**ASSIGNMENT – II**

**Q1. Server and Client reliable Communications:**

**Server :**

#include <netdb.h>

#include <netinet/in.h>

#include <stdlib.h>

#include <string.h>

#include <sys/socket.h>

#include <sys/types.h>

#define MAX 80

#define PORT 8080

#define SA struct sockaddr

void func(int sockfd)

{

    char buff[MAX];

    int n;

    // infinite loop for chat

    for (;;) {

        bzero(buff, MAX);

        read(sockfd, buff, sizeof(buff));

printf("From client: %s\t To client : ", buff);

        bzero(buff, MAX);

        n = 0;

        while ((buff[n++] = getchar()) != '\n');

        write(sockfd, buff, sizeof(buff));

        if (strncmp("exit", buff, 4) == 0) {

            printf("Server Exit...\n");

            break;

        }

    }

}

int main(){

    int sockfd, connfd, len;

    struct sockaddr\_in servaddr, cli;

    sockfd = socket(AF\_INET, SOCK\_STREAM, 0);

    if (sockfd == -1) {

        printf("socket creation failed...\n");

        exit(0);

    }

    else

        printf("Socket successfully created..\n");

    bzero(&servaddr, sizeof(servaddr));

    // assign IP, PORT

    servaddr.sin\_family = AF\_INET;

servaddr.sin\_addr.s\_addr = inet\_addr("127.0.0.1");

    servaddr.sin\_port = htons(PORT);

if ((bind(sockfd, (SA\*)&servaddr, sizeof(servaddr))) != 0) {

        printf("socket bind failed...\n");

        exit(0);

    }

    else

        printf("Socket successfully binded..\n");

    if ((listen(sockfd, 5)) != 0) {

        printf("Listen failed...\n");

        exit(0);

    }

    else

        printf("Server listening..\n");

    len = sizeof(cli);

    connfd = accept(sockfd, (SA\*)&cli, &len);

    if (connfd < 0) {

        printf("server acccept failed...\n");

        exit(0);

    }

    else

        printf("server acccept the client...\n");

    func(connfd);

    // After chatting close the socket

    close(sockfd);

}

**Client:**

#include <netdb.h>

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <sys/socket.h>

#define MAX 80

#define PORT 8080

#define SA struct sockaddr

void func(int sockfd)

{

    char buff[MAX];

    int n;

    for (;;) {

        bzero(buff, sizeof(buff));

        printf("Enter the string : ");

        n = 0;

while((buff[n++] = getchar()) != '\n');

        write(sockfd, buff, sizeof(buff));

        bzero(buff, sizeof(buff));

        read(sockfd, buff, sizeof(buff));

        printf("From Server : %s", buff);

        if ((strncmp(buff, "exit", 4)) == 0) {

            printf("Client Exit...\n");

            break;

        }

    }

}

int main()

{

    int sockfd, connfd;

    struct sockaddr\_in servaddr, cli;

    sockfd = socket(AF\_INET, SOCK\_STREAM, 0);

    if (sockfd == -1) {

        printf("socket creation failed...\n");

        exit(0);

    }

    else

    printf("Socket successfully created..\n");

    bzero(&servaddr, sizeof(servaddr));

    servaddr.sin\_family = AF\_INET;

servaddr.sin\_addr.s\_addr = inet\_addr("127.0.0.1");

    servaddr.sin\_port = htons(PORT);

if (connect(sockfd, (SA\*)&servaddr, sizeof(servaddr)) != 0) {

printf("connection with the server failed...\n");

        exit(0);

    }

    else

        printf("connected to the server..\n");

    func(sockfd);

    close(sockfd);

}

**Output**:

**Server Side:-**

Socket successfully created..

Socket successfully binded..

Server listening..

server acccept the client...

From client: hi

To client : hello

From client: exit

To client : exit

Server Exit...

**Client Side:-**

Socket successfully created..

connected to the server..

Enter the string : hi

From Server : hello

Enter the string : exit

From Server : exit

Client Exit..

**Q2 Client Server Calculator**

**Server:**

#include <netdb.h>

#include <netinet/in.h>

#include <stdlib.h>

#include <string.h>

#include <sys/socket.h>

#include <sys/types.h>

#define MAX 80

#define PORT 8080

#define SA struct sockaddr

void func(int sockfd)

{

int num1, num2, result, choice;

while(1){

n = write(sockfd, 'Enter 1st Number: ', strlen('Enter 1st Number: '));

read(sockfd, &num1, sizeof(int));

printf("Client - Number 1: %d\t ", num1);

n = write(sockfd, 'Enter 2nd Number: ', strlen('Enter 2nd Number: '));

read(sockfd, &num2, sizeof(num2));

printf("Client - Number 2: %d\t ", num2);

n = write(sockfd, "Enter Your Choice :\n1.Add\n2.Subtraction\n3.Multiplication\n4.Division\n5.Exit",strlen('Enter Your Choice :\n1.Add\n2.Subtraction\n3.Multiplication\n4.Division\n5.Exit'));

read(sockfd, &choice, sizeof(choice));

printf("Client - choice : %d\t", choice);

switch(choice){

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:

exit(0);

}

write(sockfd, &result, sizeof(int));

}

}

int main(){

int sockfd, connfd, len;

struct sockaddr\_in servaddr, cli;

sockfd = socket(AF\_INET, SOCK\_STREAM, 0);

if (sockfd == -1) {

printf("socket creation failed...\n");

exit(0);

}

else

printf("Socket successfully created..\n");

bzero(&servaddr, sizeof(servaddr));

// assign IP, PORT

servaddr.sin\_family = AF\_INET;

servaddr.sin\_addr.s\_addr = inet\_addr("127.0.0.1");

servaddr.sin\_port = htons(PORT);

if ((bind(sockfd, (SA\*)&servaddr, sizeof(servaddr))) != 0) {

printf("socket bind failed...\n");

exit(0);

}

else

printf("Socket successfully binded..\n");

if ((listen(sockfd, 5)) != 0) {

printf("Listen failed...\n");

exit(0);

}

else

printf("Server listening..\n");

len = sizeof(cli);

connfd = accept(sockfd, (SA\*)&cli, &len);

if (connfd < 0) {

printf("server acccept failed...\n");

exit(0);

}

else

printf("server acccept the client...\n");

func(connfd);

close(sockfd);

}

**Client:**

#include <netdb.h>

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <sys/socket.h>

#define MAX 80

#define PORT 8080

#define SA struct sockaddr

void func(int sockfd)

{

char buff[MAX];

int num1, num2, result, choice;

for (;;) {

bzero(buff, sizeof(buff));

n = read(sockfd, buff, sizeof(buff));

printf('server - %s\t', buff);

scanf("%d",&num1);

write(sockfd, &num1, sizeof(int));

bzero(buff, sizeof(buff));

n = read(sockfd, buff, sizeof(buff));

printf('server - %s\t', buff);

scanf("%d",&num2);

write(sockfd, &num2, sizeof(int));

bzero(buff, sizeof(buff));

n = read(sockfd, buff, sizeof(buff));

printf('server - %s\t', buff);

scanf("%d",&choice);

write(sockfd, &choice, sizeof(int));

read(sockfd, &result, sizeof(int));

printf(“Server – Result :%d\t”, result):

}

}

int main()

{

int sockfd, connfd;

struct sockaddr\_in servaddr, cli;

sockfd = socket(AF\_INET, SOCK\_STREAM, 0);

if (sockfd == -1) {

printf("socket creation failed...\n");

exit(0);

}

else

printf("Socket successfully created..\n");

bzero(&servaddr, sizeof(servaddr));

servaddr.sin\_family = AF\_INET;

servaddr.sin\_addr.s\_addr = inet\_addr("127.0.0.1");

servaddr.sin\_port = htons(PORT);

if (connect(sockfd, (SA\*)&servaddr, sizeof(servaddr)) != 0) {

printf("connection with the server failed...\n");

exit(0);

}

else

printf("connected to the server..\n");

func(sockfd);

close(sockfd);

}

**Output:**

**Server Side:-**

Socket successfully created..

Socket successfully binded..

Server listening..

server acccept the client...

client – Number 1: 8

client – Number 2: 9

client – choice: 1

**Client Side:-**

Socket successfully created..

connected to the server..

Server – Enter 1st Number: 8

Server – Enter 2nd Number: 9

Server – Enter Your Choice

1.Add

2.Subtraction

3.Multiplication

4.Division

5.Exit

1

Server – Result: 17

Client Exit..

**Q3. TCP Socket Programming for One Way Communication from client to Server**

**Server:**

#include <netdb.h>

#include <netinet/in.h>

#include <stdlib.h>

#include <string.h>

#include <sys/socket.h>

#include <sys/types.h>

#define MAX 80

#define PORT 8080

#define SA struct sockaddr

// Function designed for chat between client and server.

void func(int sockfd)

{

    char buff[MAX];

    int n;

    for (;;) {

        bzero(buff, MAX);

        read(sockfd, buff, sizeof(buff));

printf("From client: %s\t To client : ", buff);

        bzero(buff, MAX);

        n = 0;

        if (strncmp("exit", buff, 4) == 0) {

            printf("Server Exit...\n");

            break;

        }

    }

}

int main()

{

    int sockfd, connfd, len;

    struct sockaddr\_in servaddr, cli;

    sockfd = socket(AF\_INET, SOCK\_STREAM, 0);

    if (sockfd == -1) {

        printf("socket creation failed...\n");

        exit(0);

    }

    else

        printf("Socket successfully created..\n");

    bzero(&servaddr, sizeof(servaddr));

    servaddr.sin\_family = AF\_INET;

    servaddr.sin\_addr.s\_addr = htonl(INADDR\_ANY);

    servaddr.sin\_port = htons(PORT);

if ((bind(sockfd, (SA\*)&servaddr, sizeof(servaddr))) != 0) {

        printf("socket bind failed...\n");

        exit(0);

    }

    else

        printf("Socket successfully binded..\n");

    if ((listen(sockfd, 5)) != 0) {

        printf("Listen failed...\n");

        exit(0);

    }

    else

        printf("Server listening..\n");

    len = sizeof(cli);

    connfd = accept(sockfd, (SA\*)&cli, &len);

    if (connfd < 0) {

        printf("server acccept failed...\n");

        exit(0);

    }

    else

        printf("server acccept the client...\n");

    func(connfd);

    close(sockfd);

}

**Client:**

#include <netdb.h>

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <sys/socket.h>

#define MAX 80

#define PORT 8080

#define SA struct sockaddr

void func(int sockfd)

{

    char buff[MAX];

    int n;

    for (;;) {

        bzero(buff, sizeof(buff));

        printf("Enter the string : ");

        n = 0;

while((buff[n++] = getchar()) != '\n');

        write(sockfd, buff, sizeof(buff));

        bzero(buff, sizeof(buff));

        if ((strncmp(buff, "exit", 4)) == 0) {

            printf("Client Exit...\n");

            break;

        }

    }

}

int main()

{

    int sockfd, connfd;

    struct sockaddr\_in servaddr, cli;

    sockfd = socket(AF\_INET, SOCK\_STREAM, 0);

    if (sockfd == -1) {

        printf("socket creation failed...\n");

        exit(0);

    }

    else

    printf("Socket successfully created..\n");

    bzero(&servaddr, sizeof(servaddr));

    servaddr.sin\_family = AF\_INET;

servaddr.sin\_addr.s\_addr = inet\_addr("127.0.0.1");

    servaddr.sin\_port = htons(PORT);

if (connect(sockfd, (SA\*)&servaddr, sizeof(servaddr)) != 0) {

printf("connection with the server failed...\n");

        exit(0);

    }

    else

        printf("connected to the server..\n");

    func(sockfd);

    close(sockfd);

}

**Output:**

**Server Side:-**

Socket successfully created..

Socket successfully binded..

Server listening..

server acccept the client...

From client: hi

From client: exit

Server Exit...

**Client Side:-**

Socket successfully created..

connected to the server..

Enter the string : hi

Enter the string : exit

Client Exit..

**Q4.UDP Socket Programming for One Way Communication from client to Server**

**Server:**

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <string.h>

#include <sys/types.h>

#include <sys/socket.h>

#include <arpa/inet.h>

#include <netinet/in.h>

#define PORT     8080

#define MAXLINE 1024

int main() {

    int sockfd;

    char buffer[MAXLINE];

    char \*hello = "Hello from server";

    struct sockaddr\_in servaddr, cliaddr;

if ( (sockfd = socket(AF\_INET, SOCK\_DGRAM, 0)) < 0 ) {

        perror("socket creation failed");

        exit(EXIT\_FAILURE);

    }

    memset(&servaddr, 0, sizeof(servaddr));

    memset(&cliaddr, 0, sizeof(cliaddr));

    servaddr.sin\_family    = AF\_INET; // IPv4

    servaddr.sin\_addr.s\_addr = INADDR\_ANY;

    servaddr.sin\_port = htons(PORT);

if ( bind(sockfd, (const struct sockaddr \*)&servaddr,

            sizeof(servaddr)) < 0 )

    {

        perror("bind failed");

        exit(EXIT\_FAILURE);

    }

    int len, n;

    n = recvfrom(sockfd, (char \*)buffer, MAXLINE,

MSG\_WAITALL, ( struct sockaddr \*) &cliaddr, &len);

    buffer[n] = '\0';

    printf("Client : %s\n", buffer);

    return 0;

}

**Client Side:-**

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <string.h>

#include <sys/types.h>

#include <sys/socket.h>

#include <arpa/inet.h>

#include <netinet/in.h>

#define PORT     8080

#define MAXLINE 1024

int main() {

    int sockfd;

    char buffer[MAXLINE];

    char \*hello = "Hello from client";

    struct sockaddr\_in     servaddr;

if ( (sockfd = socket(AF\_INET, SOCK\_DGRAM, 0)) < 0 ) {

        perror("socket creation failed");

        exit(EXIT\_FAILURE);

    }

    memset(&servaddr, 0, sizeof(servaddr));

    servaddr.sin\_family = AF\_INET;

    servaddr.sin\_port = htons(PORT);

    servaddr.sin\_addr.s\_addr = INADDR\_ANY;

    int n, len;

sendto(sockfd, (const char \*)hello, strlen(hello),

MSG\_CONFIRM, (const struct sockaddr \*) &servaddr, sizeof(servaddr));

    printf("Hello message sent.\n");

    n = recvfrom(sockfd, (char \*)buffer, MAXLINE,

    MSG\_WAITALL, (struct sockaddr \*) &servaddr,

    &len);

    buffer[n] = '\0';

    printf("Server : %s\n", buffer);

    close(sockfd);

    return 0;

}

**Output:**

**Server Side:-**

$ ./server

Client : Hello from client

**Client Side:-**

$ ./client

Hello message sent.