ASSIGNMENT 1 “My Web Browser” ET1542 Lp2 2020

PART A: PREPARING

Task A1 Install Python

The task is to test socket programming using the book / lecture python code samples and use it between two computers. If you don’t have two computers, please test it with one of your student friends in the course.

- Install python on your computer if needed. Python can be downloaded at python.org
  If you want to learn the language, there are much help online. Here is an example:
- Direct link to a pdf book "Thinkpython"
  http://greenteapress.com/thinkpython2/thinkpython2.pdf (Python ver 3)
  The website of the book. http://greenteapress.com/wp/think-python-2e/

Task A2 Test a UDP connection

Get the sample code on Canvas. Use "UDPKlient.py" on one computer and "UDPServer.py" on another computer. Enter the correct ip-number as "hostname" in the code.
You need to find out your computer's IP number.
Think/ google on how to do it. Keep in mind that your computer may have different IP numbers on the wired connection and the wireless connection.

TASK A3 Test a TCP connection

Use "TCP client.py" on one computer and "TCPServer.py" on another computer. You probably notice no difference compared to UDP as there is so little information sent between client and server.

Please note

- This program code is the basis for later lab assignments
- Feel free to observe data exchange between client and server using Wireshark

TASK A4 Brief Summary

In your submitted report you have to write a brief summary on how it was possible to run the sample code. Please indicate if it went well / bad, if you had some problems with python or your computer / network etc.

PART B MY "MY WEB BROWSER"

Write a program in Python (or other language) that

TASK B1

Uses socket programming to contact the web page
www.ingonline.nu/tictactoe/index.php
with the aim of getting the next move in the game Tic-Tac-Toe.

NOTE! Different library functions (http / url functions) in Python that only uses standard http (port 80) may not be used. See detailed instructions below. The program should be able to contact a webserver (http-server) on port 80.

TASK B2
The program should act as a web browser and create an “http request”-message, according the http-protocol, requesting the tic-tac-toe web page and instruct the server to close the connection. It is therefore important that you understand the different fields in the HTTP request. Read chapter 2.2.3 in the course book which describes "http request message format".

TASK B3
The program should then display the http response with the header and the body which includes the HTML file from the web server. You should display the http response as a web browser does, with the decoding and interpretation of the html code. You only need to interpret the html syntax included in this response web page (not a general webpage html file).

TASK B4
Also capture the communication between your computer / browser (client) and the web server with Wireshark by filtering the communication so that no other packets are displayed. Filter only the packets between the client's ip number and the web server's ip number are displayed.

a) Take a screenshot of "Flow Graph" (will be shown at the lecture).

b) Using wireshark, also view the contents of the packet from your computer containing the http request as well as the contents of the packet from the server containing the http response. (screen shot is ok)

<u>SUBMISSION DOCUMENT</u>

The submission document must have the following content:

- Name, social security number, course code, date (This is first page)
- Summary from Task A4
- Python code from TASK B1
- Results of PART B
    - Printed HTTP response from the server TASK B3
    - "Flow Graph" screenshot TASK B4a
    - Packet content of the http request and http response TASK B4b, (i.e. screenshot from Wireshark)
    - Brief summary of PART B

Send in the submission document as a pdf-file.

NOTE: It is individual submission.
If you have worked with someone, please mention the name on the first page (plagiarism check is on).

INSTRUCTIONS WHEN SENDING HTTP-REQUEST TO TIC_TAC_TOE WEB SERVER

To write a URL to the Tic-Tac-Toe server you need some instructions:

- We will use the HTTP GET method to send the game board to the server. Read about the GET vs POST method below.
- The server responds by sending back a web page with the game board before and after the move
- You specify the game board according to this URL structure
  www.ingonline.nu/tictactoe/index.php?board=xoxoeoeex

  Try this out in your ordinary web browser!

  x = the server, o = the client (you), e = empty square
  This example gives the following set up

  x o x
  o e o
  e e x

- Always enter 9 characters as the board argument.
- The server's script / program is not a real tic-tac-toe program and will only move the first x to the first empty square.

EXTRA INFORMATION ABOUT GET AND POST:

The GET Method
Note that the query string (name/value pairs) is sent in the URL of a GET request:
/test/demo_form.php?name1=value1&name2=value2

Some other notes on GET requests:
GET requests can be cached
GET requests remain in the browser history
GET requests can be bookmarked
GET requests should never be used when dealing with sensitive data
GET requests have length restrictions
GET requests should be used only to retrieve data

The POST Method
Note that the query string (name/value pairs) is sent in the HTTP message body of a POST request:
POST /test/demo_form.asp HTTP/1.1
Host: w3schools.com
name1=value1&name2=value2

Some other notes on POST requests:
POST requests are never cached
POST requests do not remain in the browser history
POST requests cannot be bookmarked
POST requests have no restrictions on data length