COMPUTER NETWORKS PRACTICE ASSIGNMENT 2

Name: Vinayak Sethi Roll No: COE18B061

Calculator using TCP

Filename: CALC_TCP_Client.c

CALC_TCP_Client.c CALC_TCP_Server.c

```c
#include<stdio.h>
#include<stdlib.h>
#include<sys/socket.h>
#include<sys/types.h>
#include<netinet/in.h>
#include<unistd.h>
#include<string.h>

#define buffsize 150

int main()
{
    int client_socket,sin_size;
    char buffer[buffsize];
    int n,operation,num1,num2,result;
    struct sockaddr_in server_address;

    //create a socket
    client_socket = socket(AF_INET,SOCK_STREAM,0);
    if(client_socket == -1)
    {
        printf("\nSocket Creation Failure\n");
        exit(EXIT_FAILURE);
    }

    //specify an address for the socket
    server_address.sin_family = AF_INET;
    server_address.sin_port = htons(9009);
    server_address.sin_addr.s_addr = INADDR_ANY;

    sin_size = sizeof(struct sockaddr_in);

    //connect to server
    if(connect(client_socket,(struct sockaddr *)&server_address, sin_size) == 0)
        printf("Connect Successful\n");

    s:
        //Read server message
        memset(&buffer,0,sizeof(buffer));
        n = recv(client_socket,buffer,buffsize,0);
        printf("Server - %s\n",buffer);
```

```c
        scanf("%d",&num1); // Enter number 1
        send(client_socket,&num1, sizeof(int),0); // send number 1

        //Read server message
        memset(&buffer,0,sizeof(buffer));
        n = recv(client_socket,buffer,buffsize,0);
        printf("Server - %s\n",buffer);

        scanf("%d",&num2); // Enter number 2
        send(client_socket,&num2, sizeof(int),0); // send number 2

        //Read server message
        memset(&buffer,0,sizeof(buffer));
        n = recv(client_socket,buffer,buffsize,0);
        printf("Server - %s",buffer);

        scanf("%d",&operation); // Enter choice
        send(client_socket,&operation, sizeof(int),0); // send choice

        if(operation == 5)
            goto q;

        //read the result
        recv(client_socket, &result, sizeof(int),0);
        printf("Server - Your answer is %d\n",result);

        if(operation != 5)
            goto s;

    q:
        //close the socket
        printf("You have selected to exit\n");
        close(client_socket);

    return 0;
}
```

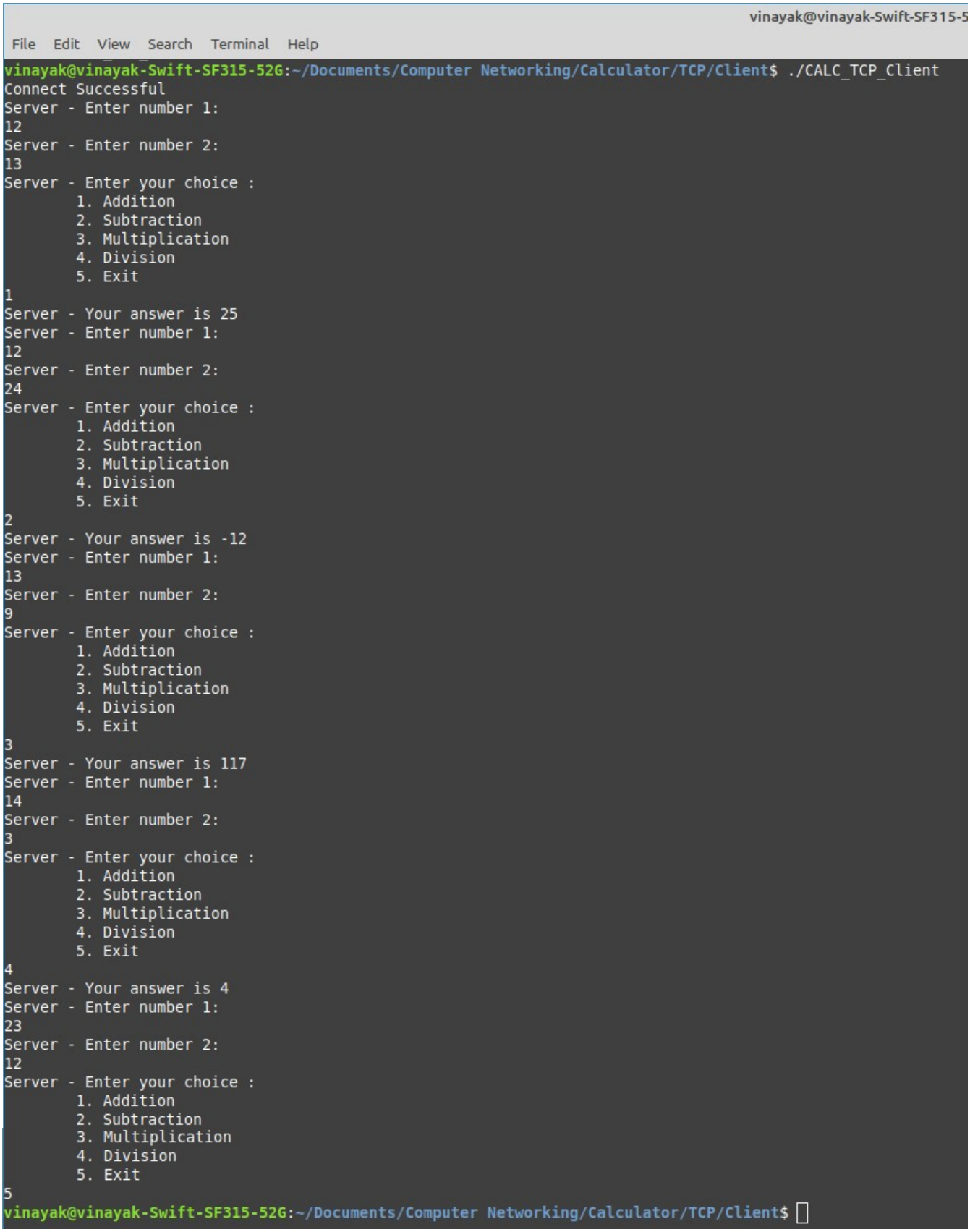
Filename: CALC_TCP_Server.c

CALC_TCP_Client.c | CALC_TCP_Server.c

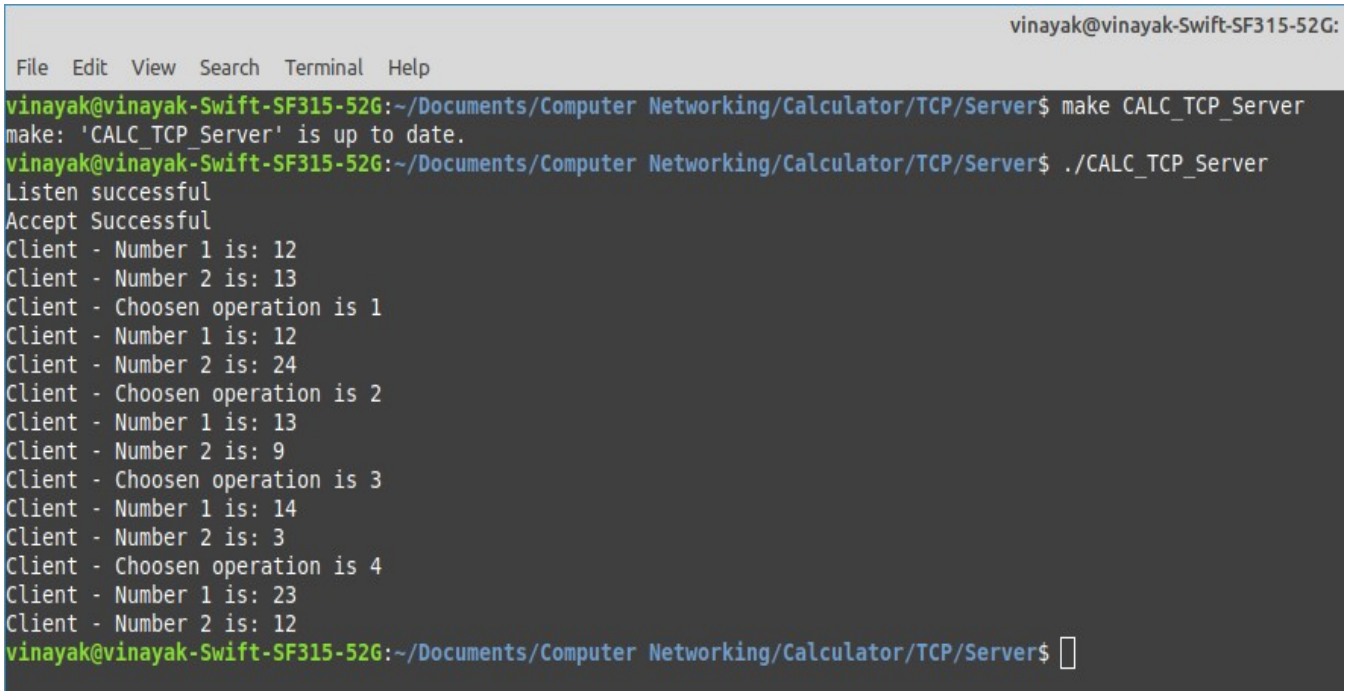
```c
#include<stdio.h>
#include<stdlib.h>
#include<sys/socket.h>
#include<sys/types.h>
#include<netinet/in.h>
#include<unistd.h>
#include<string.h>

#define buffsize 250

int main()
{
    int server_socket,client_socket,sin_size;
    char buffer[buffsize];
    struct sockaddr_in server_address, client_address;
    int n,operation,num1,num2,result;

    //create a socket
    server_socket = socket(AF_INET,SOCK_STREAM,0);
    if(server_socket == -1)
    {
        printf("\nSocket Creation Failure\n");
        exit(EXIT_FAILURE);
    }

    //specify an address for the socket
    server_address.sin_family = AF_INET;
    server_address.sin_port = htons(9009);
    server_address.sin_addr.s_addr = INADDR_ANY;
```

```c

    //bind with the client
    if( bind(server_socket, (const struct sockaddr *)&server_address, sizeof(server_address)) < 0)
    {
        printf("Could not bind to Client\n");
        exit(EXIT_FAILURE);
    }

    //listen to the incoming client request
    if(listen(server_socket, 10) == 0)
        printf("Listen successful\n");

    //accept a connection request from client
    sin_size = sizeof(struct sockaddr_in);
    if((client_socket = accept(server_socket, (struct sockaddr *)&client_address,&sin_size)) > 0)
        printf("Accept Successful\n");

    s:
        // Ask for number 1
        n = send(client_socket,"Enter number 1: ", strlen("Enter number 1 : "),0);
        recv(client_socket, &num1, sizeof(int),0); // read number 1
        printf("Client - Number 1 is: %d \n", num1);

        // Ask for number 2
        n = send(client_socket,"Enter number 2: ", strlen("Enter number 2: "),0);
        recv(client_socket, &num2, sizeof(int),0); //read number 2
        printf("Client - Number 2 is: %d \n", num2);

        //Ask for choice
        n = send(client_socket,"Enter your choice : \n\t1. Addition\n\t2. Subtraction\n\t3. Multiplication
            \n\t4. Division\n\t5. Exit\n", strlen("\nEnter your choice : \n\t1. Addition\n\t2. Subtraction
            \n\t3. Multiplication\n\t4. Division\n\t5. Exit\n"),0);
        recv(client_socket, &operation,sizeof(int),0); //  read the choice
        printf("Client - Choosen operation is %d\n",operation);

        if(operation > 5)
        {
            printf("Invalid Operation\n");
            goto s;
        }

        //perform operation
        switch(operation)
        {
            case 1: result = num1 + num2;
                    break;

            case 2: result = num1 - num2;
                    break;

            case 3: result = num1 * num2;
                    break;

            case 4: result = num1 / num2;
                    break;

            case 5: goto q;
                    break;
        }

        //show the answer
        send(client_socket, &result, sizeof(int),0);
        if(operation != 5)
            goto s;

    q:
        //close the socket
        close(client_socket);
        close(server_socket);

    return 0;
}
```

Output: ./CALC_TCP_Client



Output: ./CALC_TCP_Server



Calculator using UDP

Filename: CALC_UDP_Client.c

CALC_UDP_Client.c CALC_UDP_Server.c

```c
#include<stdio.h>
#include<stdlib.h>
#include<sys/socket.h>
#include<sys/types.h>
#include<netinet/in.h>
#include<unistd.h>
#include<string.h>

#define buffsize 150

int main()
{
    int client_socket;
    char buffer[buffsize], operation[3];
    int n,result;
    struct sockaddr_in server_address;

    //create a socket
    client_socket = socket(AF_INET,SOCK_DGRAM,0);
    if(client_socket == -1)
    {
        printf("\nSocket Creation Failure\n");
        exit(EXIT_FAILURE);
    }

    //specify an address for the socket
    server_address.sin_family = AF_INET;
    server_address.sin_port = htons(9009);
    server_address.sin_addr.s_addr = INADDR_ANY;

    socklen_t length = sizeof(server_address);

    sendto(client_socket,"Hello server", strlen("Hello server"),0,(struct sockaddr *)&server_address, sizeof(server_address));

    s:
        //Read server message
        memset(&buffer,0,sizeof(buffer));
        n = recvfrom(client_socket,buffer,sizeof(buffer),0,(struct sockaddr *)&server_address,&length);
        printf("Server - %s\n",buffer);

```

```c
        fgets(buffer, buffsize, stdin); // Enter number 1
        sendto(client_socket, buffer, sizeof(buffer),0,(struct sockaddr *)&server_address,sizeof(server_address)); // send number 1

        //Read server message
        memset(&buffer,0,sizeof(buffer));
        n = recvfrom(client_socket,(char *)buffer,sizeof(buffer),0,(struct sockaddr *)&server_address,&length);
        printf("Server - %s\n",buffer);

        fgets(buffer, buffsize, stdin); // Enter number 2
        sendto(client_socket, buffer, sizeof(buffer),0,(struct sockaddr *)&server_address,sizeof(server_address)); // send number 2

        //Read server message
        memset(&buffer,0,sizeof(buffer));
        n = recvfrom(client_socket,buffer,sizeof(buffer),0,(struct sockaddr *)&server_address,&length);
        printf("Server - %s\n",buffer);

        fgets(buffer, buffsize, stdin); // Enter choice
        strcpy(operation , buffer);
        sendto(client_socket, buffer, sizeof(buffer),0,(struct sockaddr *)&server_address,sizeof(server_address)); // send choice

        if(operation[0] == '5')
            goto q;

        //read the result
        memset(&buffer,0,sizeof(buffer));
        n = recvfrom(client_socket,buffer, sizeof(buffer),0,(struct sockaddr *)&server_address,&length);
        result = atoi(buffer);
        printf("Server - Your answer is %d\n",result);

        if(operation[0] != '5')
            goto s;

    q:
        //close the socket
        printf("You have selected to exit\n");
        close(client_socket);

    return 0;
}
```

Filename: CALC_UDP_Server.c

CALC_UDP_Client.c | CALC_UDP_Server.c

```c
#include<stdio.h>
#include<stdlib.h>
#include<sys/socket.h>
#include<sys/types.h>
#include<netinet/in.h>
#include<unistd.h>
#include<string.h>

#define buffsize 250

int main()
{
    int server_socket;
    char operation[3], buffer[20], num1[buffsize], num2[buffsize], res[buffsize];
    struct sockaddr_in server_address, client_address;
    int n,a,b,choice,result;

    //create a socket
    server_socket = socket(AF_INET,SOCK_DGRAM,0);
    if(server_socket == -1)
    {
        printf("\nSocket Creation Failure\n");
        exit(EXIT_FAILURE);
    }

    //specify an address for the socket
    server_address.sin_family = AF_INET;
    server_address.sin_port = htons(9009);
    server_address.sin_addr.s_addr = INADDR_ANY;

    //bind with the client
    if( bind(server_socket, (const struct sockaddr *)&server_address, sizeof(server_address)) < 0)
    {
        printf("Could not bind to Client\n");
        exit(EXIT_FAILURE);
    }

    socklen_t length = sizeof(server_address);
    n = recvfrom(server_socket, buffer, sizeof(buffer),0,(struct sockaddr *)&client_address,&length);

    s:
```

```c
        // Ask for number 1
        sendto(server_socket,"Enter number 1: ", strlen("Enter number 1: "),0,(struct sockaddr *)&client_address, sizeof(client_address));
        n = recvfrom(server_socket, num1, sizeof(num1),0,(struct sockaddr *)&client_address,&length); // read number 1
        a = atoi(num1);
        printf("Client - Number 1 is: %d \n", a);

        // Ask for number 2
        sendto(server_socket,"Enter number 2: ", strlen("Enter number 2: "),0,(struct sockaddr *)&client_address,sizeof(client_address));
        n = recvfrom(server_socket, num2, sizeof(num2),0,(struct sockaddr *)&client_address,&length); //read number 2
        b = atoi(num2);
        printf("Client - Number 2 is: %d \n", b);

        //Ask for choice
        sendto(server_socket,"Enter your choice : \n\t1. Addition\n\t2. Subtraction\n\t3. Multiplication\n\t4. Division\n\t5. Exit\n",
            strlen("\nEnter your choice : \n\t1. Addition\n\t2. Subtraction\n\t3. Multiplication\n\t4. Division\n\t5. Exit\n"),0,
            (struct sockaddr *)&client_address,sizeof(client_address));
        n = recvfrom(server_socket, operation,sizeof(operation),0,(struct sockaddr *)&client_address,&length); //  read the choice
        choice = atoi(operation);
        printf("Client - Choosen operation is %d\n",choice);

        if(choice > 5)
        {
            printf("Invalid Operation\n");
            goto s;
        }

        //perform operation
        switch(choice)
        {
            case 1: result = a + b;
                    break;

            case 2: result = a - b;
                    break;

            case 3: result = a * b;
                    break;

            case 4: result = a / b;
                    break;

            case 5: goto q;
                    break;
        }

        //show the answer
        sprintf(res, "%d" ,result);
        sendto(server_socket, res, sizeof(res),0,(struct sockaddr *)&client_address,sizeof(client_address));

        if(choice != 5)
            goto s;

    q:
        //close the socket
        close(server_socket);

    return 0;
}
```

Output: ./CALC_UDP_Client

```
vinayak@vinayak-Swift-SF315-52G:

File  Edit  View  Search  Terminal  Help
vinayak@vinayak-Swift-SF315-52G:~/Documents/Computer Networking/Calculator/UDP/Client$ make CALC_UDP_Client
make: 'CALC_UDP_Client' is up to date.
vinayak@vinayak-Swift-SF315-52G:~/Documents/Computer Networking/Calculator/UDP/Client$ ./CALC_UDP_Client
Server - Enter number 1:
12
Server - Enter number 2:
23
Server - Enter your choice :
        1. Addition
        2. Subtraction
        3. Multiplication
        4. Division
        5. Exit

1
Server - Your answer is 35
Server - Enter number 1:
13
Server - Enter number 2:
25
```

```
Server - Enter your choice :
        1. Addition
        2. Subtraction
        3. Multiplication
        4. Division
        5. Exit

2
Server - Your answer is -12
Server - Enter number 1:
14
Server - Enter number 2:
8
Server - Enter your choice :
        1. Addition
        2. Subtraction
        3. Multiplication
        4. Division
        5. Exit

3
Server - Your answer is 112
Server - Enter number 1:
19
Server - Enter number 2:
4
Server - Enter your choice :
        1. Addition
        2. Subtraction
        3. Multiplication
        4. Division
        5. Exit

4
Server - Your answer is 4
Server - Enter number 1:
13
Server - Enter number 2:
14
Server - Enter your choice :
        1. Addition
        2. Subtraction
        3. Multiplication
        4. Division
        5. Exit

5
vinayak@vinayak-Swift-SF315-52G:~/Documents/Computer Networking/Calculator/UDP/Client$
```

Output: ./CALC_UDP_Server

```
                                                    vinayak@vinayak-Swift-SF315-52G:
File  Edit  View  Search  Terminal  Help
vinayak@vinayak-Swift-SF315-52G:~/Documents/Computer Networking/Calculator/UDP/Server$ make CALC_UDP_Server
make: 'CALC_UDP_Server' is up to date.
vinayak@vinayak-Swift-SF315-52G:~/Documents/Computer Networking/Calculator/UDP/Server$ ./CALC_UDP_Server
Client - Number 1 is: 12
Client - Number 2 is: 23
Client - Choosen operation is 1
Client - Number 1 is: 13
Client - Number 2 is: 25
Client - Choosen operation is 2
Client - Number 1 is: 14
Client - Number 2 is: 8
Client - Choosen operation is 3
Client - Number 1 is: 19
Client - Number 2 is: 4
Client - Choosen operation is 4
Client - Number 1 is: 13
Client - Number 2 is: 14
vinayak@vinayak-Swift-SF315-52G:~/Documents/Computer Networking/Calculator/UDP/Server$
```