SSN College of Engineering, Kalavakkam

Department of Computer Science and Engineering

V Semester - CSE 'B'

UCS1511 NETWORKS LAB

Name: Likhitha Verma A
REG No: 185001084
Date: 22/09/2020

Exercise 5 : DOMAIN NAME SERVER USING UDP

Aim : To simulate the concept of Domain Name Server using UDP.

Learning objectives :

- To learn C socket programming.
- To learn how to create a UDP client server connection.
- To learn how application programs use protocol software to communicate across networks and internets.
- To learn about various system calls used in socket programming using UDP.

Algorithm :

SERVER SIDE:

1. Create a socket descriptor with **socket()** system call with AF_INET, SOCK DGRAM, default protocol and store as sockfd.
2. If sockfd returns a negative value, print error message.
3. Create sockaddr_in object and initialize with values.
4. Bind newly created socket to address given in sockaddr _in using **bind()** system call.

5. If **bind()** returns negative value, bind failed, print error message.
6. Create a structure table that has domain name and its IP addresses as members.
7. Create entries in the table and also validate the given IP addresses i.e if IP already exists in the table or IP is invalid, print appropriate messages.
8. Display the table.
9. Read the domain name requested by the client using **recvfrom()** system call.
10. Get the IP address of the requested domain name from the table and send the response using **sendto()** system call.
11. If the server wants to modify the table entries, get domain name and IP address, validate the given details and then modify the table contents.
12. Close connections on socket using **close()** and terminate program.

CLIENT SIDE:

1. Create a socket descriptor with **socket()** system call with AF_INET, SOCK DGRAM, default protocol and store as sockfd.
2. If sockfd returns a negative value, print error message.
3. Create sockaddr_in object and initialize with values.
4. Get domain name as input and send it to server using **sendto()** system call.
5. Get response from the server using **recvfrom()** system call and display.
6. Continue Step 4 and 5 until the input is '*'.
7. Close connections on socket using **close()** and terminate program.

Program:

Contents of dnsFunctions.h File :

```
#define MAX_ADDR 4
#define MAX_DOMAIN 20
typedef char string[30];

typedef struct table_row
{
    string domain;
    string address[MAX_ADDR];
}entry;


int is_IpTaken(entry table[MAX_DOMAIN], char * address){
     for(int i=0;i<MAX_DOMAIN;++i){

           for (int j = 0; j < MAX_ADDR; ++j)
           {
                if(table[i].address[j][0] &&
strcmp(table[i].address[j],address)==0){
                     printf("IP address already taken\n");
                     return 1;
                }
           }
     }
     return 0;
}

int is_IpInvalid(char * address){
     string temp;
     strcpy(temp,address);

    char *split;
    int val;
    split = strtok(temp, ".");
    while (split)
    {
        val = atoi(split);
```

```
        if (val < 0 || val > 255){
            printf("Invalid IP address\n");
        return 1;
        }
        split = strtok(NULL, ".");
    }
    return 0;
}
int createEntry(entry table[MAX_DOMAIN], char *domain, char *address)
{

    int index = -1;
    int flag = 0;

    if(is_IpTaken(table,address) == 1 || is_IpInvalid(address) == 1)
    {
      return 2;
    }

    for (int i = 0; i < MAX_DOMAIN; i++)
    {
        if (strcmp(table[i].domain, domain) == 0)
        {
            for (int j = 0; j < MAX_ADDR; j++)
                if (!table[i].address[j][0])
                {
                    strcpy(table[i].address[j], address);
                    flag = 1;
                    break;
                }
            break;
        }
        if (!table[i].domain[0] && index == -1)
            index = i;
    }

    if (!flag)
    {
        strcpy(table[index].domain, domain);
        strcpy(table[index].address[0], address);
```

```
        flag = 1;
    }
    return flag;
}

entry *getAddress(entry *table, char *const domain)
{
    static entry result;
    bzero(&result, sizeof(entry));
    strcpy(result.domain, domain);

    for (int i = 0; i < MAX_DOMAIN; i++)
    {
        if (strcmp(table[i].domain, domain) == 0)
        {
            for (int j = 0; j < MAX_ADDR; j++)
            {
                strcpy(result.address[j], table[i].address[j]);
            }
            break;
        }
    }
    return (&result);
}
```

SERVER SIDE:

```
#include<stdio.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include<string.h>
#include <unistd.h> //for read, write and close
#include <stdlib.h>
#include"dnsFunctions.h"

void print_table(entry table[MAX_DOMAIN]){
     printf("\t\tDomain Name\t\tAddress\n");
```

```
      printf("\t\t***************************************\n");
      for (int i = 0; i < MAX_DOMAIN; i++)
      {
            if (table[i].domain[0])
            {
                  printf("\t\t%-15s \t %-15s\n", table[i].domain,
table[i].address[0]);
                  for (int j = 1; j < MAX_ADDR &&
table[i].address[j][0]; j++)
                        printf("\t\t%-15s \t %-15s\n", " ",
table[i].address[j]);
            }
      }
      printf("\n");
}
int main()
{

      int sockfd,n;
      struct sockaddr_in seraddr, cli;
      char buff[100],op;
      socklen_t len;

      entry table[MAX_DOMAIN],*result;
      bzero(table,MAX_DOMAIN * sizeof(entry));

      //Socket Creation
      sockfd=socket(AF_INET,SOCK_DGRAM,0);
      if(sockfd<0)
            perror("Socket Creation failed...!");
      else
            printf("Socket Created Successfully...!\n");

      bzero(&seraddr,sizeof(seraddr));

      seraddr.sin_family=AF_INET;
      seraddr.sin_port=htons(3335);
      seraddr.sin_addr.s_addr=htonl(INADDR_ANY);

      //binding to the socket
```

```
     if(bind(sockfd,(struct sockaddr *) &seraddr,
sizeof(seraddr))<0)
          perror("Bind failed...!");
     else
          printf("Bound Successfully...!\n");

     len=sizeof(cli);

     createEntry(table, "www.yahoo.com", "10.2.45.67");
     createEntry(table, "www.annauniv.edu", "197.34.53.122");
     createEntry(table, "www.google.com", "142.89.78.66");

     print_table(table);

     while (1)
     {
          int t=0;
          recvfrom(sockfd, buff, sizeof(buff), 0, (struct sockaddr
*)&cli, &len);
          result = getAddress(table, buff);
          sendto(sockfd, result, sizeof(entry), 0, (struct sockaddr
*)&cli, len);


          printf("Do you want to modify the table (y/n) ?");
          scanf(" %c",&op);

          if(op=='y'){
               string domain, address;
               printf("Enter the domain name: ");
               scanf(" %s",domain);
               do{
                    printf("Enter the IP address: ");
                    scanf(" %s",address);


                    createEntry(table,domain,address);
               }while(t==2);

               print_table(table);
```

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

CLIENT SIDE:

```
#include<stdio.h>
#include<sys/types.h>
#include<sys/socket.h>
#include<netinet/in.h>
#include<string.h>
#include <unistd.h> //for read write and close
#include <arpa/inet.h> //for inet_addr()

#define MAX_ADDR 4
#define MAX_DOMAIN 20
typedef char string[30];
typedef struct table_row
{
      string domain;
      string address[MAX_ADDR];
}entry;

int main()
{
      entry request;
      int sockfd, connfd;
      struct sockaddr_in servaddr, cli;
      char buff[30] = {0};

      sockfd = socket(AF_INET, SOCK_DGRAM, 0);
      if(sockfd<0)
            perror("Socket Creation failed...!");
      else
            printf("Socket Created Successfully...!\n");
```

```
	bzero(&servaddr, sizeof(servaddr));
	// assign IP, PORT
	servaddr.sin_family = AF_INET;
	servaddr.sin_addr.s_addr = inet_addr("127.0.0.1");
	servaddr.sin_port = htons(3335);
	socklen_t len = sizeof(entry);
	while(1)
	{
		bzero(&request, sizeof(entry));
		printf("Enter the domain name: ");
		scanf(" %[^\n]", request.domain);

		if (strcmp(request.domain, "*") == 0)
			break;

		sendto(sockfd, request.domain, sizeof(request.domain), 0,
(struct sockaddr *)&servaddr, sizeof(servaddr));
		recvfrom(sockfd, &request, sizeof(entry), 0, (struct
sockaddr *)&servaddr, &len);

		if (!request.address[0][0])
			printf("No such entry in the DNS..!!\n");
		else
		{
			printf("The IP Address is: \n");
			for (int i = 0; i < MAX_ADDR; i++)
			{
				if (request.address[i][0])
					printf("%s\n", request.address[i]);
			}
			printf("\n");
		}
	}
	close(sockfd);
}
```

OUTPUT : SERVER SIDE :

```
msml@MSMLs-MacBook-Pro 5-ex % gcc server.c -o s
msml@MSMLs-MacBook-Pro 5-ex % ./s
Socket Created Successfully...!
Bound Successfully...!
                Domain Name             Address
                ****************************************
                www.yahoo.com            10.2.45.67
                www.annauniv.edu         197.34.53.122
                www.google.com           142.89.78.66

Do you want to modify the table (y/n) ?y
Enter the domain name: www.yahoo.com
Enter the IP address: 196.34.53.122
                Domain Name             Address
                ****************************************
                www.yahoo.com            10.2.45.67
                                         196.34.53.122
                www.annauniv.edu         197.34.53.122
                www.google.com           142.89.78.66

Do you want to modify the table (y/n) ?n
Do you want to modify the table (y/n) ?y
Enter the domain name: www.gmail.com
Enter the IP address: 300.8.35.79
Invalid IP address
                Domain Name             Address
                ****************************************
                www.yahoo.com            10.2.45.67
                                         196.34.53.122
                www.annauniv.edu         197.34.53.122
                www.google.com           142.89.78.66

Do you want to modify the table (y/n) ?y
Enter the domain name: www.gmail.com
Enter the IP address: 197.34.53.122
IP address already taken
                Domain Name             Address
                ****************************************
                www.yahoo.com            10.2.45.67
                                         196.34.53.122
                www.annauniv.edu         197.34.53.122
                www.google.com           142.89.78.66

Do you want to modify the table (y/n) ?y
Enter the domain name: www.gmail.com
Enter the IP address: 190.23.45.67
                Domain Name             Address
                ****************************************
                www.yahoo.com            10.2.45.67
```

```
Enter the domain name: www.gmail.com
Enter the IP address: 190.23.45.67
               Domain Name             Address
               ***************************************
               www.yahoo.com            10.2.45.67
                                        196.34.53.122
               www.annauniv.edu         197.34.53.122
               www.google.com           142.89.78.66
               www.gmail.com            190.23.45.67

Do you want to modify the table (y/n) ?n
^C
msml@MSMLs-MacBook-Pro 5-ex %
```

CLIENT SIDE 1 :

```
msml@MSMLs-MacBook-Pro 5-ex % gcc client.c -o c1
msml@MSMLs-MacBook-Pro 5-ex % ./c1
Socket Created Successfully...!
Enter the domain name: www.yahoo.com
The IP Address is:
10.2.45.67

Enter the domain name: www.yahoo.com
The IP Address is:
10.2.45.67
196.34.53.122

Enter the domain name: www.annauniv.edu
The IP Address is:
197.34.53.122

Enter the domain name: *
msml@MSMLs-MacBook-Pro 5-ex %
```

CLIENT SIDE 2 :

```
msml@MSMLs-MacBook-Pro 5-ex % gcc client.c -o c2
msml@MSMLs-MacBook-Pro 5-ex % ./c2
Socket Created Successfully...!
Enter the domain name: www.gmail.com
No such entry in the DNS..!!
Enter the domain name: www.google.com
The IP Address is:
142.89.78.66

Enter the domain name: www.gmail.com
The IP Address is:
190.23.45.67

Enter the domain name: *
msml@MSMLs-MacBook-Pro 5-ex %
```

Learning Outcomes:

- I have learnt to create socket programming using UDP.
- I have learnt to validate IP addresses.
- I have learnt to work with connection-less protocol.
- I have learnt to maintain a table for performing DNS functions.