Atividade sobre SMTP

Redes de Computadores

Sabrina Araújo Cardoso - 118210114

Lab. 3: SMTP Lab

- Código para enviar uma mensagem via SMTP:

```
from socket import *

msg = "\r\n I love computer networks!"
endmsg = "\r\n.\r\n"

# Choose a mail server (e.g. Google mail server) and call it mailserver
mailserver = "smtp.gmail.com"
serverPort = 587

# Create socket called clientSocket and establish a TCP connection with mailserver
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((mailserver, serverPort))

recv = clientSocket.recv(1024).decode()
print(recv)
if recv[:3] != '220':
    print('220 reply not received from server.')
# Send HELO command and print server response.
heloCommand = 'HELO Alice\r\n'
clientSocket.send(heloCommand.encode())
recv1 = clientSocket.recv(1024).decode()
print(recv1)
if recv1[:3] != '250':
    print('250 reply not received from server.')

# Send MAIL FROM command and print server response.
mailFrom = "MAIL FROM <araujocardososabrina@gmail.br>\r\n"
clientSocket.send(mailFrom.encode())
recv3 = clientSocket.recv(1024).decode()
print(recv3)
if recv3[:3] != '250':
    print('250 reply not received from server.')

# Send RCPT TO command and print server response.
rcptTo = "RCPT TO <sabrina.cardoso@ee.ufcg.edu.br>\r\n"
clientSocket.send(rcptTo.encode())
recv4 = clientSocket.recv(1024).decode()
print(recv4)
if recv4[:3] != '250':
    print('250 reply not received from server.')

# Send DATA command and print server response.
data = "DATA"
clientSocket.send(data.encode())
recv5 = clientSocket.recv(1024).decode()
print(recv5)
if recv5[:3] != '250':
    print('250 reply not received from server.')

# Send message data.
message = "hello from the Computer Networks discipline. \r\n"
# Message ends with a single period.
clientSocket.send(message.encode())
recv6 = clientSocket.recv(1024).decode()
print(recv6)
if recv6[:3] != '250':
    print('250 reply not received from server.')

# Send QUIT command and get server response.
quit = "QUIT \t\n"
clientSocket.send(quit.encode())
recv7 = clientSocket.recv(1024).decode()
print(recv7)
if recv7[:3] != '250':
    print('250 reply not received from server.')
clientSocket.close()
```

Questão 1 e 2

- Código dos exercícios opcionais:

```
from socket import *
import ssl
import base64

msg = "\r\n I love computer networks!"
endmsg = "\r\n.\r\n"

# Using the Google mail server
mailserver = "smtp.gmail.com"
port = 587

# Create socket called clientSocket and establish a TCP connection with mailserver
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((mailserver, port))
recv = clientSocket.recv(1024).decode()
print(recv)
if recv[:3] != '220':
    print('220 reply not received from server.')

# Send HELO command and print server response.
heloCommand = 'HELO Alice\r\n'
```

```
clientSocket.send(heloCommand.encode())
recv1 = clientSocket.recv(1024).decode()
print(recv1)
if recv1[:3] != '250':
  print('250 reply not received from server.')

# Send STARTTLS command
tlsCommand = 'STARTTLS\r\n'
clientSocket.send(tlsCommand.encode())
tlsRecv = clientSocket.recv(1024).decode()
print(tlsRecv)
if tlsRecv[:3] != '220':
  print('220 reply not received from server.')

# Add SSL after the STARTTLS command
clientSocket = ssl.wrap_socket(clientSocket)

#Send HELO again for handshake after STARTTLS
clientSocket.send(heloCommand.encode())
recv1 = clientSocket.recv(1024).decode()
print(recv1)
if recv1[:3] != '250':
  print('250 reply not received from server.')

# Send AUTH login command
authCommand = 'AUTH LOGIN\r\n'
clientSocket.write(authCommand.encode())
authRecv = clientSocket.recv(1024).decode()
print(authRecv)
if authRecv[:3] != '334':
  print('334 reply not received from server.')

# Send Username in base64 (Note: sender must enable access from 'less secure apps')
username = "araujocardososabrina@gmail.com"# put in the sender's username here
encodedUsername = base64.b64encode(username.encode())
stringUsername = str(encodedUsername, "utf-8") + "\r\n"
clientSocket.write(stringUsername.encode())
userRecv = clientSocket.recv(1024).decode()
print(userRecv)
if userRecv[:3] != '334':
  print('334 reply not received from server.')

# Send password in base64
password = "password" # put in the sender's password here
encodedPassword = base64.b64encode(password.encode())
stringPassword = str(encodedPassword, "utf-8") + "\r\n"
clientSocket.write(stringPassword.encode())
passRecv = clientSocket.recv(1024).decode()
print(passRecv)
if passRecv[:3] != '235':
  print('235 reply not received from server.')

# Send MAIL FROM command and print server response.
sender = "<araujocardososabrina@gmail.com>"# put in the sender's username, in '< >' he
re
mailFromCommand = 'MAIL FROM: ' + sender + '\r\n'
clientSocket.send(mailFromCommand.encode())
recv2 = clientSocket.recv(1024).decode()
```

```
print(recv2)
if recv2[:3] != '250':
  print('250 reply not received from server.')

# Send RCPT TO command and print server response.
recipient = "sabrina.cardoso@ee.ufcg.edu.br"# put in the recipient's username, in '<
 >' here
rcptToCommand = 'RCPT TO: ' + recipient + '\r\n'
clientSocket.send(rcptToCommand.encode())
recv3 = clientSocket.recv(1024).decode()
print(recv3)
if recv3[:3] != '250':
  print('250 reply not received from server.')

# Send DATA command and print server response.
dataCommand = 'DATA\r\n'
clientSocket.send(dataCommand.encode())
recv4 = clientSocket.recv(1024).decode()
print(recv4)
if recv4[:3] != '354':
  print('354 reply not received from server.')

# Send message data
clientSocket.send(msg.encode())

# Message ends with a single period.
clientSocket.send(endmsg.encode())
recv5 = clientSocket.recv(1024).decode()
print(recv5)
if recv5[:3] != '250':
  print('250 reply not received from server.')

# Send QUIT command and get server response.
quitCommand = 'QUIT\r\n'
clientSocket.send(quitCommand.encode())
recv6 = clientSocket.recv(1024).decode()
print(recv6)
if recv6[:3] != '221':
  print('221 reply not received from server.')

# Close the socket after sending QUIT
clientSocket.close()
```