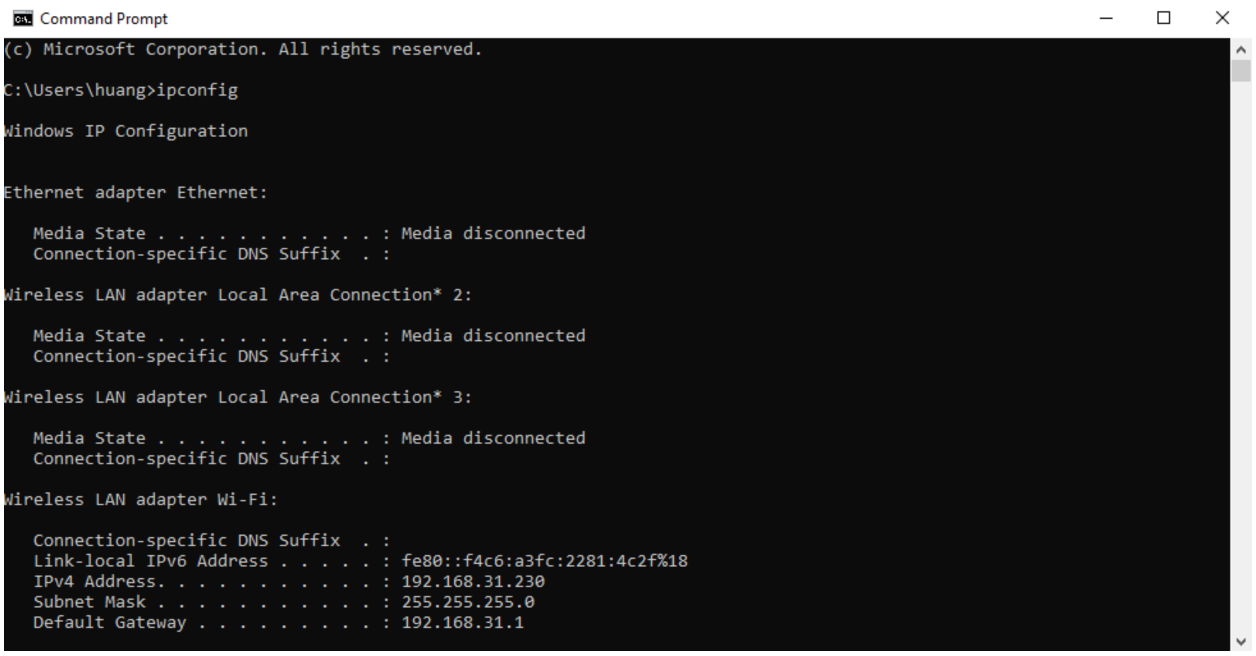
CS-GY 6843 Wireshark Socket Programming 1

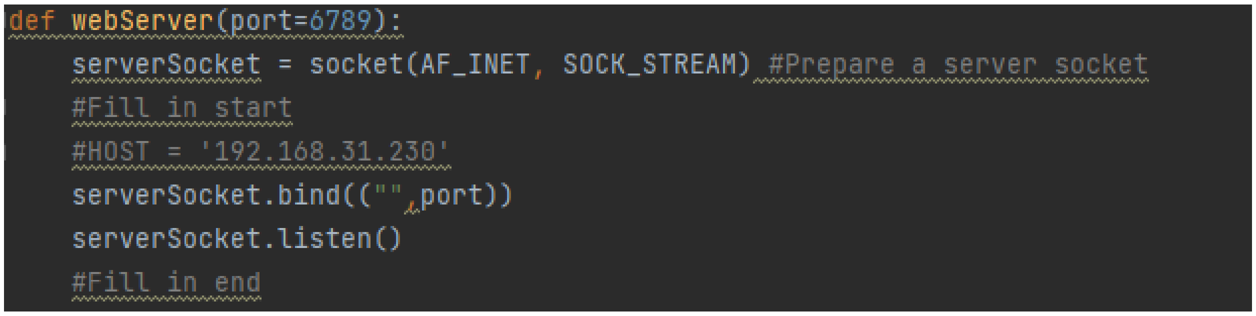
02/26/2022

Yibei Huang



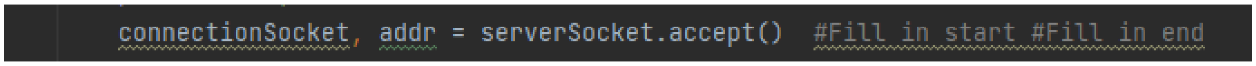
```
(c) Microsoft Corporation. All rights reserved.

C:\Users\huang>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 2:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Local Area Connection* 3:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::f4c6:a3fc:2281:4c2f%18
   IPv4 Address. . . . . . . . . . . : 192.168.31.230
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.31.1
```

For preparing the server socket, we need to first bind the serversocket to specific IP address and port number. I left the IP address blank so that it can be reachable by all networks. And the socket.listen make the socket start listening incoming connection.



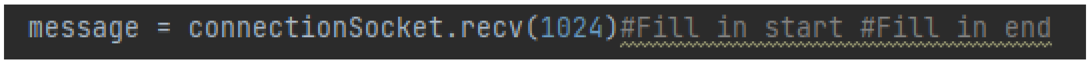
```
def webServer(port=6789):
    serverSocket = socket(AF_INET, SOCK_STREAM) #Prepare a server socket
    #Fill in start
    #HOST = '192.168.31.230'
    serverSocket.bind(("",port))
    serverSocket.listen()
    #Fill in end
```

The socket.accept will catch the address and socket information of the incoming connection.



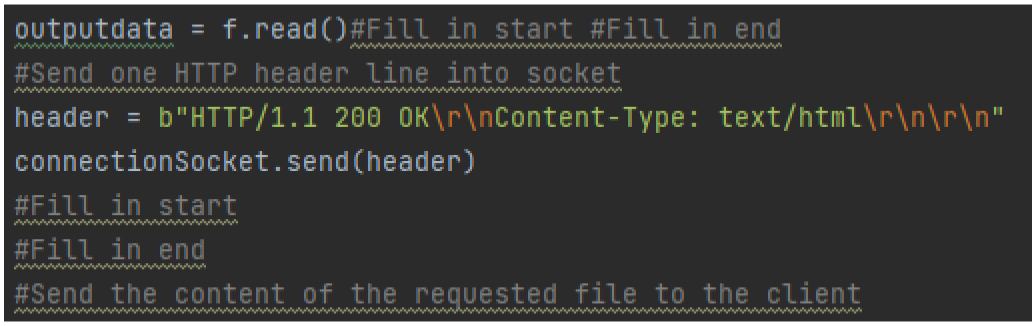
```
        connectionSocket, addr = serverSocket.accept()  #Fill in start #Fill in end
```

The socket.recv will receive the data from the connection until no more data to read



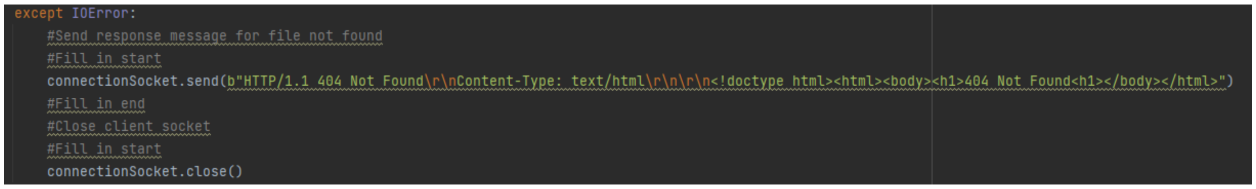
```
 message = connectionSocket.recv(1024)#Fill in start #Fill in end
```

The f.read reads all the data of the file. After defining the head information we use socket.send to send the header information to the client. The "\r\n\r\n" left return lines for following file content data.



```
outputdata = f.read()#Fill in start #Fill in end
#Send one HTTP header line into socket
header = b"HTTP/1.1 200 OK\r\nContent-Type: text/html\r\n\r\n"
connectionSocket.send(header)
#Fill in start
#Fill in end
#Send the content of the requested file to the client
```

When an error catched, say if the file is not found then we just use socket.send to send a different header this case "404 NOT FOUND" followed by some data content(html).



```
except IOError:
    #Send response message for file not found
    #Fill in start
    connectionSocket.send(b"HTTP/1.1 404 Not Found\r\nContent-Type: text/html\r\n\r\n<!doctype html><html><body><h1>404 Not Found<h1></body></html>")
    #Fill in end
    #Close client socket
    #Fill in start
    connectionSocket.close()
```