Ex: 8

Regno : 185001161

**DOMAIN NAME SERVER USING UDP**

Simulate the concept of Domain Name Server using UDP.

Server should perform the following:

1.Maintain a table which contains IP address and the corresponding Server name and display

the table.

2.When a request is for an IP address (Given a server name) is from a client is received, verify

the table and send the corresponding IP address to the server.

3. Make server to accept multiple client request simultaneously.

4. Also modify the server.

Client should do the following:

1. Request for an IP address is given to the server using the server name

2. Receive the corresponding IP address and display it.

**SERVER CODE:**

#include<stdio.h>

#include<stdlib.h>

#include<ctype.h>

#include<netdb.h>

#include<unistd.h>

#include<sys/types.h>

#include<arpa/inet.h>

#include<string.h>

#include<netinet/in.h>

#include<sys/socket.h>

#define PORT 7228

struct table

{

char ip[100];

char server[100];

};

int ip\_validity(char ip[100])

{

int i=0;

char token[4];

int k=0,dot=0;

int j=0;

int flag=1;

int num;

while(ip[i]!='\0')

{

if(ip[i]=='.' || ip[i+1]=='\0')

{

if (ip[i+1]=='\0')

{

token[k]=ip[i];

k++;

i++;

dot--;

}

if (k>3)

{

flag=0;

break;

}

k=0;

dot++;

if (dot == 4)

{

flag=0;

break;

}

// Checking for the characters other than number...

for(j=0;j<strlen(token);j++)

{

if (isdigit(token[j])==0)

{

flag=0;

}

}

if (flag==0)

break;

// Checking whether it is in the range...

num = atoi(token);

if (num < 0 || num > 255)

{

flag=0;

break;

}

bzero(&token,sizeof(token));

}

else

{

token[k]=ip[i];

k++;

}

i++;

}

return flag;

}

int check\_domain(char dom[],int n,struct table t[])

{

int i;

int flag=-1;

for(i=0;i<n;i++)

{

if(strcmp(dom,t[i].server)==0)

{

flag=i;

break;

}

}

return flag;

}

int check\_ip(char ip[],int n,struct table t[])

{

int i;

int flag=1;

for(i=0;i<n;i++)

{

if(strcmp(ip,t[i].ip)==0)

{

flag=0;

break;

}

}

return flag;

}

void print\_table(int n,struct table t[])

{

int i;

printf("\n\tDomain Name\t\t\tIp Adrress\n");

for(i=0;i<n;++i)

{

printf("\t%-20s\t%s\n",t[i].server,t[i].ip);

}

}

struct table\* modify(int n,struct table t[])

{

int indx;

char domain[100];

char ipaddr[100];

a:

printf("\nEnter Domain name:");

scanf("%s",domain);

indx=check\_domain(domain,n,t);

if (indx !=-1)

{

b:

printf("\nEnter ip address:");

scanf("%s",ipaddr);

if (ip\_validity(ipaddr))

{

if (check\_ip(ipaddr,n,t))

{

strcpy(t[indx].ip,ipaddr);

printf("\nModified Table is\n");

print\_table(n,t);

}

else

{printf("\nIP address already exist\n");goto b;}

}

else

{printf("\nInvalid Ip address\n");goto b;}

}

else

{printf("\nDomain Name doesn't exist\n");goto a;}

return t;

}

struct table \*create\_table(int n)

{

int i;

char test[100];

strcpy(test,"Automatically generated for testing !");

//printf("\nEnter the no.of server names to be Added: %s",test);

//scanf("%d",&n);

struct table \*t;

t = (struct table\*)malloc(sizeof(struct table)\*n);

/\*for(i=0;i<n;++i)

{

printf("\nEnter Domain name:");

scanf("%s",t[i].server);

printf("\nEnter Ip address :");

scanf("%s",t[i].ip);

printf("\n");

}\*/

strcpy(t[0].server,"www.yahoo.com");

strcpy(t[0].ip,"10.2.45.67");

strcpy(t[1].server,"www.annauniv.edu");

strcpy(t[1].ip,"197.34.53.122");

strcpy(t[2].server,"www.google.com");

strcpy(t[2].ip,"142.89.78.66");

print\_table(n,t);

return t;

}

char \*get\_ip(char buff[],struct table t[],int n)

{

int i;

for(i=0;i<n;i++)

{

if(strcmp(buff,t[i].server)==0)

{

return t[i].ip;

}

}

}

int main(int argc,char \*\*argv)

{

int sockfd,ret,no,len;

struct sockaddr\_in serverAddr;

int newfd;

struct sockaddr\_in clientAddr;

char buffer[1024];

char str[1000];

sockfd = socket(AF\_INET,SOCK\_DGRAM,0);

if (sockfd < 0)

{

printf("\n Error in Connection\n");

}

printf("\nServer Socket is created\n");

bzero(&serverAddr,sizeof(serverAddr));

serverAddr.sin\_addr.s\_addr=INADDR\_ANY;

serverAddr.sin\_family=AF\_INET;

serverAddr.sin\_port=htons(PORT);

ret = bind(sockfd,(struct sockaddr\*)&serverAddr,sizeof(serverAddr));

int n;

int x;

char test[100];

strcpy(test,"Automatically generated for testing !");

printf("\nEnter the no.of server names to be Added: %s",test);

//scanf("%d",&n);

n=3;

struct table \*t;

t = create\_table(n);

while(1)

{

printf("\n1.Modify\n2.Exit \n Enter choice:");

scanf("%d",&x);

if (x==1)

{

t=modify(n,t);

}

else if(x==2)

{

break;

}

}

printf("\nListening ....\n");

len = sizeof(clientAddr);

while(1)

{

no = recvfrom(sockfd,buffer,sizeof(buffer),0,(struct sockaddr\*)&clientAddr,&len);

strcpy(buffer,get\_ip(buffer,t,n));

sendto(sockfd,buffer,sizeof(buffer),0,(struct sockaddr \*)&clientAddr,len);

}

return 0;

}

/\*Output:

sree@sree-VirtualBox:~/Desktop/Networks Lab/A7 DNS$ gcc server.c -o server

sree@sree-VirtualBox:~/Desktop/Networks Lab/A7 DNS$ ./server

Server Socket is created

Enter the no.of server names to be Added: Automatically generated for testing !

Domain Name Ip Adrress

www.yahoo.com 10.2.45.67

www.annauniv.edu 197.34.53.122

www.google.com 142.89.78.66

1.Modify

2.Exit

Enter choice:2

Listening ....

^C

sree@sree-VirtualBox:~/Desktop/Networks Lab/A7 DNS:\*/

**CLIENT CODE:**

#include<stdio.h>

#include<stdlib.h>

#include<ctype.h>

#include<netdb.h>

#include<unistd.h>

#include<sys/types.h>

#include<arpa/inet.h>

#include<string.h>

#include<netinet/in.h>

#include<sys/socket.h>

#define PORT 7228

int main(int argc,char \*\*argv)

{

int sockfd,ret,len,n;

struct sockaddr\_in serverAddr;

char buffer[1024];

char str[1000];

sockfd = socket(AF\_INET,SOCK\_DGRAM,0);

if(sockfd < 0)

{

printf("-----Error in Connection-----.\n");

exit(1);

}

//printf("-----Client Socket is Created!.\n");

bzero(&serverAddr,sizeof(serverAddr));

serverAddr.sin\_family = AF\_INET;

serverAddr.sin\_addr.s\_addr = inet\_addr("127.0.0.1");

serverAddr.sin\_port = htons(7228);

len = sizeof(serverAddr);

int x;

char domain[100];

while(1)

{

printf("\n\n1.Request\n2.Exit \n Enter choice:");

scanf("%d",&x);

if (x==1)

{

printf("\nEnter the domain name:");

scanf("%s",buffer);

sendto(sockfd,buffer,sizeof(buffer),0,(struct sockaddr \*)&serverAddr,len);

recvfrom(sockfd,buffer,sizeof(buffer),0,(struct sockaddr\*)&serverAddr,&len);

printf("\nIP address :%s",buffer);

}

else if(x==2)

{

break;

}

}

}

/\*Output:

sree@sree-VirtualBox:~/Desktop/Networks Lab/A7 DNS$ ./client

1.Request

2.Exit

Enter choice:1

Enter the domain name:www.yahoo.com

IP address :10.2.45.67

1.Request

2.Exit

Enter choice:2

client :2

sree@sree-VirtualBox:~/Desktop/Networks Lab/A7 DNS$ ./client

1.Request

2.Exit

Enter choice:1

Enter the domain name:www.google.com

IP address :142.89.78.66

1.Request

2.Exit

Enter choice:2\*/