CNT Assignment 3B

Name: Kuber Kishore **Class:** TY CSAI - A

Roll Number: 57 **Batch:** 3

Assignment: Write the client server programs in C using UDP Berkeley socket primitives for wired /wireless network for following

a. to say Hello to Each other

b. Calculator (Trigonometry)

Code

server.c

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <arpa/inet.h>

#define PORT 8080
#define BUF_SIZE 1024

int main() {
    int sockfd;
    char buffer[BUF_SIZE];
    struct sockaddr_in servaddr, cliaddr;
    socklen_t len = sizeof(cliaddr);

    // Create socket
    if ((sockfd = socket(AF_INET, SOCK_DGRAM, 0)) < 0) {
        perror("socket creation failed");
        exit(EXIT_FAILURE);
    }

    // Server info
    memset(&servaddr, 0, sizeof(servaddr));
    servaddr.sin_family = AF_INET;
    servaddr.sin_addr.s_addr = INADDR_ANY;
    servaddr.sin_port = htons(PORT);

    // Bind
    if (bind(sockfd, (const struct sockaddr *)&servaddr, sizeof(servaddr)) <
0) {
        perror("bind failed");
        close(sockfd);
```

```
        exit(EXIT_FAILURE);
    }

    printf("UDP Hello Server listening on port %d...\n", PORT);

    // Receive message
    int n = recvfrom(sockfd, buffer, BUF_SIZE, 0, (struct sockaddr *)&cliaddr,
&len);
    buffer[n] = '\0';
    printf("Client says: %s\n", buffer);

    // Send response
    char *reply = "Hello Client!";
    sendto(sockfd, reply, strlen(reply), 0, (struct sockaddr *)&cliaddr, len);

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

client.c

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <arpa/inet.h>

#define PORT 8080
#define BUF_SIZE 1024

int main() {
    int sockfd;
    char buffer[BUF_SIZE];
    struct sockaddr_in servaddr;
    socklen_t len = sizeof(servaddr);

    // Create socket
    if ((sockfd = socket(AF_INET, SOCK_DGRAM, 0)) < 0) {
        perror("socket creation failed");
        exit(EXIT_FAILURE);
    }

    // Server info
    memset(&servaddr, 0, sizeof(servaddr));
    servaddr.sin_family = AF_INET;
    servaddr.sin_port = htons(PORT);
    servaddr.sin_addr.s_addr = INADDR_ANY;  // localhost

    // Send hello
```

```
    char *hello = "Hello Server!";
    sendto(sockfd, hello, strlen(hello), 0, (const struct sockaddr
*)&servaddr, len);
    printf("Sent to server: %s\n", hello);

    // Receive response
    int n = recvfrom(sockfd, buffer, BUF_SIZE, 0, (struct sockaddr
*)&servaddr, &len);
    buffer[n] = '\0';
    printf("Server says: %s\n", buffer);

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

calc_server.c

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <arpa/inet.h>
#include <math.h>

#define PORT 9090
#define BUF_SIZE 1024

int main() {
    int sockfd;
    char buffer[BUF_SIZE];
    struct sockaddr_in servaddr, cliaddr;
    socklen_t len = sizeof(cliaddr);

    // Create socket
    if ((sockfd = socket(AF_INET, SOCK_DGRAM, 0)) < 0) {
        perror("socket creation failed");
        exit(EXIT_FAILURE);
    }

    // Server info
    memset(&servaddr, 0, sizeof(servaddr));
    servaddr.sin_family = AF_INET;
    servaddr.sin_addr.s_addr = INADDR_ANY;
    servaddr.sin_port = htons(PORT);

    // Bind
    if (bind(sockfd, (const struct sockaddr *)&servaddr, sizeof(servaddr)) <
0) {
        perror("bind failed");
```

```
        close(sockfd);
        exit(EXIT_FAILURE);
    }

    printf("UDP Calculator Server listening on port %d...\n", PORT);

    // Receive request
    int n = recvfrom(sockfd, buffer, BUF_SIZE, 0, (struct sockaddr *)&cliaddr,
&len);
    buffer[n] = '\0';
    printf("Client request: %s\n", buffer);

    double result = 0.0;
    if (strcmp(buffer, "sin 30") == 0) {
        result = sin(30 * M_PI / 180.0);  // Convert to radians
    }

    char reply[BUF_SIZE];
    snprintf(reply, sizeof(reply), "Result: %f", result);

    // Send result
    sendto(sockfd, reply, strlen(reply), 0, (struct sockaddr *)&cliaddr, len);

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

calc_client.c

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <arpa/inet.h>

#define PORT 9090
#define BUF_SIZE 1024

int main() {
    int sockfd;
    char buffer[BUF_SIZE];
    struct sockaddr_in servaddr;
    socklen_t len = sizeof(servaddr);

    // Create socket
    if ((sockfd = socket(AF_INET, SOCK_DGRAM, 0)) < 0) {
        perror("socket creation failed");
        exit(EXIT_FAILURE);
    }
```

```c
    // Server info
    memset(&servaddr, 0, sizeof(servaddr));
    servaddr.sin_family = AF_INET;
    servaddr.sin_port = htons(PORT);
    servaddr.sin_addr.s_addr = INADDR_ANY;

    // Send request
    char *request = "sin 30";
    sendto(sockfd, request, strlen(request), 0, (const struct sockaddr
*)&servaddr, len);
    printf("Sent to server: %s\n", request);

    // Receive response
    int n = recvfrom(sockfd, buffer, BUF_SIZE, 0, (struct sockaddr
*)&servaddr, &len);
    buffer[n] = '\0';
    printf("Server response: %s\n", buffer);

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

Output

Server terminal

root@Kuber:/mnt/d/VIT/TY/Sem V/CNT/Lab/Assignment 3B# gcc server.c -o server

root@Kuber:/mnt/d/VIT/TY/Sem V/CNT/Lab/Assignment 3B# ./server

UDP Hello Server listening on port 8080...

Client says: Hello Server!

Client terminal

root@Kuber:/mnt/d/VIT/TY/Sem V/CNT/Lab/Assignment 3B# gcc client.c -o client

root@Kuber:/mnt/d/VIT/TY/Sem V/CNT/Lab/Assignment 3B# ./client

Sent to server: Hello Server!

Server says: Hello Client!

Calc_Server terminal

root@Kuber:/mnt/d/VIT/TY/Sem V/CNT/Lab/Assignment 3B# gcc calc_server.c -o calc_server

root@Kuber:/mnt/d/VIT/TY/Sem V/CNT/Lab/Assignment 3B# ./calc_server

UDP Calculator Server listening on port 9090...

Client request: sin 30

Calc_Client terminal

root@Kuber:/mnt/d/VIT/TY/Sem V/CNT/Lab/Assignment 3B# gcc calc_client.c -o calc_client

root@Kuber:/mnt/d/VIT/TY/Sem V/CNT/Lab/Assignment 3B# ./calc_client

Sent to server: sin 30

Server response: Result: 0.500000