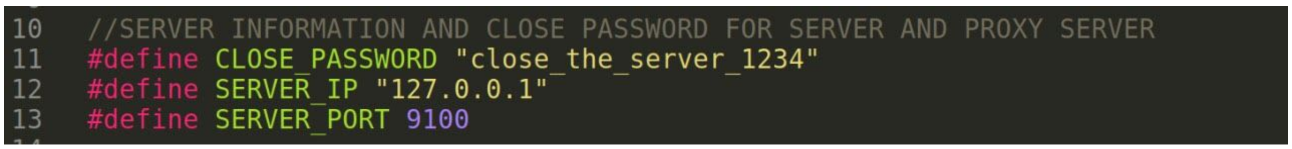
Assignment-3
CS342
Group 36

B Venkatesh - 180101013
Harsh Motwani - 180101028

Important Instructions:

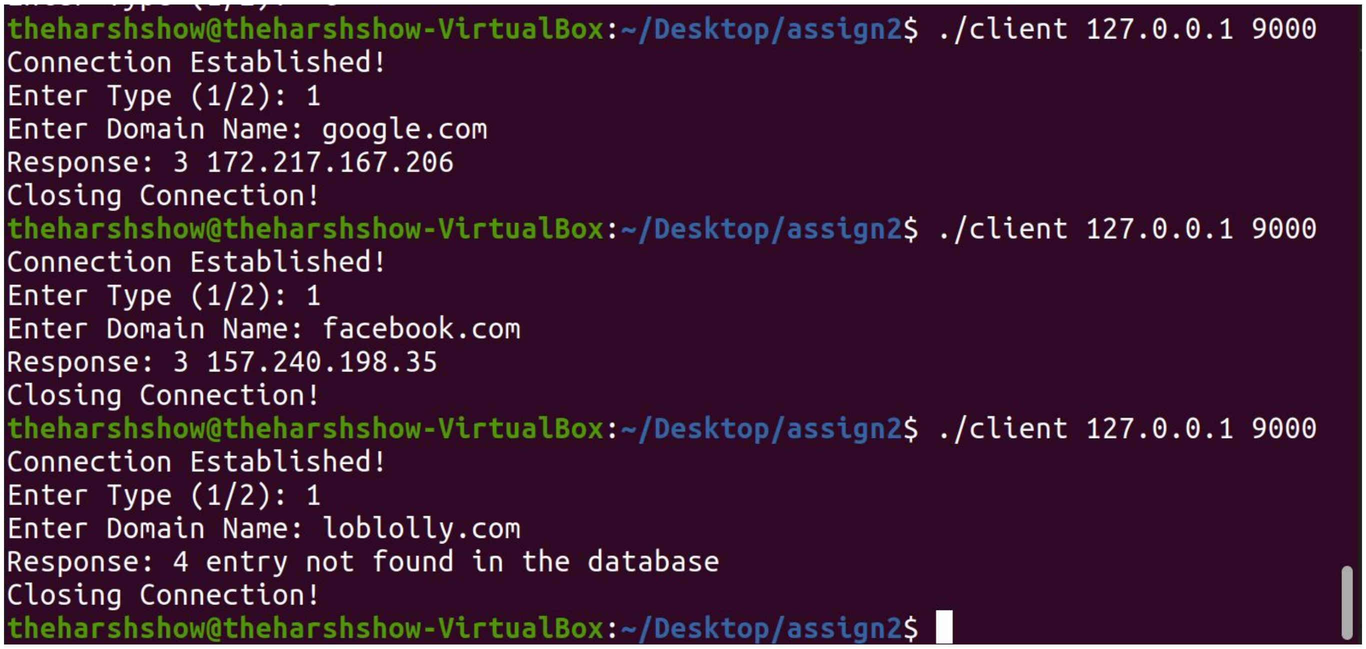
- In order to create executable files for all three participants - DNS server, client and proxy server - you have two options:
    - Compile each one of them individually with:
        - `gcc server.c -o server`
        - `gcc proxy_server.c -o proxy_server`
        - `gcc client.c -o client`
    - Or simply run '`./execute.sh`' (A shell script we provided in the folder). If it doesn't work, change it's permissions with '`chmod 777 execute.sh`'
- Now that the executables are ready, it's time to run them:
    - Start with the DNS Server: `./server 9100`
    - Then to the Proxy Server: `./proxy_server <Port of your choice greater than 2000>`
    - Then the client server: `./client <IP Address of Proxy Server> <Proxy Server Port>`
- You can do the following (All functionalities from the client side):
    - **Get IP Address corresponding to a Domain Name (Type 1).** For this, enter 1 in the first field and enter the domain name in the second.
    - **Get Domain Name corresponding to an IP Address (Type 2).** For this, enter 2 in the first field and the IP Address in the second field.
    - **Close the Proxy Server and DNS Server by using a password.** For this enter any type (1 or 2) in the first field and then enter the "close_the_server_1234" (The CLOSE_PASSWORD macro in server.c and proxy_server.c) in the second field.
- You can change the DNS Server IP Address and Port, and CLOSE_PASSWORD in the following macros in **proxy_server.c:**



```
//SERVER INFORMATION AND CLOSE PASSWORD FOR SERVER AND PROXY SERVER
#define CLOSE_PASSWORD "close_the_server_1234"
#define SERVER_IP "127.0.0.1"
#define SERVER_PORT 9100
```

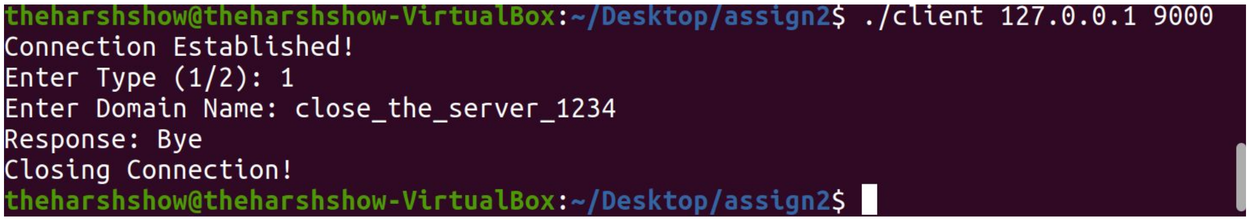
The proxy server is set up at any port of our choice. The client runs with the two arguments as specified in the question, i.e., proxy server IP and port number.
For checking the programs, we ran all the three hosts on the same computer, on three different terminals.
All input is given only on the client terminal.

After one request, the connection with the client will be terminated and the connection socket resources will be released. The reply will either be the requested information **(Type 3)** or the message that the data does not exist in the database **(Type 4)**:



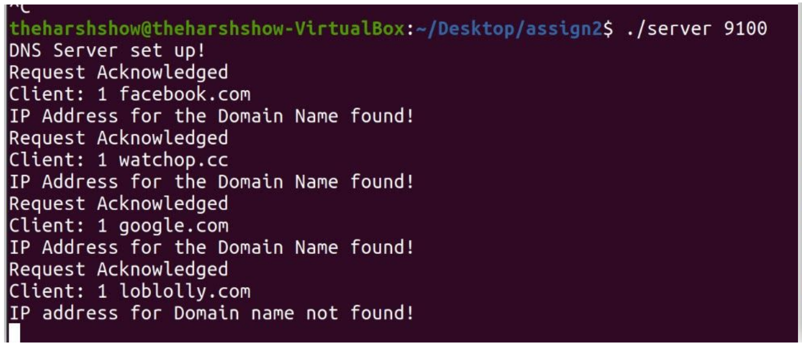
```
theharshshow@theharshshow-VirtualBox:~/Desktop/assign2$ ./client 127.0.0.1 9000
Connection Established!
Enter Type (1/2): 1
Enter Domain Name: google.com
Response: 3 172.217.167.206
Closing Connection!
theharshshow@theharshshow-VirtualBox:~/Desktop/assign2$ ./client 127.0.0.1 9000
Connection Established!
Enter Type (1/2): 1
Enter Domain Name: facebook.com
Response: 3 157.240.198.35
Closing Connection!
theharshshow@theharshshow-VirtualBox:~/Desktop/assign2$ ./client 127.0.0.1 9000
Connection Established!
Enter Type (1/2): 1
Enter Domain Name: loblolly.com
Response: 4 entry not found in the database
Closing Connection!
theharshshow@theharshshow-VirtualBox:~/Desktop/assign2$
```

The Proxy server and DNS server will keep running until a termination message is sent from a client, who knows the termination password (Refer to page 1):

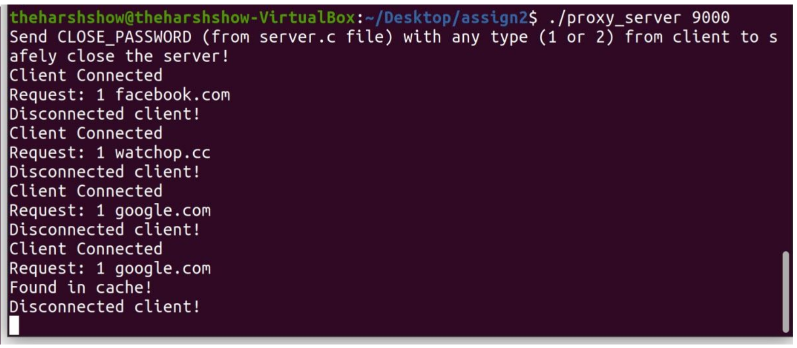


```
theharshshow@theharshshow-VirtualBox:~/Desktop/assign2$ ./client 127.0.0.1 9000
Connection Established!
Enter Type (1/2): 1
Enter Domain Name: close_the_server_1234
Response: Bye
Closing Connection!
theharshshow@theharshshow-VirtualBox:~/Desktop/assign2$
```

Proxy server and DNS Server logs can be seen in seen in their corresponding terminals:



```
^C
theharshshow@theharshshow-VirtualBox:~/Desktop/assign2$ ./server 9100
DNS Server set up!
Request Acknowledged
Client: 1 facebook.com
IP Address for the Domain Name found!
Request Acknowledged
Client: 1 watchop.cc
IP Address for the Domain Name found!
Request Acknowledged
Client: 1 google.com
IP Address for the Domain Name found!
Request Acknowledged
Client: 1 loblolly.com
IP address for Domain name not found!
```



```
theharshshow@theharshshow-VirtualBox:~/Desktop/assign2$ ./proxy_server 9000
Send CLOSE_PASSWORD (from server.c file) with any type (1 or 2) from client to s
afely close the server!
Client Connected
Request: 1 facebook.com
Disconnected client!
Client Connected
Request: 1 watchop.cc
Disconnected client!
Client Connected
Request: 1 google.com
Disconnected client!
Client Connected
Request: 1 google.com
Found in cache!
Disconnected client!
```

The cache is implemented using a circular linked list starting with a node with a pointer called cache (**struct node *cache**). Nodes are inserted at the back of the linked list. There can only be a maximum of three nodes in this linked list. When the cache is full, in order to insert the next element into the list, the start element is deleted and it's pointer moved forward (This is done systematically in the code). After this the new node is inserted.