UTS Insearch
UNIVERSITY OF TECHNOLOGY SYDNEY

PROJECT 1

Diploma of IT
Network Fundamentals

Subject Credit Points:
Subject Coordinator: Claire Pierse
Weight: 10%
DUE:

PROJECT 1 : SOCKET PROGRAMMING WEB SERVER

TASK DESCRIPTION

In this assignment, you will learn the basics of socket programming for TCP connections in python: how to create a socket, bind it to a specific address and port, as well as send and receive a HTTP packet. You will also learn some basics of HTTP header format.

OBJECTIVES

You will develop a web server that handles one HTTP request at a time. Your web server should accept and parse the HTTP request, get the requested file from the server's file system, create an HTTP response message consisting of the requested file preceded by header lines, and then send the response directly to the client. If the requested file is not present in the server, the server should send an HTTP "404 Not Found" message back to the client.

BRIEF

CODE

Below you will find the skeleton code for the Web server. You are to complete the skeleton code. The places where you need to fill in code are marked with #Fill in start and #Fill in end. Each place may require one or more lines of code.

RUNNING THE SERVER

Put an HTML file (e.g., HelloWorld.html) in the same directory that the server is in. Run the server program. Determine the IP address of the host that is running the server (e.g., 128.238.251.26). From another host, open a browser and provide the corresponding URL. For example:

http://128.238.251.26:6789/HelloWorld.html

'HelloWorld.html' is the name of the file you placed in the server directory. Note also the use of the port number after the colon. You need to replace this port number with whatever port you have used in the server code. In the above example, we have used the port number **6789**. The browser should then display the contents of HelloWorld.html. If you omit ":6789", the browser will assume port 80 and you will get the web page from the server only if your server is listening at port 80. Then try to get a file that is not present at the server. You should get a "404 Not Found" message.

SUBMISSION

WHAT TO SUBMIT FOR PROJECT 1:

Document your project in report format and include relevant screen clippings of the process. This report can be in .docx or pdf format. Upload to the canvas Project 1 assignment link

WHAT TO SUBMIT FOR PROJECT 1:

(1) Project report. This report can be in .docx or .pdf format.
(2) The completed code - python file(.py)
(3) Video - In addition to the written submission, you will create a short video using Canvas Studio. The video should verify that:

a) you receive the contents of the HTML file from the server, the HTML file is displayed in the browser
b) when you try to get a file that is not present at the server, you will get "404 Not Found" message
c) your aim is to demonstrate your understanding of the project by explaining the python code that you have written
d) the video should be approx. 3 minutes in duration.

Note your tutor may ask specific questions to each of you to check your work and your understanding of the project.

SKELETON PYTHON CODE FOR THE WEB SERVER

```
#import socket module
from socket import *
import sys # In order to terminate the program

serverSocket = socket(AF_INET, SOCK_STREAM)
#Prepare a sever socket
#Fill in start
#Fill in end
while True:
    #Establish the connection
    print('Ready to serve...')
    connectionSocket, addr =   #Fill in start              #Fill in end
    try:
        message =   #Fill in start          #Fill in end
        filename = message.split()[1]
        f = open(filename[1:])
        outputdata = #Fill in start       #Fill in end
        #Send one HTTP header line into socket
        #Fill in start
        #Fill in end
        #Send the content of the requested file to the client
        for i in range(0, len(outputdata)):
            connectionSocket.send(outputdata[i].encode())
        connectionSocket.send("\r\n".encode())

        connectionSocket.close()
    except IOError:
```


```
        #Send response message for file not found
        #Fill in start
        #Fill in end
        #Close client socket
        #Fill in start
        #Fill in end
serverSocket.close()
sys.exit()#Terminate the program after sending the corresponding data
```

EXTENSION CHALLENGE - TO SECURE "HD" GRADE

Currently, the web server handles only one HTTP request at a time. Implement a multithreaded server that is capable of serving multiple requests simultaneously. Using threading, first create a main thread in which your modified server listens for clients at a fixed port. When it receives a TCP connection request from a client, it will set up the TCP connection through another port and services the client request in a separate thread. There will be a separate TCP connection in a separate thread for each request/response pair.

Instead of using a browser, write your own HTTP client to test your server. Your client will connect to the server using a TCP connection, send an HTTP request to the server, and display the server response as an output. You can assume that the HTTP request sent is a GET method.The client should take command line arguments specifying the server IP address or host name, the port at which the server is listening, and the path at which the requested object is stored at the server. The following is an input command format to run the client. client.py server_host server_port filename

ASSESSMENT CRITERIA

CRITERIA	WEIGHT	SLOs	PLOs
System Design	**20**	**2**	**B2**
Develop and implement designs	**40**	**4**	C2
Test Solutions	**40**	**3**	B2

SLOs: subject learning outcomes
PLOs: program learning outcomes