**Project Report**

By

Srihareendra Bodduluri

Shreyas Pandey

***Aim :***

The project consists of two parts, a client and a server which communicate based on TCP/IP protocol. The server performs following operations

* Returns 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 (

c. If nothing is found, appropriate message is communicated

* Keep track of number of requests for each record
* Add new record to the list
* Delete a record from the list
* Reject inquiry if another inquiry has been received from the originated address in the last X seconds.

***Operating System: Ubuntu***

***Platform : C***

***Header file: uthash.h***

***Server Side Code Explanation***

**IMP**: *MAKE SURE THE INPUT FILES ARE IN THE PATH and uthash.h file is on server folder*

Command Line Arguments : **[<port number>] <location of Ip databased.txt> [<retry time.>].**

**8889 Domain.txt 5**

**OUTPUT: Domain.txt, log.txt**

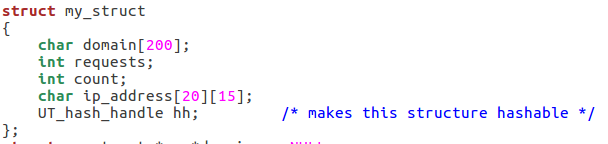
**DataStructure: HashTable used to store Domain information and corresponding IP addresses.**

Domain name is the Key of the hash table. (char array)

Request field contains information regarding number of requests (int)

Count gives number of Ip addresses associated with a domain (int)

Ip address is an array of IP addresses (up to 15 max)



Basic Work flow

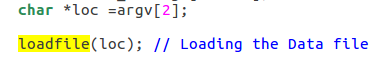
1. Load Data into hashTable
2. Create a Socket
3. Bind Socket
4. Listen
5. Accept
6. Switch (receive)
7. Case 1: Returns the IP address(es) for a domain.

Case 2: Add a new record to the list.  
 Case 3: Delete record from the hash table.  
 Case 4: Return the max requests  
 Case 5: Return the min requests  
 Case 6: Shut down and Record the hash

We have four helper functions:-

1) 

This function loads the datafile (Domain.txt in our case) which contains the details of the DNS and corresponding IP addresses.



2) 

This function contains all the six different cases and depending upon the request the cases are executed

3) 

For option 2nd, when a new entry is to be added then this function is called from HandleTCPClient to add a new entry.

4) 

This function adds a ‘\0’ at the end of the string.

In the main function, a variable ‘flag’ is used to check if it is a first client request.

Depending on it a user based second counter is set up so that at least n-second gap is maintained between requests.

Aditionally Log is generated for every database change and shutting down the server at (log.txt)

***Client Side Code Explanation:-***

Basic Work Flow

1. Create a socket
2. Connet to server
3. Send Message
4. Receive Response

Client needs to connect to the server port or else it will exit

We have created a function:-

1) 

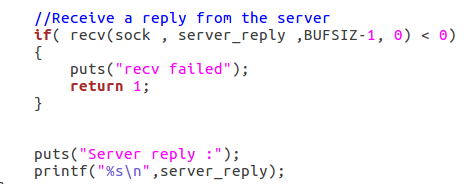
This function checks the validity of the IP address which passed through the command line arguments.

In this function, we check if the correct number of argument is passed.

We also check the IP address of server passed is correct through the method check\_ip.

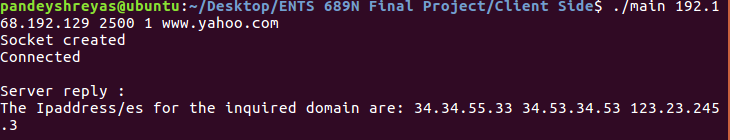
If the argument passed is correct, we use connect() method to connect with the server and once connection is setup, the data is send by send() method.

After the data send is checked and corresponding output is send to client back recv() function is used to receive the data from the server end.

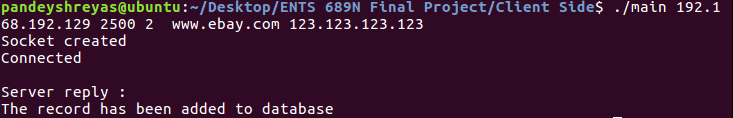


***OUTPUTS:***

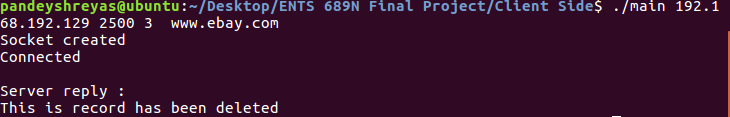
Output for Option 1(Getting IP address from DNS)



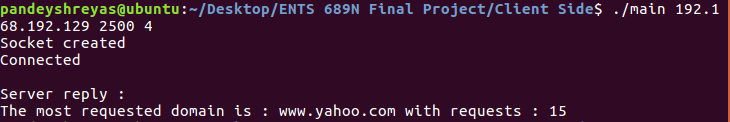
Output for Option 2(Record to be added in database)



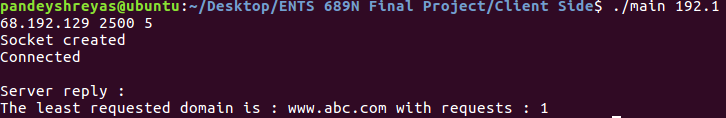
Output for Option 3(Record Deletion)



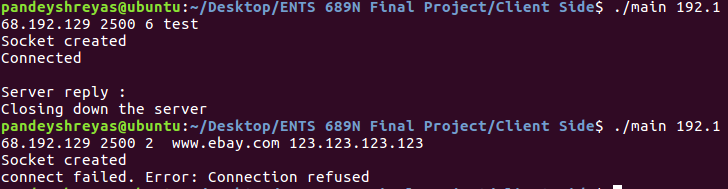
Output for Option 4(Maximum Search)



Output for Option 5(Minimum Search)



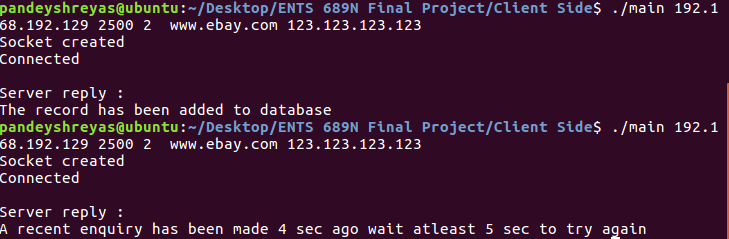
Output for Option 6-Server Closed



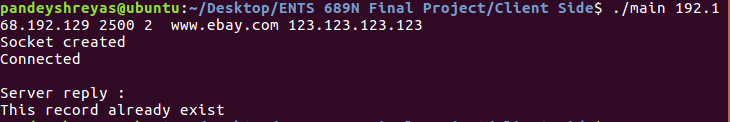
Invalid Command Line Argument passed.

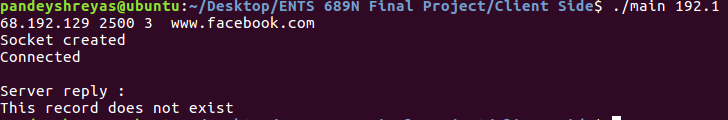


ime Gap set to 5 seconds. Second Command line argument was send within 1 second.



When record already exists.



When the record does not exist.

Passing invalid IP address

