**DISTRIBUTED COMPONENTS LABORATORY**

**CA1**

**Aim:**

To develop a distributed application using SUN RPC that implements telephone directory operation.

**Algorithm:**

**Step 1:** Start

**Step 2:** The file with .x extension is created.

**Step 3:** Using the file, rpcgen command is executed and more files are generated that are required for the application.

**Step 4:** The server and client code are modified according to the given application.

**Step 5:** make command is used to compile the code and generate the required object files.

**Step 6:** The client and server code are run in parallel terminals.

**Step 7:** Stop

**Program:**

**phone.x file:**

struct customer {

    char name[100];

    char phone\_number[10];

    char address[100];

};

struct name {

    char str[100];

};

program phone {

    version PHONE\_VERS {

        customer search(name)=1;

    }=1;

}=0x23451111;

**Phone\_server.c:**

/\*

 \* This is sample code generated by rpcgen.

 \* These are only templates and you can use them

 \* as a guideline for developing your own functions.

 \*/

#include "phone.h"

#define N 4

customer \*

search\_1\_svc(name \*argp, struct svc\_req \*rqstp)

{

    static customer  result;

    struct customer c[10];

    strcpy(c[0].name,"Vishnu");

    strcpy(c[0].phone\_number,"7418529631");

    strcpy(c[0].address,"31/7A SV Layout");

    strcpy(c[1].name,"Balaji");

    strcpy(c[1].phone\_number,"7894561231");

    strcpy(c[1].address,"31/8A VV Layout");

    strcpy(c[2].name,"Mani");

    strcpy(c[2].phone\_number,"9874563211");

    strcpy(c[2].address,"31/9A WW Layout");

    strcpy(c[3].name,"Raghul");

    strcpy(c[3].phone\_number,"7896523141");

    strcpy(c[3].address,"31/4A XX Layout");

    strcpy(c[4].name,"Sairam");

    strcpy(c[4].phone\_number,"7412589631");

    strcpy(c[4].address,"31/6A YY Layout");

    int index = -1;

    for(int i=0;i<N;i++) {

        if(strcmp(argp->str,c[i].name)==0) {

            index = i;

            break;

        }

    }

    if(index!=-1) {

        strcpy(result.name,c[index].name);

        strcpy(result.phone\_number,c[index].phone\_number);

        strcpy(result.address,c[index].address);

    }

    else {

        strcpy(result.name,"NA");

        strcpy(result.phone\_number,"NA");

        strcpy(result.address,"NA");

    }

    return &result;

}

**Phone\_client.c**

/\*

 \* This is sample code generated by rpcgen.

 \* These are only templates and you can use them

 \* as a guideline for developing your own functions.

 \*/

#include "phone.h"

void

phone\_1(char \*host)

{

    CLIENT \*clnt;

    customer  \*result\_1;

    name  search\_1\_arg;

#ifndef DEBUG

    clnt = clnt\_create (host, phone, PHONE\_VERS, "udp");

    if (clnt == NULL) {

        clnt\_pcreateerror (host);

        exit (1);

    }

#endif  /\* DEBUG \*/

    char buffer[100][100];

    int counter[100] = {0};

    int index = 0;

    while(361) {

        int choice;

        printf("1.Search phone number\n2.Exit\n");

        scanf("%d",&choice);

        if(choice==1) {

            printf("Enter customer name: ");

            char cname[100];

            scanf("%s",cname);

            int location = -1;

            for(int i=0;i<index;i++) {

                if(strcmp(buffer[i],cname)==0) {

                    location = i;

                    break;

                }

            }

            if(location==-1) {

                strcpy(search\_1\_arg.str,cname);

                result\_1 = search\_1(&search\_1\_arg, clnt);

                if (result\_1 == (customer \*) NULL) {

                    clnt\_perror (clnt, "call failed");

                }

                else {

                    if(strcmp(result\_1->name,"NA")==0) {

                        printf("Data not available\n");

                    }

                    else {

                        printf("The data retrieved from server is:\n");

                        strcpy(buffer[index++],result\_1->name);

                    }

                    printf("Customer name: %s\n",result\_1->name);

                    printf("Customer phone number: %s\n",result\_1->phone\_number);

                    printf("Customer address: %s\n",result\_1->address);

                }

            }

            else {

                printf("Details alread fetched!\n");

                printf("Number of times searched: %d\n",++counter[location] + 1);

            }

        }

        else if(choice==2) {

            printf("Bye! Thanks for using our service!\n");

            break;

        }

    }

#ifndef DEBUG

    clnt\_destroy (clnt);

#endif   /\* DEBUG \*/

}

int

main (int argc, char \*argv[])

{

    char \*host;

    if (argc < 2) {

        printf ("usage: %s server\_host\n", argv[0]);

        exit (1);

    }

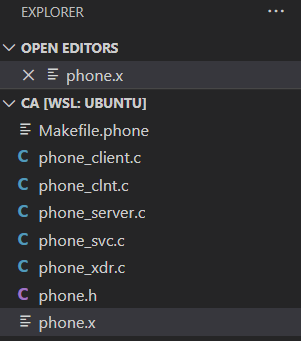
    host = argv[1];

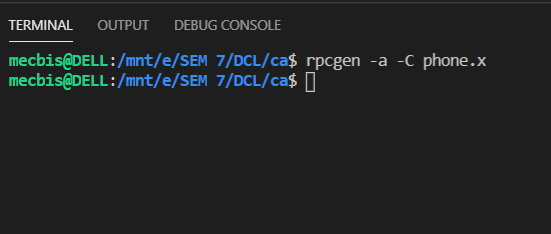
    phone\_1 (host);

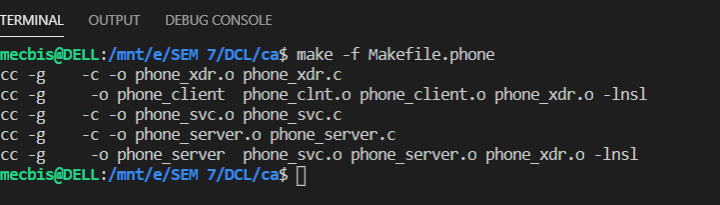
exit (0);

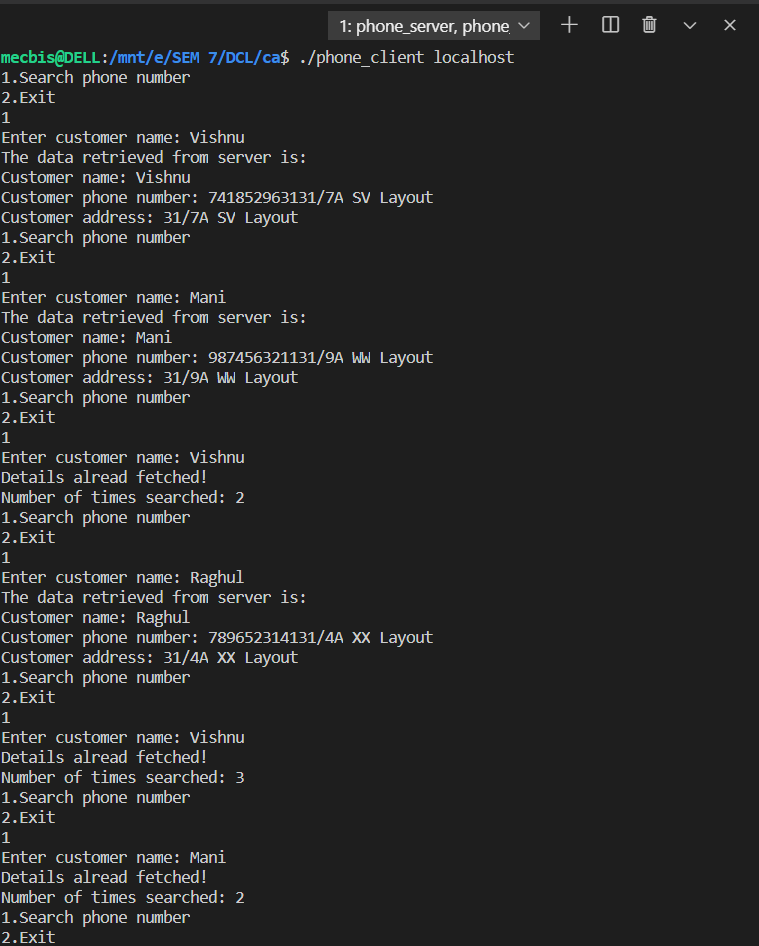
}

**Output:**



s





**Result:**

The given application is implemented and the output is obtained successfully.