ENTS 689N: Network Programming

Final Project, due 12/15/2017 8:30 pm

- Write each program (client and the server) in a separate file and name it properly. Using comments write name and UID of your group members on top of your program.
- Attach all your files to one email and send it to (masoud@umd.edu). Include ENTS689N in the subject line.
- Indent properly, use comments when needed and choose appropriate variable names.
- ***Do not copy*** programs from anyone (and don't let anyone copy your code).
- Make sure that your codes can be compiled and executed.

IP & Domain Dossier System

The project consists of two parts, a client and a server which communicate based on TCP/IP protocol. The server uses a text file fostering the domain and IP database. In the text file each row shows a domain name as well as number of times that the record has been requested followed by the IP address(es) which correspond to that domain. Sample content of the text file is as follow:

```
www.yahoo.com  5  98.138.253.109          206.190.36.45 98.139.183.24
www.google.com  0 64.233.169.99
www.umd.edu   15 128.8.237.77
```

When the server starts, it reads the data from the file into linked list. Server is capable of performing following tasks:

1. Return the IP address(es) for a domain
   a. If the IP address exists in the data file, it will be returned. In case of multiple addresses all must be returned.
   b. If the IP address does not exist, the server tries to find and add it to the database (hint: use *gethostbyname()*).
   c. If nothing is found, appropriate message is communicated
2. Keep track of number of requests for each record
3. Add new record to the list
4. Delete a record from the list
5. Reject inquiry if another inquiry has been received from the originated address in the last X seconds.

The server receives port number, data file name and accepted time gap between multiple requests (in seconds) as command line parameters.

The client uses command line arguments to communicate with the server. The first command line parameter is the server IP and the second parameter is the port number. The third parameter specifies the type of request from the server and can have the following values:

Request code	Action
1	Find IP for a domain
2	Add a record to the list
3	Delete a record from the list
4	Report the most requested record(s)
5	Report the least requested record(s)
6	Shut down

Some of the requests need extra parameters which will follow the third parameter. Here are some examples (assume that the .exe file name is **client**):

Command	Action
client 129.2.133.250 1070 1 www.yahoo.com	Server returns the IP for www.yahoo.com
client 129.2.133.250 1070 2 www.ebay.com 10.125.6.138	Server adds www.ebay.com and its IP to the list
client 129.2.133.250 1070 3 www.ebay.com	Server removes www.ebay.com record from the list
client 129.2.133.250 1070 4	Server returns the most requested record
client 129.2.133.250 1070 5	Server returns the least requested record
client 129.2.133.250 1070 6 SECURITYCODE	Server saves the list into the file and shuts down

The last command (shut down) should only be executed if the request is followed by a predetermined security code. Upon shut down changes to the list must be written into the data file. Both client and server must detect errors in the input and respond properly. Error checking can be done either at the client or the server side. If you check for the input errors at the server side, the server must transmit appropriate error message to the client should an error occur. The following are some errors that you should handle:

- Number of command line parameters is not enough for the request
- The record to be deleted does not exist
- The record to be added already exists
- The IP address format is wrong (e.g. 999.888.123.3 is not a valid IP address)
- Another inquiry had been made 6 seconds ago, wait 10 seconds before another submission

Note that the program must be case insensitive (i.e. www.ebay.com and www.eBay.com are the same)

SUBMIT A MAXIMUM 4 PAGE REPORT IN PDF or WORD FORMAT ALONG WITH YOUR PROJECT, TO CLARIFY AND EXPLAIN IMPORTANT PARTS OF YOUR CODE.

OPTIONAL FEATURES FOR BONUS:

You need to do both of the two following optional features to, upgrade your final grade to the next level (B^- -> B -> B^+ -> A^- -> A-> A^+):

1. Implement a binary search tree to improve the performance.
2. Produce a log file for all the events that occur on the server side. Each line in the log file starts with a date and time, and provides detailed information for the event. The following shows an example:

```
2011/12/8  5:30 ("www.test.com" , "123.2.3.45") added by client 64.2.1.56
2011/12/8  5:32 request for shutting down the server from client 64.2.1.56 failed
2011/12/8  18:32  Server shut down by the client 128.2.132.54
```