**PROGRAM CODE:**

**CLIENT:**

#include<stdio.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<stdlib.h>

#include<string.h>

#include<unistd.h>

#include<arpa/inet.h>

#include<sys/types.h>

#include <fcntl.h>

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

{

struct sockaddr\_in server,client;

char fname[300],buff[1024],str[100]="\0";

int newfd1,i,j;

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

if(sfd<0)

{

perror("Cannot create socket");

exit(0);

}

bzero(&server,sizeof(server));

server.sin\_family=AF\_INET;

server.sin\_port=htons(4000);

server.sin\_addr.s\_addr=inet\_addr(argv[1]);

while(1)

{

int len=sizeof(server);

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

gets(str);

int n=sendto(sfd,str,sizeof(str),MSG\_CONFIRM,(struct sockaddr\*)&server,len);

sleep(3);

n=recvfrom(sfd,str,sizeof(str),MSG\_WAITALL,(struct sockaddr\*)&server,&len);

str[n]='\0';

printf("\nThe IP address is: %s\n",str);

}

close(sfd);

return 0;

}

**SERVER:**

#include<stdio.h>

#include<sys/types.h>

#include<sys/socket.h>

#include<netinet/in.h>

#include<stdlib.h>

#include<string.h>

#include<unistd.h>

#include<arpa/inet.h>

#include<sys/types.h>

#include <fcntl.h>

typedef struct{

char server\_name[100];

char ip\_addr[3][100];

}dns;

int main()

{

struct sockaddr\_in server,client;

char choice[100],d\_name[100],ip\_add[100],ip2[100];

char str[100]="\0",new\_str[100];

int newfd1,i=0;

int j,num,k,x,y;

dns table[5];

for(j=0;j<5;j++)

for(k=0;k<3;k++)

strcpy(table[j].ip\_addr[k],"");

strcpy(table[0].server\_name,"www.yahoo.com");

strcpy(table[0].ip\_addr[0],"10.2.45.67");

strcpy(table[1].server\_name,"www.google.com");

strcpy(table[1].ip\_addr[0],"142.89.78.66");

strcpy(table[2].server\_name,"www.annauniv.edu");

strcpy(table[2].ip\_addr[0],"197.34.53.122");

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

if(sfd<0)

{

perror("Cannot create socket");

exit(0);

}

bzero(&server,sizeof(server));

server.sin\_family=AF\_INET;

server.sin\_port=htons(4000);

server.sin\_addr.s\_addr=htonl(INADDR\_ANY);

int bs=bind(sfd,(struct sockaddr\*)&server,sizeof(server));

if(bs<0)

{

perror("Bind error");

exit(1);

}

printf("\nDo you want to modify (yes or no): ");

scanf("%s",choice);

if(strcmp(choice,"yes")==0)

{

getchar();

printf("\nEnter Domain name: ");

gets(d\_name);

for(j=0;j<5;j++)

if(strcmp(table[j].server\_name,d\_name)==0) break;

if(j==5)

{

printf("Invalid domain name\n");

printf("\nEnter domain name: ");

gets(d\_name);

}

printf("Enter IP address: ");

gets(ip\_add); strcpy(ip2,ip\_add);

char\* token=strtok(ip2,".");

while(token!=NULL)

{

num=atoi(token);

if(num>255)

{

printf("Invalid IP address");

printf("\nEnter a valid IP address: ");

gets(ip\_add);

break;

}

token=strtok(NULL,".");

}

for(k=0;k<5;k++)

{

if( (strcmp(table[k].server\_name,d\_name)==0) && (strcmp(table[k].ip\_addr[0],ip\_add)==0) )

{

printf("IP address already exists");

printf("\nEnter valid IP address: ");

gets(ip\_add);

break;

}

}

strcpy(table[j].ip\_addr[1],ip\_add);

printf("\nSucessfully modified");

printf("\nThe updated table is:\n");

printf("\nServer name\t IP address");

for(x=0;x<3;x++)

{

printf("\n%s\t",table[x].server\_name);

for(y=0;y<2;y++)

printf("%s\n\t\t",table[x].ip\_addr[y]);

}

}//end if yes

while(1)

{

int len=sizeof(client);

int n=recvfrom(sfd,str,sizeof(str),MSG\_WAITALL,(struct sockaddr\*)&client,&len);

str[n]='\0';

for(j=0;j<5;j++)

{

if(strcmp(table[j].server\_name,str)==0)

{

k=0;

strcpy(str,""); strcpy(new\_str,"");

while(strcmp(table[j].ip\_addr[k],"")!=0 && k<3)

{

strcpy(new\_str, table[j].ip\_addr[k]);

if(strcmp(table[j].ip\_addr[k+1],"")!=0 && k<2)

strcat(new\_str," and ");

strcat(str,new\_str);

k++;

}

i=1;

break;

}

}

if(i==0)

strcpy(str,"No such server exists");

n=sendto(sfd,str,sizeof(str),MSG\_CONFIRM,(struct sockaddr\*)&client,len);

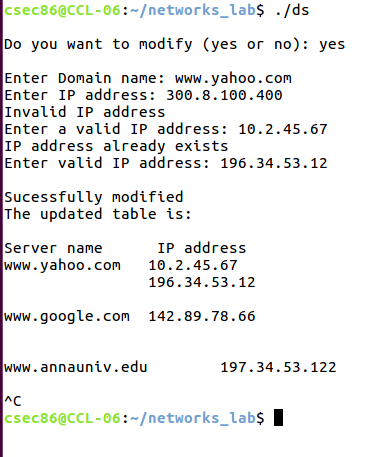
}

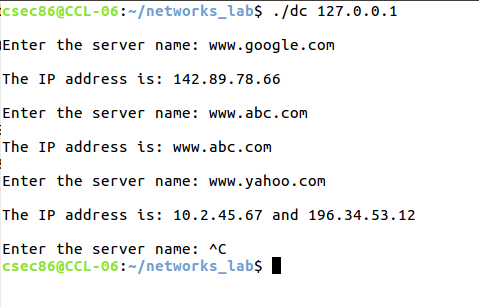
close(sfd);

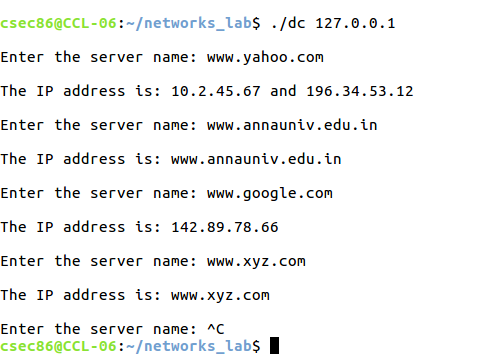
return 0;

}

**OUTPUT:**

****

****

****