Programming Assignment 1

(Due Saturday, Oct 3, 2015 at 11:59pm – **max 50% credit for up to 24hr late**)

This assignment is fairly simple and will give you some practice with C/C++ network programming. It is imperative to successfully complete and understand this assignment for subsequent work. Feel free to speak to the course staff if you are facing difficulty.

About cooperation with other students:

This assignment is meant to be done alone. Absolutely **NO** cooperation is allowed. If there are questions, ask the course staff; they are there to help you. For this assignment, you must do all thinking, research, and coding by yourself. ***Do not even discuss with anyone how far you are with this assignment!***

ANY HELP (EXCEPT FROM COURSE STAFF) IS PROHIBITED!!!

Individuals found guilty of violating above policy will be referred to the disciplinary committee.
(Warning: We will use software to measure the code similarity)

More Important Note:

We are giving you an additional responsibility to REPORT all incidences of cooperation that you may observe. You MUST report to the instructor (or the TAs) such incidences. If you do not, we will consider you as guilty as those who you witnessed cooperating. All coding, debugging, web search for functions, etc. must be done by individual students!

For syntax related issues, make sure to consult www.google.com before asking the course staff. They will do the same!

Preamble:

You will use **Linux programming environment** for this assignment. An account has been created for you on course server (110.93.234.139). If your student ID number is 20xy-zw-0abc, then your username will be nxyzwabc (e.g., username for 2015-10-0893 is n1510893). We suggest you use the **SecureCRT** client or **putty** client to telnet (ssh) to the course server. These two are much nicer than the windows default telnet client. Your username and password will be emailed to you; you **may** reset the password to something you can easily remember (Google for how to change your Linux password). If you ever forget your password (we suggest you do not), contact the instructor. You MUST NOT share your password with anyone including the course staff.

You will need to submit all the files zipped in a file whose name is the same as your student ID. **See submission instructions for more details and follow them precisely.**

The Assignment:

The assignment consists of one optional part and two (very) easy parts and are based on the material covered in class (or assigned for reading or explained in tutorial).

Part-I: Create a simple text-based webpage in your home directory on the course server. Write something creative but not more than a few sentences of text on that webpage. The information on the page must include your name and roll number somewhere. Make your webpage accessible at http://110.93.234.139/~nxyzwabc. All you need to do is modify the index.html file in the public_html directory within your home directory.

For Part-I, you do not need to submit anything – it is just to give you practice with editing on course server. We very strongly encourage you to use an editor on course (emacs, nano, vi, etc.) rather than writing stuff in notepad and then moving it to server.

<u>Part-II:</u> This part will make you familiar with code that performs DNS operation. We are providing you the skeleton code which you **should not** change, except for the additions of your own code. The filename we are providing you is mydns.cc. Please do not change the filename. The program takes the fully qualified hostname of a host as a command line argument and prints a list of corresponding IP addresses, in dotted decimal notation, one on each line. Your program must not crash if a bogus hostname is provided. Well, it should not crash in any case. As a starter, simply type nslookup <hostname> on the command prompt (on course server) to see something similar to what we are expecting. Observe the output of nslookup using following hostnames:
www.cmu.edu
web.mit.edu
www.yahoo.com
chand.lums.edu.pk
www.facebook.com
Verify that mydns.cc is also resulting similar results for above hostnames. For this part, you will be using the function `gethostbyname()` which makes your life much easier as the dns implementer. You do not need to worry about creating UDP sockets and then sending out DNS messages. Similarly, no worries about NS, CNAME and A type resource records. The function `gethostbyname()` will do it all for you. How lucky!

All you will need to do is to interpret the values returned by `gethostbyname()` and display them in dotted decimal notation. You will need a couple of standard C and socket API functions to complete this assignment. Please ask the TAs only after you have researched them on google. The starting point will be to understand **what** is returned by the function `gethostbyname()`.

The skeleton file is in C and does not use any C++ specific operations. But you are free to add extra header files and use C++ functionality. More specifically, you might want to use `cout` instead of `printf` statements. See class website for starter file.

<u>Final word about compiling</u>: On some machines, for example suraj server or one of your local machines, g++ may require adding a network library (using `-lxnet`) on command line. There is no such need on course server. Use `-g` option on the command line if you need to debug your code using gdb debugger.

<u>Part-III:</u> In this part, you will write a short client program that gets the time from the time server running on course server (or suraj or some other) machine. The starter code is provided once again and you need to implement a client which connects to a server at TCP port 13 to retrieve the time of the day. If the server behaves and responds, your client displays the time of the day and quits. If the server does not exist, your client should gracefully quit after reporting that the server can not be contacted. If the server machine exists but is not running the time server, your program should report so and then quit. Finally, some servers may not respond at all in which case, we do not require you to do much unless you are a bit ambitious. In any case, your program should not crash. A **reference implementation** is available on the class website (hosted at LMS) and runs on the course server machine.

The server port is fixed (13) but the server name is provided at the command prompt as a requirement to run the program. Once again, the starter code (provided in mydaytime.cc) is in C but you will use g++ to compile the program and, therefore, are free to use additional C++ functionality. But do NOT change the skeleton code we provided.

Where do I start and where do I get help?

Start reading the Beej's guide to Unix Socket Programming. **It IS useful.** Or talk to the teaching staff if you have any questions. You should also consult man pages (manual pages, if you like) on the course server machine; this is available by simply typing `man gethostbyname` on the command prompt.

What and Where to submit?

1. Very importantly, **strictly** stick to the following submission guidelines.
2. Use the names of the files specified in this handout.

(a) For Part-I, we do not need anything.
(b) For Part-II, we need the completed mydns.cc file.
(c) For Part-III, we need the completed mydaytime.cc file.

3. Zip all these files together in one file named <your_student_num>.zip where <your_student_num> is an **8-digit** number (not your username on the course server). Be very careful with this, our script might throw away zip files that do not follow this convention. **No contest will be allowed in that case.**
4. **Email submissions will not be accepted.**
5. Do just one submission of your final zip file through LMS (use the ***Assignment link*** in your portfolio, and **NOT** the Dropbox).
6. **If you are not enrolled in the course,** you may attempt the assignment for your own practice but we will NOT grade your assignment; do not email your assignment to us. Once again, do not send your assignment to us unless you are registered for the course.
7. **Last but not the least:** The course server IP address is 110.93.234.139. You have an account on this server. If, for some reason, you are unable to connect to this, you can try connecting to venus.lums.edu.pk (from on-campus only). If that also is not an option, you can work on an alternate server mureekh.lums.edu.pk which is accessible from inside and outside LUMS. We have also created an account for you on the mureekh server.