EXPERIMENT - 9

Name	Anumeya Sehgal
Registration Number	23BAI1203
Course Code	BCSE308P
Course Title	Computer Networks Lab
Date	11 October, 2024

AIM: DNS Client & Server

SERVER PROGRAM:

```
#include <arpa/inet.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>

#define PORT 8080
#define BUF_SIZE 512

int main() {
  int sockfd;
  struct sockaddr_in server_addr;
  char buffer[BUF_SIZE];
  socklen_t addr_len = sizeof(server_addr);

  if ((sockfd = socket(AF_INET, SOCK_DGRAM, 0)) < 0) {
     perror("socket creation failed");
     exit(EXIT_FAILURE);
  }

  memset(&server_addr, 0, sizeof(server_addr));
```

```
  server_addr.sin_family = AF_INET;
  server_addr.sin_port = htons(PORT);
  server_addr.sin_addr.s_addr = INADDR_ANY;

  char dns_query[BUF_SIZE];
  printf("Enter domain to query: ");
  scanf("%s", dns_query);

  sendto(sockfd, dns_query, strlen(dns_query), 0,
         (const struct sockaddr *)&server_addr, addr_len);
  printf("DNS query sent: %s\n", dns_query);

  int recv_len = recvfrom(sockfd, buffer, BUF_SIZE, 0,
                          (struct sockaddr *)&server_addr,
&addr_len);
  if (recv_len < 0) {
     perror("recvfrom failed");
  } else {
     buffer[recv_len] = '\0';
     printf("DNS response received: %s\n", buffer);
  }

  close(sockfd);
  return 0;
}
```

SERVER PROGRAM:

```
#include <arpa/inet.h>
#include <netdb.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
```

```
#define PORT 8080
#define BUF_SIZE 512

void resolve_domain_to_ip(const char *domain, char *ip_buffer) {
  struct hostent *he;
  struct in_addr **addr_list;

  if ((he = gethostbyname(domain)) == NULL) {
     strcpy(ip_buffer, "DNS resolution failed");
     return;
  }

  addr_list = (struct in_addr **)he→h_addr_list;
  if (addr_list[0] ≠ NULL) {
     strcpy(ip_buffer, inet_ntoa(*addr_list[0]));
  } else {
     strcpy(ip_buffer, "No IP found");
  }
}

void handle_dns_query(int sockfd, struct sockaddr_in
*client_addr,
                      socklen_t addr_len) {
  char buffer[BUF_SIZE], response[BUF_SIZE];
  memset(response, 0, BUF_SIZE);
  int recv_len = recvfrom(sockfd, buffer, BUF_SIZE, 0,
                          (struct sockaddr *)client_addr,
&addr_len);
  if (recv_len < 0) {
     perror("recvfrom failed");
     return;
  }

  buffer[recv_len] = '\0';
  printf("Received DNS query for: %s\n", buffer);
```

```
  resolve_domain_to_ip(buffer, response);

  sendto(sockfd, response, strlen(response), 0, (struct sockaddr
*)client_addr,
         addr_len);
  printf("DNS response sent: %s\n", response);
}

int main() {
  int sockfd;
  struct sockaddr_in server_addr, client_addr;
  socklen_t addr_len = sizeof(client_addr);

  if ((sockfd = socket(AF_INET, SOCK_DGRAM, 0)) < 0) {
     perror("socket creation failed");
     exit(EXIT_FAILURE);
  }

  memset(&server_addr, 0, sizeof(server_addr));
  server_addr.sin_family = AF_INET;
  server_addr.sin_addr.s_addr = INADDR_ANY;
  server_addr.sin_port = htons(PORT);

  if (bind(sockfd, (const struct sockaddr *)&server_addr,
sizeof(server_addr)) <
     0) {
     perror("bind failed");
     close(sockfd);
     exit(EXIT_FAILURE);
  }

  printf("Enhanced DNS Server is running on port %d...\n",
PORT);
```

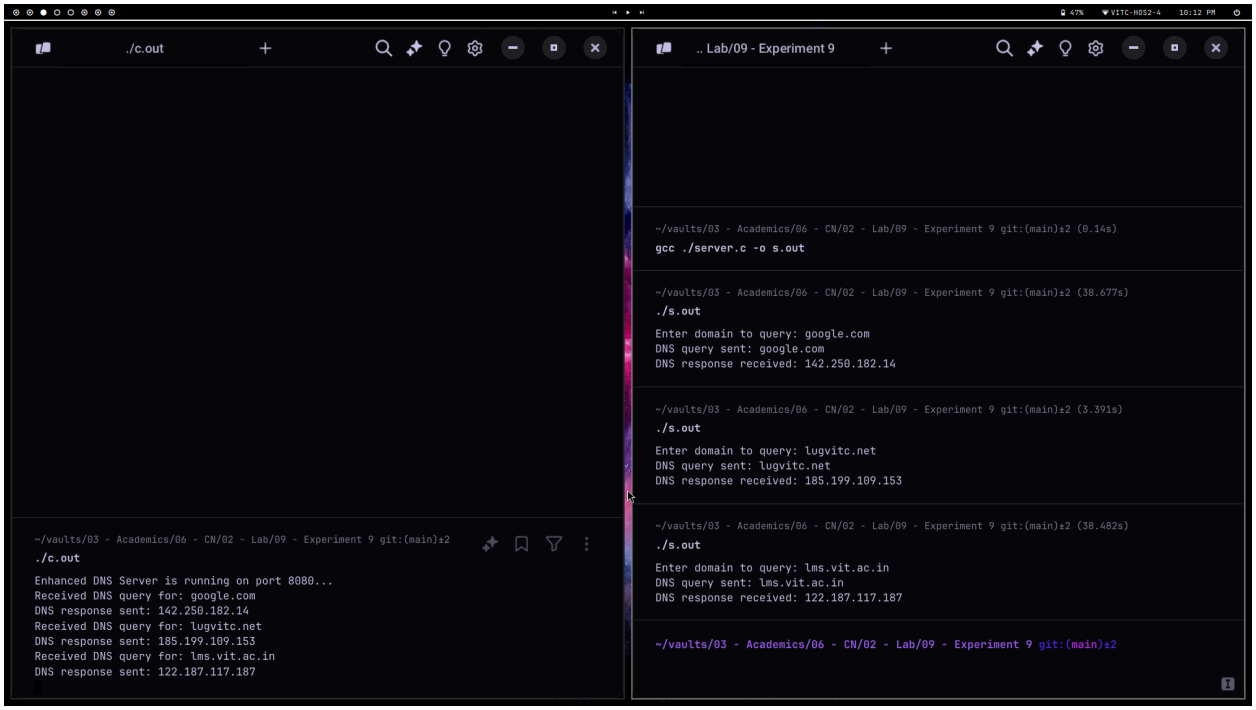
```
  while (1) {
     handle_dns_query(sockfd, &client_addr, addr_len);
  }

  close(sockfd);
  return 0;
}
```

OUTPUT:


.. Lab/09 - Experiment 9
~/vaults/03 - Academics/06 - CN/02 - Lab/09 - Experiment 9 git:(main)±2 (0.14s)
gcc ./server.c -o s.out
~/vaults/03 - Academics/06 - CN/02 - Lab/09 - Experiment 9 git:(main)±2 (38.677s)
./s.out
Enter domain to query: google.com
DNS query sent: google.com
DNS response received: 142.250.182.14
~/vaults/03 - Academics/06 - CN/02 - Lab/09 - Experiment 9 git:(main)±2 (3.391s)
./s.out
Enter domain to query: lugvitc.net
DNS query sent: lugvitc.net
DNS response received: 185.199.109.153
~/vaults/03 - Academics/06 - CN/02 - Lab/09 - Experiment 9 git:(main)±2 (3
./s.out
Enter domain to query: lms.vit.ac.in
DNS query sent: lms.vit.ac.in
DNS response received: 122.187.117.187
~/vaults/03 - Academics/06 - CN/02 - Lab/09 - Experiment 9 git:(main)±2
obsidian --help


./c.out
~/vaults/03 - Academics/06 - CN/02 - Lab/09 - Experiment 9 git:(main)±2
./c.out
Enhanced DNS Server is running on port 8080...
Received DNS query for: google.com
DNS response sent: 142.250.182.14
Received DNS query for: lugvitc.net
DNS response sent: 185.199.109.153
Received DNS query for: lms.vit.ac.in
DNS response sent: 122.187.117.187


./c.out
~/vaults/03 - Academics/06 - CN/02 - Lab/09 - Experiment 9 git:(main)±2
./c.out
Enhanced DNS Server is running on port 8080...
Received DNS query for: google.com
DNS response sent: 142.250.182.14
Received DNS query for: lugvitc.net
DNS response sent: 185.199.109.153
Received DNS query for: lms.vit.ac.in
DNS response sent: 122.187.117.187
.. Lab/09 - Experiment 9
~/vaults/03 - Academics/06 - CN/02 - Lab/09 - Experiment 9 git:(main)±2 (0.14s)
gcc ./server.c -o s.out
~/vaults/03 - Academics/06 - CN/02 - Lab/09 - Experiment 9 git:(main)±2 (38.677s)
./s.out
Enter domain to query: google.com
DNS query sent: google.com
DNS response received: 142.250.182.14
~/vaults/03 - Academics/06 - CN/02 - Lab/09 - Experiment 9 git:(main)±2 (3.391s)
./s.out
Enter domain to query: lugvitc.net
DNS query sent: lugvitc.net
DNS response received: 185.199.109.153
~/vaults/03 - Academics/06 - CN/02 - Lab/09 - Experiment 9 git:(main)±2 (38.482s)
./s.out
Enter domain to query: lms.vit.ac.in
DNS query sent: lms.vit.ac.in
DNS response received: 122.187.117.187
~/vaults/03 - Academics/06 - CN/02 - Lab/09 - Experiment 9 git:(main)±2