Tuesday, 24 April 2018

Nicola D'Aniello

Coursework for computer networking

this document describe the functioning of the client.py and server.py included with it.

The Server

The server.py is a program that acts as a UDP server. It listens to port 5454 on the localhost until he receives a total of 3 UDP packets. Those packets are assumed to contain short strings.

Once the three packets are received, one reply packet is sent back to the client containing the concatenation of the 3 messages previously received, with an hash symbol (#) between them.

A second reply packet is sent containing the sum of the length of the three packets previously received.

At the end the server clear the received messages and restart the process.

The Client

The client.py is a program that acts as a UDP client. It creates a UDP socket, and send a total of 3 packets (containing string entered by the user) to the server (localhost) on port 5454.

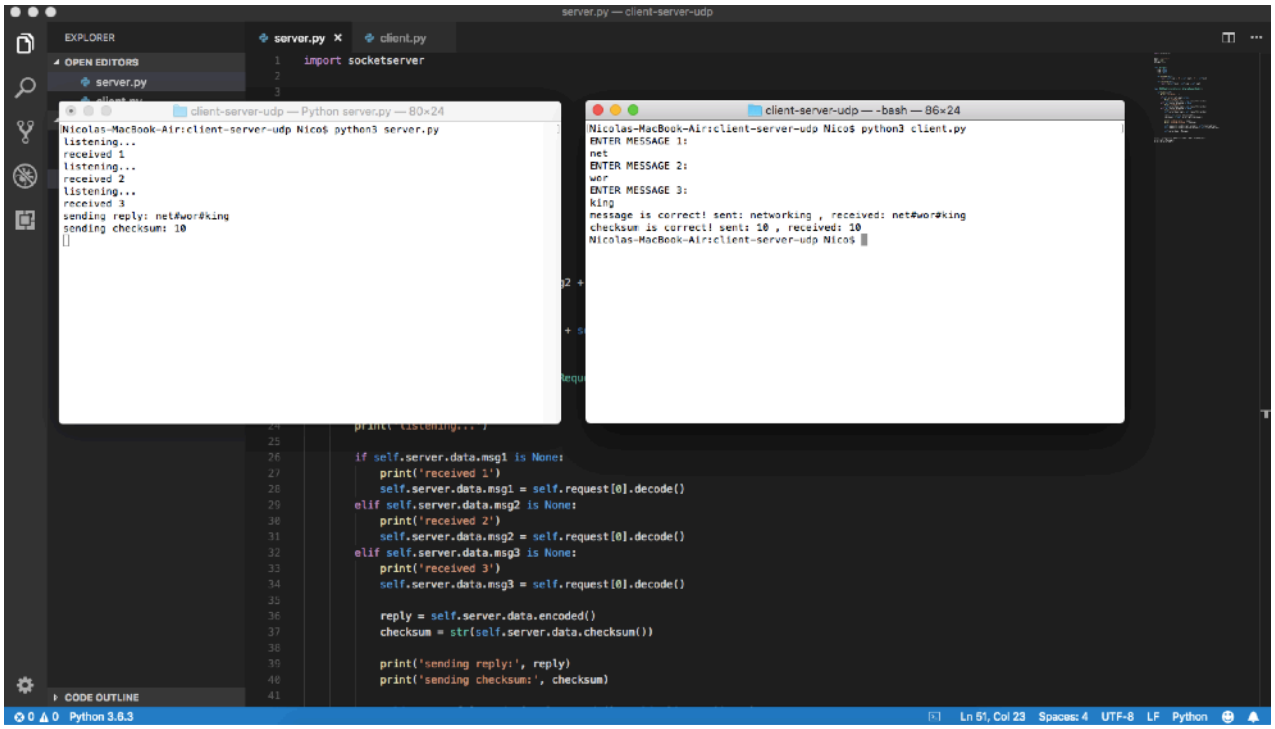
Once the messages are sent it awaits for a reply packet that is assumed contains a message with the concatenation of the 3 previously sent messages, separated by an hash symbol (#).

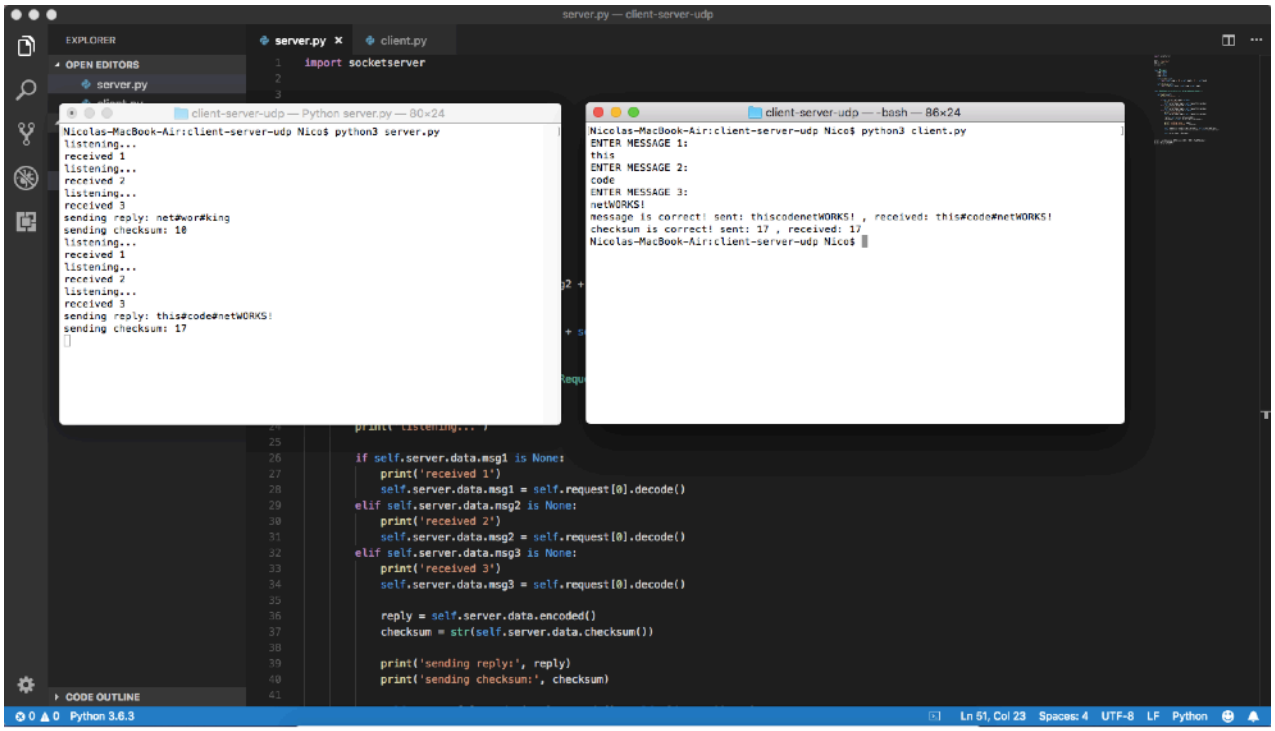
After receiving the first packet it awaits for a second packet which is assumed contains the sum of the length of the three previously sent messages.

Once the second packet is received the client check wether the received message is equal to the concatenation of the sent messages and displays a string with the answer.

It also check wether the checksum received is the same as the checksum calculated and displays another message with the answer.

After that the client terminates.


server.py — client-server-udp
EXPLORER
server.py
client.py
OPEN EDITORS
server.py
import socketserver
client-server-udp — Python server.py — 80×24
Nicolas-MacBook-Air:client-server-udp Nico$ python3 server.py
listening...
received 1
listening...
received 2
listening...
received 3
sending reply: net#wor#king
sending checksum: 10
client-server-udp — -bash — 86×24
Nicolas-MacBook-Air:client-server-udp Nico$ python3 client.py
ENTER MESSAGE 1:
net
ENTER MESSAGE 2:
wor
ENTER MESSAGE 3:
king
message is correct! sent: networking , received: net#wor#king
checksum is correct! sent: 10 , received: 10
Nicolas-MacBook-Air:client-server-udp Nico$
if self.server.data.msg1 is None:
print('received 1')
self.server.data.msg1 = self.request[0].decode()
elif self.server.data.msg2 is None:
print('received 2')
self.server.data.msg2 = self.request[0].decode()
elif self.server.data.msg3 is None:
print('received 3')
self.server.data.msg3 = self.request[0].decode()
reply = self.server.data.encoded()
checksum = str(self.server.data.checksum())
print('sending reply:', reply)
print('sending checksum:', checksum)
CODE OUTLINE
Python 3.6.3
Ln 51, Col 23
Spaces: 4
UTF-8
LF
Python


server.py — client-server-udp
EXPLORER
server.py
client.py
OPEN EDITORS
server.py
import socketserver
client-server-udp — Python server.py — 80×24
Nicolas-MacBook-Air:client-server-udp Nico$ python3 server.py
listening...
received 1
listening...
received 2
listening...
received 3
sending reply: net#wor#king
sending checksum: 10
listening...
received 1
listening...
received 2
listening...
received 3
sending reply: this#code#netWORKS!
sending checksum: 17
client-server-udp — -bash — 86×24
Nicolas-MacBook-Air:client-server-udp Nico$ python3 client.py
ENTER MESSAGE 1:
this
ENTER MESSAGE 2:
code
ENTER MESSAGE 3:
netWORKS!
message is correct! sent: thiscodenetWORKS! , received: this#code#netWORKS!
checksum is correct! sent: 17 , received: 17
Nicolas-MacBook-Air:client-server-udp Nico$
if self.server.data.msg1 is None:
print('received 1')
self.server.data.msg1 = self.request[0].decode()
elif self.server.data.msg2 is None:
print('received 2')
self.server.data.msg2 = self.request[0].decode()
elif self.server.data.msg3 is None:
print('received 3')
self.server.data.msg3 = self.request[0].decode()
reply = self.server.data.encoded()
checksum = str(self.server.data.checksum())
print('sending reply:', reply)
print('sending checksum:', checksum)
CODE OUTLINE
Python 3.6.3
Ln 51, Col 23
Spaces: 4
UTF-8
LF
Python