Name – Pranav Patel Roll No. - 68

Div – IT(B) Subject – Computer Network

Assignment - Implement Simple Calculator for basic operations on two operands (+,-,/,\*) using TCP Socket.

**Code**

SERVER :

#include <stdio.h>  
#include <netdb.h>  
#include <netinet/in.h>  
#include <stdlib.h>  
#include <string.h>  
#include <sys/socket.h>  
#include <sys/types.h>  
#include <unistd.h>  
  
#define MAX 80  
#define PORT 8080  
#define SA struct sockaddr  
  
// Function to perform calculator operations  
double calculate(double num1, double num2, char operator) {  
    switch (operator) {  
        case '+':  
            return num1 + num2;  
        case '-':  
            return num1 - num2;  
        case '\*':  
            return num1 \* num2;  
        case '/':  
            return num2 != 0 ? num1 / num2 : 0;  
        default:  
            return 0;  
    }  
}  
  
void performCalculation(int connfd) {  
    char buffer[MAX];  
    char operator;  
    double num1, num2, result;  
  
    for (;;) {  
        bzero(buffer, MAX);  
        read(connfd, buffer, sizeof(buffer));  
  
        if (strncmp(buffer, "exit", 4) == 0) {  
            printf("Client Exit...\n");  
            break;  
        }  
  
        sscanf(buffer, "%lf %c %lf", &num1, &operator, &num2);  
        result = calculate(num1, num2, operator);  
        snprintf(buffer, MAX, "%.2f", result);  
        write(connfd, buffer, sizeof(buffer));  
    }  
}  
  
int main() {  
    int sockfd, connfd, len;  
    struct sockaddr\_in servaddr, cli;  
  
    sockfd = socket(AF\_INET, SOCK\_STREAM, 0);  
    if (sockfd == -1) {  
        perror("socket creation failed");  
        exit(EXIT\_FAILURE);  
    }  
  
    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) {  
        perror("socket bind failed");  
        exit(EXIT\_FAILURE);  
    }  
  
    if (listen(sockfd, 5) != 0) {  
        perror("listen failed");  
        exit(EXIT\_FAILURE);  
    }  
  
    printf("Server listening..\n");  
    len = sizeof(cli);  
  
    connfd = accept(sockfd, (SA\*)&cli, &len);  
    if (connfd < 0) {  
        perror("server accept failed");  
        exit(EXIT\_FAILURE);  
    }  
  
    printf("server accept the client...\n");  
    performCalculation(connfd);  
    close(sockfd);  
  
    return 0;

}

CLIENT :

#include <stdio.h>  
#include <netdb.h>  
#include <netinet/in.h>  
#include <stdlib.h>  
#include <string.h>  
#include <sys/socket.h>  
#include <unistd.h>  
  
#define MAX 80  
#define PORT 8080  
#define SA struct sockaddr  
  
int main() {  
    int sockfd, connfd;  
    struct sockaddr\_in servaddr, cli;  
  
    sockfd = socket(AF\_INET, SOCK\_STREAM, 0);  
    if (sockfd == -1) {  
        perror("socket creation failed");  
        exit(EXIT\_FAILURE);  
    }  
  
    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) {  
        perror("connection with the server failed");  
        exit(EXIT\_FAILURE);  
    }  
  
    char buffer[MAX];  
    printf("Enter the expression (e.g., 2 + 3, 5 - 1, 4 \* 2, 6 / 2):\n");  
    printf("Type 'exit' to quit.\n");  
  
    while (1) {  
        fgets(buffer, MAX, stdin);  
        write(sockfd, buffer, sizeof(buffer));  
  
        if (strncmp(buffer, "exit", 4) == 0) {  
            printf("Client Exit...\n");  
            break;  
        }  
  
        bzero(buffer, MAX);  
        read(sockfd, buffer, sizeof(buffer));  
        printf("Result from server: %s\n", buffer);  
    }  
  
    close(sockfd);  
  
    return 0;  
}

**Output**



