

COMPUTER NETWORKS PRACTICE ASSIGNMENT 2

Name: Vinayak Sethi

Roll No: COE18B061

Calculator using TCP

Filename: CALC_TCP_Client.c

```
1  #include<stdio.h>
2  #include<stdlib.h>
3  #include<sys/socket.h>
4  #include<sys/types.h>
5  #include<netinet/in.h>
6  #include<unistd.h>
7  #include<string.h>
8
9  #define bufsize 150
10
11 int main()
12 {
13     int client_socket,sin_size;
14     char buffer[bufsize];
15     int n,operation,num1,num2,result;
16     struct sockaddr_in server_address;
17
18     //create a socket
19     client_socket = socket(AF_INET,SOCK_STREAM,0);
20     if(client_socket == -1)
21     {
22         printf("\nSocket Creation Failure\n");
23         exit(EXIT_FAILURE);
24     }
25
26     //specify an address for the socket
27     server_address.sin_family = AF_INET;
28     server_address.sin_port = htons(9009);
29     server_address.sin_addr.s_addr = INADDR_ANY;
30
31     sin_size = sizeof(struct sockaddr_in);
32
33     //connect to server
34     if(connect(client_socket,(struct sockaddr *)&server_address, sin_size) == 0)
35         printf("Connect Successful\n");
36
37     s:
38         //Read server message
39         memset(&buffer,0,sizeof(buffer));
40         n = recv(client_socket,buffer,bufsize,0);
41         printf("Server - %s\n",buffer);
```

```

42
43     scanf("%d",&num1); // Enter number 1
44     send(client_socket,&num1, sizeof(int),0); // send number 1
45
46     //Read server message
47     memset(&buffer,0,sizeof(buffer));
48     n = recv(client_socket,buffer,buffsize,0);
49     printf("Server - %s\n",buffer);
50
51     scanf("%d",&num2); // Enter number 2
52     send(client_socket,&num2, sizeof(int),0); // send number 2
53
54     //Read server message
55     memset(&buffer,0,sizeof(buffer));
56     n = recv(client_socket,buffer,buffsize,0);
57     printf("Server - %s",buffer);
58
59     scanf("%d",&operation); // Enter choice
60     send(client_socket,&operation, sizeof(int),0); // send choice
61
62     if(operation == 5)
63         goto q;
64
65     //read the result
66     recv(client_socket, &result, sizeof(int),0);
67     printf("Server - Your answer is %d\n",result);
68
69     if(operation != 5)
70         goto s;
71
72     q:
73     //close the socket
74     printf("You have selected to exit\n");
75     close(client_socket);
76
77     return 0;
78 }

```

Filename: CALC_TCP_Server.c

```

1  #include<stdio.h>
2  #include<stdlib.h>
3  #include<sys/socket.h>
4  #include<sys/types.h>
5  #include<netinet/in.h>
6  #include<unistd.h>
7  #include<string.h>
8
9  #define buffsize 250
10
11 int main()
12 {
13     int server_socket,client_socket,sin_size;
14     char buffer[buffsize];
15     struct sockaddr_in server_address, client_address;
16     int n,operation,num1,num2,result;
17
18     //create a socket
19     server_socket = socket(AF_INET,SOCK_STREAM,0);
20     if(server_socket == -1)
21     {
22         printf("\nSocket Creation Failure\n");
23         exit(EXIT_FAILURE);
24     }
25
26     //specify an address for the socket
27     server_address.sin_family = AF_INET;
28     server_address.sin_port = htons(9009);
29     server_address.sin_addr.s_addr = INADDR_ANY;

```

```

30
31 //bind with the client
32 if( bind(server_socket, (const struct sockaddr *)&server_address, sizeof(server_address)) < 0)
33 {
34     printf("Could not bind to Client\n");
35     exit(EXIT_FAILURE);
36 }
37
38 //listen to the incoming client request
39 if(listen(server_socket, 10) == 0)
40     printf("Listen successful\n");
41
42 //accept a connection request from client
43 sin_size = sizeof(struct sockaddr_in);
44 if((client_socket = accept(server_socket, (struct sockaddr *)&client_address,&sin_size)) > 0)
45     printf("Accept Successful\n");
46
47 s:
48     // Ask for number 1
49     n = send(client_socket,"Enter number 1: ", strlen("Enter number 1 : "),0);
50     recv(client_socket, &num1, sizeof(int),0); // read number 1
51     printf("Client - Number 1 is: %d \n", num1);
52
53     // Ask for number 2
54     n = send(client_socket,"Enter number 2: ", strlen("Enter number 2: "),0);
55     recv(client_socket, &num2, sizeof(int),0); //read number 2
56     printf("Client - Number 2 is: %d \n", num2);
57
58     //Ask for choice
59     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);
60     recv(client_socket, &operation,sizeof(int),0); // read the choice
61     printf("Client - Chosen operation is %d\n",operation);
62
63     if(operation > 5)
64     {
65         printf("Invalid Operation\n");
66         goto s;
67     }
68
69     //perform operation
70     switch(operation)
71     {
72     case 1: result = num1 + num2;
73             break;
74
75     case 2: result = num1 - num2;
76             break;
77
78     case 3: result = num1 * num2;
79             break;
80
81     case 4: result = num1 / num2;
82             break;
83
84     case 5: goto q;
85             break;
86
87     }
88
89     //show the answer
90     send(client_socket, &result, sizeof(int),0);
91     if(operation != 5)
92         goto s;
93
94 q:
95     //close the socket
96     close(client_socket);
97     close(server_socket);
98
99     return 0;
100 }
101

```


Output: ./CALC_TCP_Client

```
vinayak@vinayak-Swift-SF315-5
File Edit View Search Terminal Help
vinayak@vinayak-Swift-SF315-526:~/Documents/Computer Networking/Calculator/TCP/Client$ ./CALC_TCP_Client
Connect Successful
Server - Enter number 1:
12
Server - Enter number 2:
13
Server - Enter your choice :
    1. Addition
    2. Subtraction
    3. Multiplication
    4. Division
    5. Exit
1
Server - Your answer is 25
Server - Enter number 1:
12
Server - Enter number 2:
24
Server - Enter your choice :
    1. Addition
    2. Subtraction
    3. Multiplication
    4. Division
    5. Exit
2
Server - Your answer is -12
Server - Enter number 1:
13
Server - Enter number 2:
9
Server - Enter your choice :
    1. Addition
    2. Subtraction
    3. Multiplication
    4. Division
    5. Exit
3
Server - Your answer is 117
Server - Enter number 1:
14
Server - Enter number 2:
3
Server - Enter your choice :
    1. Addition
    2. Subtraction
    3. Multiplication
    4. Division
    5. Exit
4
Server - Your answer is 4
Server - Enter number 1:
23
Server - Enter number 2:
12
Server - Enter your choice :
    1. Addition
    2. Subtraction
    3. Multiplication
    4. Division
    5. Exit
5
vinayak@vinayak-Swift-SF315-526:~/Documents/Computer Networking/Calculator/TCP/Client$
```

Output: ./CALC_TCP_Server

```
vinayak@vinayak-Swift-SF315-52G:
File Edit View Search Terminal Help
vinayak@vinayak-Swift-SF315-52G:~/Documents/Computer Networking/Calculator/TCP/Server$ make CALC_TCP_Server
make: 'CALC_TCP_Server' is up to date.
vinayak@vinayak-Swift-SF315-52G:~/Documents/Computer Networking/Calculator/TCP/Server$ ./CALC_TCP_Server
Listen successful
Accept Successful
Client - Number 1 is: 12
Client - Number 2 is: 13
Client - Chosen operation is 1
Client - Number 1 is: 12
Client - Number 2 is: 24
Client - Chosen operation is 2
Client - Number 1 is: 13
Client - Number 2 is: 9
Client - Chosen operation is 3
Client - Number 1 is: 14
Client - Number 2 is: 3
Client - Chosen operation is 4
Client - Number 1 is: 23
Client - Number 2 is: 12
vinayak@vinayak-Swift-SF315-52G:~/Documents/Computer Networking/Calculator/TCP/Server$
```

Calculator using UDP

Filename: CALC_UDP_Client.c

```
CALC_UDP_Client.c x CALC_UDP_Server.c x
1 #include<stdio.h>
2 #include<stdlib.h>
3 #include<sys/socket.h>
4 #include<sys/types.h>
5 #include<netinet/in.h>
6 #include<unistd.h>
7 #include<string.h>
8
9 #define bufsize 150
10
11 int main()
12 {
13     int client_socket;
14     char buffer[bufsize], operation[3];
15     int n,result;
16     struct sockaddr_in server_address;
17
18     //create a socket
19     client_socket = socket(AF_INET,SOCK_DGRAM,0);
20     if(client_socket == -1)
21     {
22         printf("\nSocket Creation Failure\n");
23         exit(EXIT_FAILURE);
24     }
25
26     //specify an address for the socket
27     server_address.sin_family = AF_INET;
28     server_address.sin_port = htons(9009);
29     server_address.sin_addr.s_addr = INADDR_ANY;
30
31     socklen_t length = sizeof(server_address);
32
33     sendto(client_socket,"Hello server", strlen("Hello server"),0,(struct sockaddr *)&server_address, sizeof(server_address));
34
35     s:
36     //Read server message
37     memset(&buffer,0,sizeof(buffer));
38     n = recvfrom(client_socket,buffer,sizeof(buffer),0,(struct sockaddr *)&server_address,&length);
39     printf("Server - %s\n",buffer);
40
```

```

41     fgets(buffer, buffsize, stdin); // Enter number 1
42     sendto(client_socket, buffer, sizeof(buffer), 0, (struct sockaddr *)&server_address, sizeof(server_address)); // send number 1
43
44     //Read server message
45     memset(&buffer, 0, sizeof(buffer));
46     n = recvfrom(client_socket, (char *)buffer, sizeof(buffer), 0, (struct sockaddr *)&server_address, &length);
47     printf("Server - %s\n", buffer);
48
49     fgets(buffer, buffsize, stdin); // Enter number 2
50     sendto(client_socket, buffer, sizeof(buffer), 0, (struct sockaddr *)&server_address, sizeof(server_address)); // send number 2
51
52     //Read server message
53     memset(&buffer, 0, sizeof(buffer));
54     n = recvfrom(client_socket, buffer, sizeof(buffer), 0, (struct sockaddr *)&server_address, &length);
55     printf("Server - %s\n", buffer);
56
57     fgets(buffer, buffsize, stdin); // Enter choice
58     strcpy(operation, buffer);
59     sendto(client_socket, buffer, sizeof(buffer), 0, (struct sockaddr *)&server_address, sizeof(server_address)); // send choice
60
61     if(operation[0] == '5')
62     {
63         goto q;
64     }
65
66     //read the result
67     memset(&buffer, 0, sizeof(buffer));
68     n = recvfrom(client_socket, buffer, sizeof(buffer), 0, (struct sockaddr *)&server_address, &length);
69     result = atoi(buffer);
70     printf("Server - Your answer is %d\n", result);
71
72     if(operation[0] != '5')
73     {
74         goto s;
75     }
76
77 q:
78     //close the socket
79     printf("You have selected to exit\n");
80     close(client_socket);
81
82     return 0;
83 }

```

Filename: CALC_UDP_Server.c

```

1  #include<stdio.h>
2  #include<stdlib.h>
3  #include<sys/socket.h>
4  #include<sys/types.h>
5  #include<netinet/in.h>
6  #include<unistd.h>
7  #include<string.h>
8
9  #define buffsize 250
10
11 int main()
12 {
13     int server_socket;
14     char operation[3], buffer[20], num1[buffsize], num2[buffsize], res[buffsize];
15     struct sockaddr_in server_address, client_address;
16     int n,a,b,choice,result;
17
18     //create a socket
19     server_socket = socket(AF_INET, SOCK_DGRAM, 0);
20     if(server_socket == -1)
21     {
22         printf("\nSocket Creation Failure\n");
23         exit(EXIT_FAILURE);
24     }
25
26     //specify an address for the socket
27     server_address.sin_family = AF_INET;
28     server_address.sin_port = htons(9009);
29     server_address.sin_addr.s_addr = INADDR_ANY;
30
31     //bind with the client
32     if( bind(server_socket, (const struct sockaddr *)&server_address, sizeof(server_address)) < 0)
33     {
34         printf("Could not bind to Client\n");
35         exit(EXIT_FAILURE);
36     }
37
38     socklen_t length = sizeof(server_address);
39     n = recvfrom(server_socket, buffer, sizeof(buffer), 0, (struct sockaddr *)&client_address, &length);
40
41     s:

```

```

42 // Ask for number 1
43 sendto(server_socket, "Enter number 1: ", strlen("Enter number 1: "), 0, (struct sockaddr *)&client_address, sizeof(client_address));
44 n = recvfrom(server_socket, num1, sizeof(num1), 0, (struct sockaddr *)&client_address, &length); // read number 1
45 a = atoi(num1);
46 printf("Client - Number 1 is: %d \n", a);
47
48 // Ask for number 2
49 sendto(server_socket, "Enter number 2: ", strlen("Enter number 2: "), 0, (struct sockaddr *)&client_address, sizeof(client_address));
50 n = recvfrom(server_socket, num2, sizeof(num2), 0, (struct sockaddr *)&client_address, &length); // read number 2
51 b = atoi(num2);
52 printf("Client - Number 2 is: %d \n", b);
53
54 // Ask for choice
55 sendto(server_socket, "Enter your choice : \n\t1. Addition\n\t2. Subtraction\n\t3. Multiplication\n\t4. Division\n\t5. Exit\n",
56         strlen("\nEnter your choice : \n\t1. Addition\n\t2. Subtraction\n\t3. Multiplication\n\t4. Division\n\t5. Exit\n"), 0,
57         (struct sockaddr *)&client_address, sizeof(client_address));
58 n = recvfrom(server_socket, operation, sizeof(operation), 0, (struct sockaddr *)&client_address, &length); // read the choice
59 choice = atoi(operation);
60 printf("Client - Chosen operation is %d\n", choice);
61
62 if(choice > 5)
63 {
64     printf("Invalid Operation\n");
65     goto s;
66 }
67
68 //perform operation
69 switch(choice)
70 {
71     case 1: result = a + b;
72             break;
73
74     case 2: result = a - b;
75             break;
76
77     case 3: result = a * b;
78             break;
79
80     case 4: result = a / b;
81             break;
82
83     case 5: goto q;
84             break;
85 }
86
87 //show the answer
88 sprintf(res, "%d", result);
89 sendto(server_socket, res, sizeof(res), 0, (struct sockaddr *)&client_address, sizeof(client_address));
90
91 if(choice != 5)
92     goto s;
93
94 q:
95 //close the socket
96 close(server_socket);
97
98 return 0;
99 }
100

```

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/Clients$

```

Output: ./CALC_UDP_Server

```

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 - Chosen operation is 1
Client - Number 1 is: 13
Client - Number 2 is: 25
Client - Chosen operation is 2
Client - Number 1 is: 14
Client - Number 2 is: 8
Client - Chosen operation is 3
Client - Number 1 is: 19
Client - Number 2 is: 4
Client - Chosen operation is 4
Client - Number 1 is: 13
Client - Number 2 is: 14
vinayak@vinayak-Swift-SF315-52G:~/Documents/Computer Networking/Calculator/UDP/Server$

```