Data Networking

Python Project

README:

For creating the Server using Python, I followed following steps:
1) Create a socket on the assigned port and localhost
2) Bind the socket to that Port
3) Start listening through that port
4) As soon as the client requests a webpage through the browser store the URL
5) Scan the URL and split it to retrieve the exact name of file
6) Search for the specified file in the Directory
7) Open the file
8) Read it and store it in a variable
9) Send the file to the browser by passing variable to the socket
10) Print the Status for successfully sending the file
11) Measure the length of file
12) Display the Content length and Content type on the server.
13) If Client has requested for any other file, display the 404 status code
14) Close the connection and break the loop.

The only Problem I faced during this assignment was on how server can send only a specified file and gives the error message to other requested files. For this, I referred to Coursera material for Python and there I got this idea of URL splitting.

Python code for the server:

```
from socket import *
port_server=8080
socket_server = socket(AF_INET, SOCK_STREAM)                    #Create the socket
socket_server.bind(('',port_server))                           #Bind the socket to assigned port and localhost
socket_server.listen(1)                                        #Listen on the specified Port
print 'The Server is listening on the port:', port_server
while True:                                                    #Wait for the request to arrive
        print 'Waiting for the request....'
        socket_client, addr = socket_server.accept()          #Accept the request from client's socket/Browser
        try:                                                  #If Browser requested for the HTML file'
                url = socket_client.recv(1024)                #Store it in the url
                file_name = url.split()[1]                    #Split the URL with whitespace as a delimiter
                p = open(file_name[1:])                       #Open the 1st element of the filename
                file_data = p.read()                          #Read the file and store it in the variable
                socket_client.send(file_data)                 #Send the file to the Client/Browser
                k = len(file_data.encode('utf-8'))            #Measure the length of HTML file in bytes
                socket_client.send('\n\r\n HTTP/1.1 200 OK \r\n')   #Send the Status code to the Client
                socket_client.send('\n\r\n Content-Type: text/html \r\n')#Send the Content type to the Client
                socket_client.send('\n\r\n Content-Length: 458 \r\n')    #Send the Content Length to the Client
                print 'HTTP/1.1 200 OK'                       #Display the status code on the server
                print 'Content-Type: text/html'              #Display the Content type on the server
                print 'Content-Length: ',k                   #Display the Content length on the server
                socket_client.close()                        #Close the Socket
        except:                                               # If Browser has requested for any other file
                print "404 Not Found"                        #Print the error response on the Terminal window
                socket_client.send('\n\r 404 Not Found\r\n') #Send the error response to the Browser
        break                                                 #Serve the client's request only for one time
```

HTML File:

```
<!DOCTYPE html>
<html>
<head>
 <title>DN</title>
 <style>
  body {
        color: yellow;
        text-decoration: underline;
        text-align:center;
        background-color:black;
         }
 </style>
</head>
 <body>
        <BR><BR><BR><BR><BR><BR><BR><BR><BR>
        <p>DN Project 1 - Python Assignment</p>
        <BR>
        <p>To implement a HTTP server in Python which handles just one request from
user</p>
        <BR>
        <p>Sample HTML file to be accessed by server</p>
        <BR>
 </body>
</html>
```