Jordan University of Science & Technology
Department of Network Engineering and Security
NES416- Network Programming
Programming Assignment 4 (CLO2, CLO3, CLO4)
Due Date: See E-Learning

Goal: investigate multithreaded TCP/UDP socket programming

Description:
You will implement a client/server multi-threaded character counting applications that responds to TCP requests.

When the serve runs, it listens to the assigned TCP port (in fact, it will take service_name not port number as an argument). When a TCP connection is made to the port, the server (actually the corresponding thread, i.e the server is multithreaded) will create a separate thread for each connection, process the request, and reply to that connection with how many a given character is repeated in a given string. The client when collecting all the results will print out the overall result and close the connection.

Your client should take hostname and port number as command arguments. However, you need to use server name in your code rather that its IP address. That is, you need to edit the `/etc/hosts` file in your machine and add a line that specifies the server name as ***nes416_server*** and assign it the IP address of your machine. In your code you should use the appropriate functions to convert between hostnames and IP addresses. Furthermore, you need to use the service name in the command line rather than the port number. Edit the appropriate file(s) and use the ***countp*** as your service name. for more information, see the man page of the corresponding files for more information

The client program should be composed of some number of threads. The number is determined by the user from the standard input. i.e when the client runs, it asks the user to enter three pieces of information. For example, if you want to search for character N (case sensitive) in the string "Hello NES students" using 4 threads, you would ask the user the following

```
Enter the string: Hello NES students
Enter the character to search for: N
Enter number of required threads: 4
```

Each thread created by the client will be allocated part of the original string (substring) and asks the server to count the given character in that substring. The client, when all the threads finishes, should output the final result to the user indicating the total number of the given character in the original string. the original string is divided between threads equally (of course, the last thread might have less , if the string size in not evenly divisible by number of threads). Note you can't have threads created without being assigned a substring to work at.

Based on that, the client will create the required number of threads, each of which will connect to the server asking to look for the given character in a substring. The character to look for will be the first one in the sent request.

Hints:

- ☐ Ask questions as early as possible.
- ☐ DO NOT use the header file "unp.h" from the book
- ☐ Submit your source code one zipped file according to the naming rules
- ☐ Your programs should be compiled and run without any single error or warning.
- ☐ Comment and **error-check** you code
- ☐ NO CHEATING/COPYING
- ☐ Your program for the client needs to take two arguments that specify the **hostname** and the **service name** of the server. Your program for server needs to take an argument that specifies the **service name** to work on
- ☐ Don't use the bind() function on the clients
- ☐ Test your code by running the three clients simultaneously.