```
from socket import * # Import socket module
import sys, os, errno

# Create a TCP server socket
#(AF_INET is used for IPv4 protocols)
#(SOCK_STREAM is used for TCP)

clientSocket = socket(AF_INET, SOCK_STREAM)


clientSocket.setsockopt(SOL_SOCKET, SO_RCVBUF, 1)
clientSocket.setsockopt(SOL_SOCKET, SO_RCVLOWAT, 5000)

# Assign a port number
# serverPort = 6789

if len(sys.argv) < 4:
      print("Usage: python3 " + sys.argv[0] + " serverAddr serverPort filename")
      sys.exit(1)
serverAddr = sys.argv[1]
serverPort = int(sys.argv[2])
fileName = sys.argv[3]

# Connect to the server
try:
      clientSocket.connect((serverAddr,serverPort))
except error as e:
      print("Connection to server failed. " + str(e))
      sys.exit(1)

print('------The client is ready to send--------')
print(str(clientSocket.getsockname()) + '-->' + str(clientSocket.getpeername()))

try:
      getRequest = "GET /" + fileName + " HTTP/1.1\r\nHost: " + serverAddr + "\r\n"
      getRequest = getRequest + "Accept: text/html\r\nConnection: keep-alive\r\n"
      getRequest = getRequest + "User-agent: RoadRunner/1.0\r\n\r\n"
      clientSocket.send(getRequest.encode())
      # clientSocket.send(("GET /" + fileName + " HTTP/1.1\r\n").encode())
      # clientSocket.send(("Host: " + serverAddr + "\r\n").encode())
      # clientSocket.send("Accept: text/html\r\n".encode())
      # clientSocket.send("Connection: keep-alive\r\n".encode())
      # clientSocket.send("User-agent: RoadRunner/1.0\r\n\r\n".encode())
except error as e:
      print("Error sending GET request: " + str(e))
      clientSocket.close()
      sys.exit(1)

message = ""
while True:
      try:
            newPart = clientSocket.recv(1024)
            message = message + newPart.decode()
            if not newPart:
                  print (message, flush=True)
                  break
            if message[len(message)-1] != "\n":
                  continue
            else:
```

```
                print(message, flush=True)
                message = ""
      except error as e:
            print('Error reading socket: ' + str(e))
            sys.exit(1)

print("Buffer size: %d" %clientSocket.getsockopt(SOL_SOCKET, SO_RCVBUF))
clientSocket.close()
sys.exit(0)
```