UDP Server- Client

Explanation:

In the UDP Server Client program , the following are the main steps that take place in :

In the UDP Server the following steps are performed:

1. A UDP socket is created using create().
2. Then this socket is bind to the server address using bind(). Bind() operation is usually assigning a name to a socket
3. Then it wait until for the datagram packet to arrive from the client socket.
4. Server then process the datagram packet and then sends a reply to the client
5. Then it goes back to the third step and again wait for the next datagram packet to arrive
6.

In the UDP Client the following steps are performed:

1. Here also , a UPP socket is created using create()
2. Client Socket then send a message to the server
3. Then it waits for the server response
4. It then replies to the server and if required, it goes to back to the step two
5. Then socket descriptor is closed and the client exit.

UDP -Client Server Code with Documentation

server.c

```
/*
   UDP-Server
*/
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
```

```
#include<arpa/inet.h>
#include<sys/socket.h>

#define BUFLEN 512  //Max length of buffer
#define PORT 8888   //The port to listen for incoming data

void die(char *sockfd)
{

        perror(sockfd);
    exit(1);
}

int main(void)
{

    struct sockaddr_in si_me, si_other;
    float op1;
    float op2;
    char operator;
    int sockfd, i, slen = sizeof(si_other) , recv_len;
    char buf[BUFLEN];
    int result;
        char res[BUFLEN];

    //create a UDP socket
```

```
    if ((sockfd=socket(AF_INET, SOCK_DGRAM, IPPROTO_UDP)) == -1)//UDP
Socket Creation
    {
        die("socket");
    }
    puts("Socket Created");

    // this zero out the structure that
    memset((char *) &si_me, 0, sizeof(si_me));
     //Assiging IP, PORT
    si_me.sin_family = AF_INET;
    si_me.sin_port = htons(PORT);
    si_me.sin_addr.s_addr = htonl(INADDR_ANY);

    //Binding the socket with the server address
    if( bind(sockfd , (struct sockaddr*)&si_me, sizeof(si_me) ) == -1)
    {
        die("bind");
    }
    puts("Binding Done");

    //keep listening for data
    while(1)
    {
        printf("Waiting for data...");
        fflush(stdout);
```

```
        //Try to receive some data
        if ((recv_len = recvfrom(sockfd, buf, BUFLEN, 0, (struct sockaddr *)
&si_other, &slen)) == -1)
        {
           puts("Recv failed");
            die("recvfrom()");
        }



        printf(" Received Data: %s\n" , buf);
                    //Code for Calculator Functioning
         if(sscanf(buf, "%f %1[+-/*] %f", &op1, &operator, &op2) == 3)
    {
        printf("Operand 1: %.2f\n", op1);
        printf("Operand 2: %.2f\n", op2);
        printf("Operator: %c\n", operator);
         switch ( operator )
  {
     case '+' :
      result = op1 + op2;
   printf("Result is: %.2f + %.2f = %d\n",op1, op2, result);
     break;

     case '-' :
       result = op1 - op2;
   printf("Result is: %.2f - %.2f = %d\n",op1, op2, result);
     break;
```

```
  case '*' :
     result = op1 * op2;
printf("Result is: %.2f * %.2f = %d\n",op1, op2, result);
  break;

  case '/' :
    result = op1 / op2;
printf("Result is: %.2f / %.2f = %d\n",op1, op2, result);
  break;
        default :
    printf ("Invalid Selection \n");
      }
      }
      sprintf(res, "%i", result);

     //Replying the client with same data
     if (sendto(sockfd, res, recv_len, 0, (struct sockaddr*) &si_other, slen) == -
1)
     {
         die("sendto()");
     }



 }

 close(sockfd);
```

```
    return 0;
}

//Code Ends
//  Below is the Client Code

client.c
/*
    UDP-Client
*/
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
#include<arpa/inet.h>
#include<sys/socket.h>
 #define SERVER "127.0.0.1"
#define BUFLEN 512  //Max length of buffer
#define PORT 8888   //The port on which data will be sent

void die(char *sockfd)
{
        //For any error message
    perror(sockfd);
    exit(1);
}
```

```
int main(void)
{
    struct sockaddr_in si_other;
    int sockfd, i, slen=sizeof(si_other);

    char buf[BUFLEN];
    char message[BUFLEN];

    if ( (sockfd=socket(AF_INET, SOCK_DGRAM, IPPROTO_UDP)) == -
1)//Creating  Socket
    {
        die("socket");
    }
    puts("Socket created");

    memset((char *) &si_other, 0, sizeof(si_other));
          //Assigning IP, PORT
    si_other.sin_family = AF_INET;//Server Domain
    si_other.sin_port = htons(PORT);

    if (inet_aton(SERVER , &si_other.sin_addr) == 0)
    {
        fprintf(stderr, "inet_aton() failed\n");
        exit(1);
    }
 //Communication between Client and Server
```

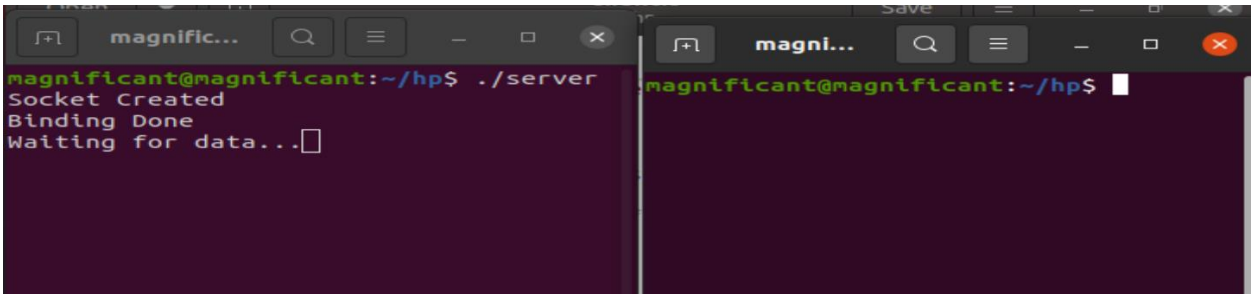
```
    while(1)
    {
        printf("Enter the expression in the num1 operand num2 form  ");
        gets(message);


        //Sending the Data to the Server
        if (sendto(sockfd, message, strlen(message) , 0 , (struct sockaddr *)
&si_other, slen)==-1)
        {
            die("sendto()");
        }
         puts("Data Sent to the server");
                    memset(buf,'\0', BUFLEN);//Clearing the buffer by filling null as it
may have previously rec data
        //trying  to receive some data
        if (recvfrom(sockfd, buf, BUFLEN, 0, (struct sockaddr *) &si_other, &slen)
== -1)
        {
        puts("rec fail");
            die("recvfrom()");
            break;
        }




    }
 //Socket Closed
```

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

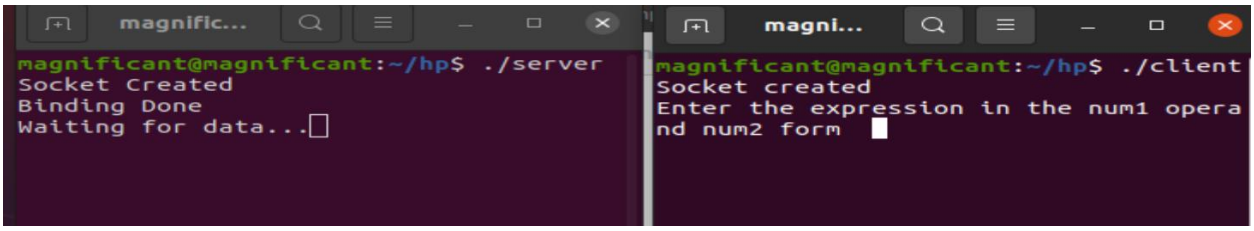
OUTPUT (Left Screen is Server Side, Right Screen is Client Side)

Firstly, we compile the c program using the following :

Then we run the server by using ./server command



Then we run the client by suing ./client command



Here the server sends the output according to the user information entered :

```
magnificant@magnificant:~/hp$ ./server
Socket Created
Binding Done
Waiting for data... Received Data: 88+8
Operand 1: 88.00
Operand 2: 8.00
Operator: +
Result is: 88.00 + 8.00 = 96
Waiting for data...
```

```
magnificant@magnificant:~/hp$ ./client
Socket created
Enter the expression in the num1 opera
nd num2 form  88+8
Data Sent to the server
Enter the expression in the num1 opera
nd num2 form
```

```
magnificant@magnificant:~/hp$ ./server
Socket Created
Binding Done
Waiting for data... Received Data: 88+8
Operand 1: 88.00
Operand 2: 8.00
Operator: +
Result is: 88.00 + 8.00 = 96
Waiting for data... Received Data: 10*5
Operand 1: 10.00
Operand 2: 5.00
Operator: *
Result is: 10.00 * 5.00 = 50
Waiting for data...
```

```
magnificant@magnificant:~/hp$ ./client
Socket created
Enter the expression in the num1 opera
nd num2 form  88+8
Data Sent to the server
Enter the expression in the num1 opera
nd num2 form  10*5
Data Sent to the server
Enter the expression in the num1 opera
nd num2 form
```

```
magnificant@magnificant:~/hp$ ./server
Socket Created
Binding Done
Waiting for data... Received Data: 88+8
Operand 1: 88.00
Operand 2: 8.00
Operator: +
Result is: 88.00 + 8.00 = 96
Waiting for data... Received Data: 10*5
Operand 1: 10.00
Operand 2: 5.00
Operator: *
Result is: 10.00 * 5.00 = 50
Waiting for data... Received Data: 75-8
Operand 1: 75.00
Operand 2: 8.00
Operator: -
Result is: 75.00 - 8.00 = 67
Waiting for data...
```

```
magnificant@magnificant:~/hp$ ./client
Socket created
Enter the expression in the num1 opera
nd num2 form  88+8
Data Sent to the server
Enter the expression in the num1 opera
nd num2 form  10*5
Data Sent to the server
Enter the expression in the num1 opera
nd num2 form  75-8
Data Sent to the server
Enter the expression in the num1 opera
nd num2 form
```

```
magnific...
Socket Created
Binding Done
Waiting for data... Received Data: 88+8
Operand 1: 88.00
Operand 2: 8.00
Operator: +
Result is: 88.00 + 8.00 = 96
Waiting for data... Received Data: 10*5
Operand 1: 10.00
Operand 2: 5.00
Operator: *
Result is: 10.00 * 5.00 = 50
Waiting for data... Received Data: 75-8
Operand 1: 75.00
Operand 2: 8.00
Operator: -
Result is: 75.00 - 8.00 = 67
Waiting for data... Received Data: 30/2
Operand 1: 30.00
Operand 2: 2.00
Operator: /
Result is: 30.00 / 2.00 = 15
Waiting for data...
```

```
magni...
magnificant@magnificant:~/hp$ ./client
Socket created
Enter the expression in the num1 opera
nd num2 form  88+8
Data Sent to the server
Enter the expression in the num1 opera
nd num2 form  10*5
Data Sent to the server
Enter the expression in the num1 opera
nd num2 form  75-8
Data Sent to the server
Enter the expression in the num1 opera
nd num2 form  30/2
Data Sent to the server
Enter the expression in the num1 opera
nd num2 form
```