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;
```

Vishnuvardhan S
17I361

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

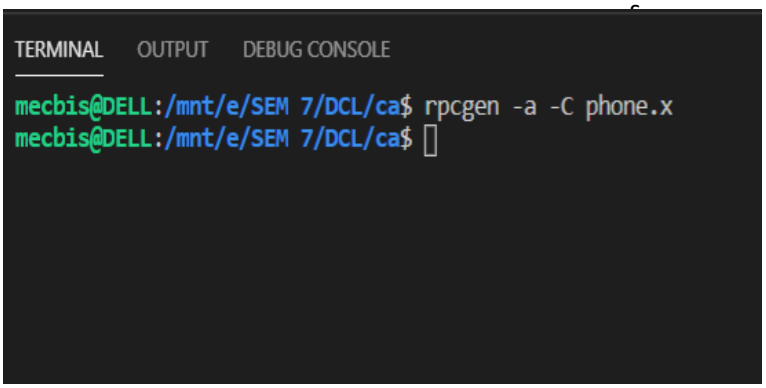
```
/*
 * This is sample code generated by rpcgen.
 * These are only templates and you can use them
 * as a guideline for developing your own functions.
 */

#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;
```

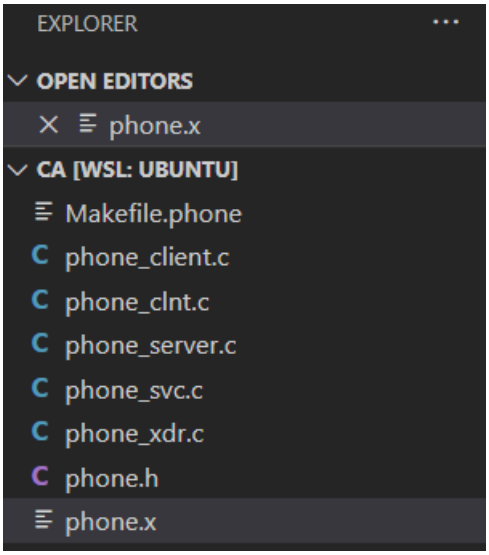
```
                    break;
                }
            }
            if(location==-1) {
                strcpy(search_1_arg.str,cname);
                result_1 = search_1(&search_1_arg, clnt);
                if (result_1 == (customer *) NULL) {
                    clnt_perror (clnt, "call failed");
                }
                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]);
```

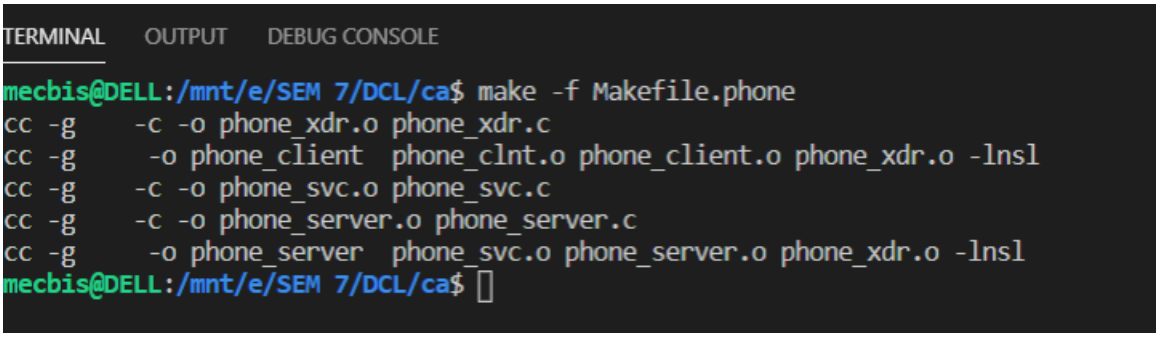
Vishnuvardhan S
17I361

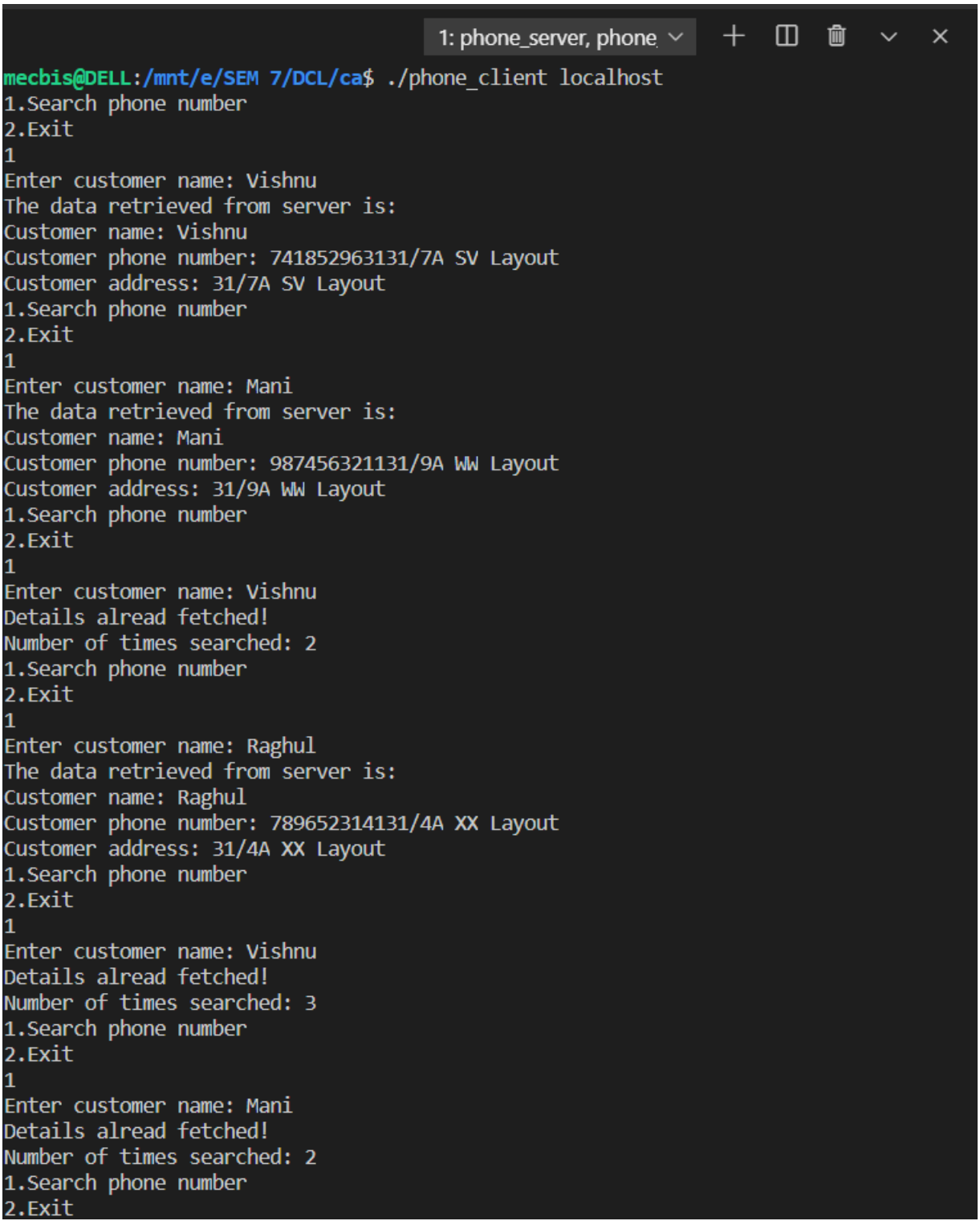
```
        exit (1);
    }
    host = argv[1];
    phone_1 (host);
exit (0);
}
```

Output:









Result:

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

Vishnuvardhan S
17I361