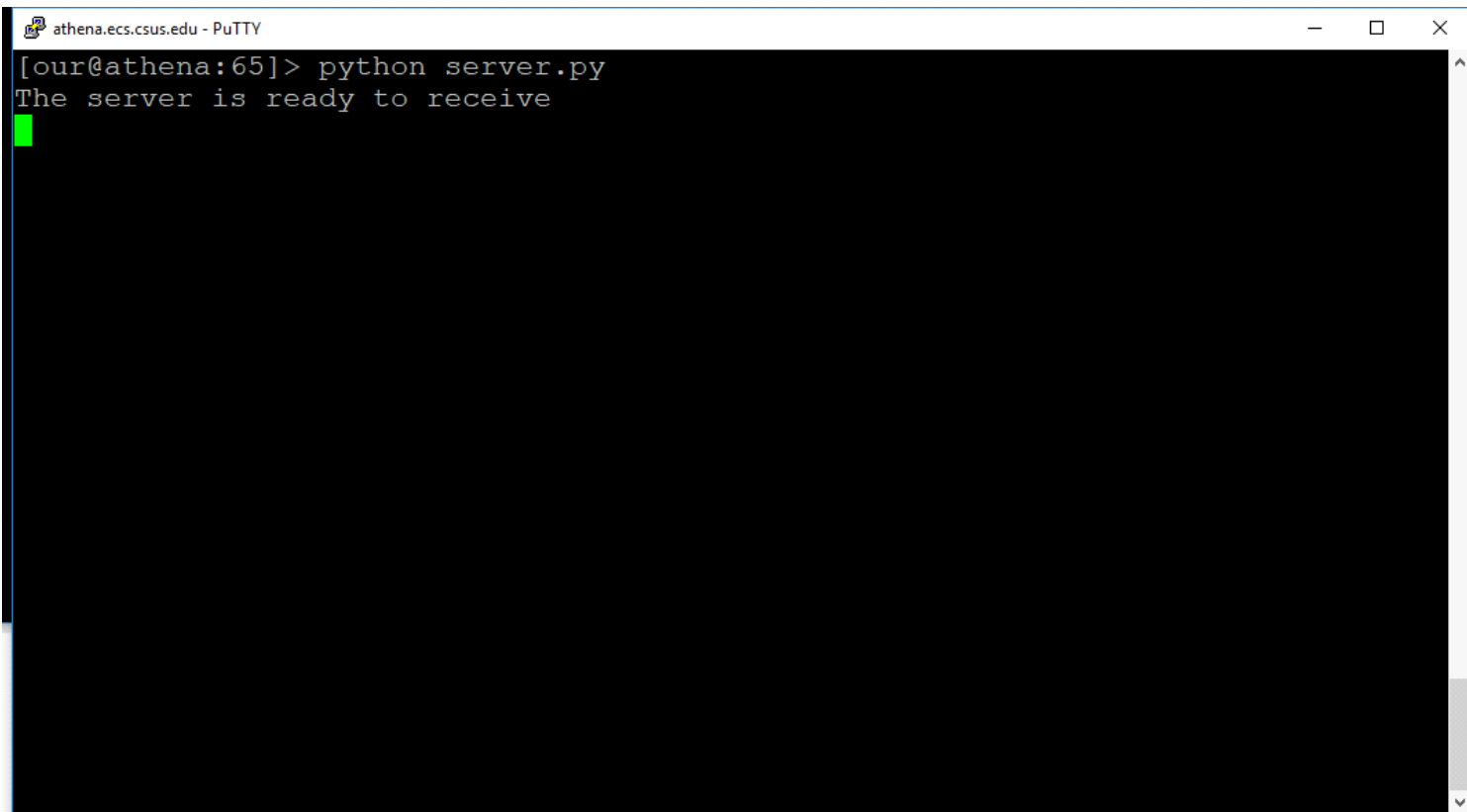
```
server.py
1  #UDP SERVER CODE
2  from socket import*
3
4
5  serverPort = 1603
6  serverSocket = socket(AF_INET,SOCK_DGRAM)
7  server_address = ('127.0.0.1' , 1603)
8  serverSocket.bind(server_address)
9  print("The server is ready to receive")
10 run = True
11 while run:
12     message,clientAddress = serverSocket.recvfrom(2048)
13     #print("Caught incoming message")
14     modifiedMessage = message.decode().upper()
15     serverSocket.sendto(modifiedMessage.encode(),clientAddress)
16     if(message == 'quit'):
17         print("Client Exited")
18     if(message == 'serverquit'):
19         serverSocket.close()
20         print("Server Terminated by Client")
21         run = False
22
```

TCP Client Side

A terminal window titled 'athena.ecs.csus.edu - PuTTY' with a black background and white text. The text shows a user running 'python client.py', entering 'asd' for a lowercase sentence, receiving 'ASD' from the server, and then being prompted for another lowercase sentence.

```
athena.ecs.csus.edu - PuTTY
[our@athena:33]> python client.py
input lowercase sentence:asd
From Server: ASD
input lowercase sentence:
```

TCP Server Side

A terminal window titled 'athena.ecs.csus.edu - PuTTY' with a black background and white text. The text shows a user running 'python server.py', which outputs 'The server is ready to receive', followed by a green cursor.

```
athena.ecs.csus.edu - PuTTY
[our@athena:65]> python server.py
The server is ready to receive
█
```

TCP Client Code

```
client.py
1  #TCP CLIENT CODE
2  from socket import*
3  server_address = ('127.0.0.1',1603)
4  clientSocket = socket(AF_INET,SOCK_STREAM)
5  clientSocket.connect(server_address)
6  while True:
7      sentence = raw_input('input lowercase sentence:')
8      clientSocket.send(sentence.encode())
9      modifiedSentence = clientSocket.recv(1024)
10     print('From Server: ' + modifiedSentence.decode())
11     clientSocket.close()
```

TCP Server Code

```
server.py
1  #TCP SERVER CODE
2  from socket import*
3  server_address = ('127.0.0.1' , 1603)
4  serverSocket = socket(AF_INET,SOCK_STREAM)
5  serverSocket.bind(server_address)
6  serverSocket.listen(1)
7  while True:
8      print 'The server is ready to receive'
9      connectionSocket, addr = serverSocket.accept()
10     sentence = connectionSocket.recv(1024).decode()
11     capitalizedSentence = sentence.upper()
12     connectionSocket.send(capitalizedSentence.encode())
13     connectionSocket.close()
14
```