Pradeep KM
3122225001092

Exercise 2: Echo Client Server

Server algorithm

1. Create a socket.
2. Define server address (IP, port).
3. Bind the socket to the server address.
4. Listen for incoming connections.
5. Accept a connection from the client.
6. Receive the message from the client.
7. Send the received message back to the client (echo).
8. Close the client connection.
9. Close the server socket.

Client algorithm

1. Create a socket.
2. Define the server address (IP, port).
3. Connect to the server.
4. Enter a message to send.
5. Send the message to the server.
6. Receive the echoed message from the server.
7. Print the echoed message.
8. Close the client socket.

server.c

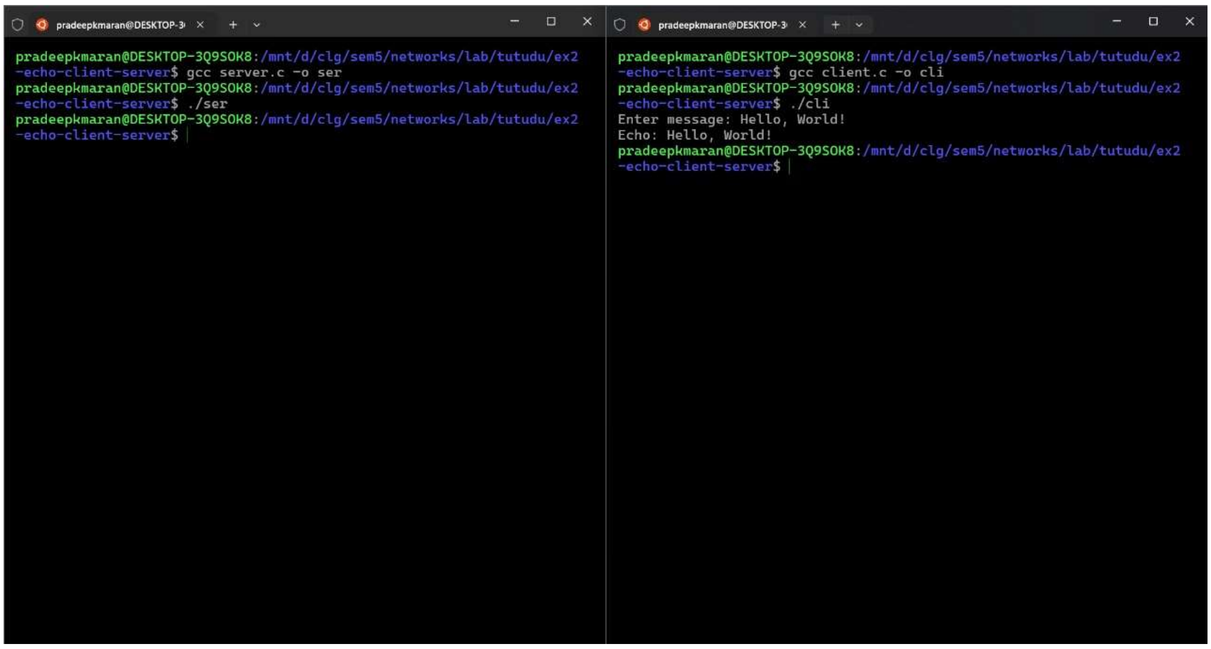
```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>

int main() {
    int server_socket, client_socket;
    struct sockaddr_in server_addr, client_addr;
    socklen_t client_len;
    char buffer[1024];

    server_socket = socket(AF_INET, SOCK_STREAM, 0);
    server_addr.sin_family = AF_INET;
    server_addr.sin_addr.s_addr = INADDR_ANY;
    server_addr.sin_port = htons(8080);
    bind(server_socket, (struct sockaddr *)&server_addr, sizeof(server_addr));
    listen(server_socket, 1);

    client_len = sizeof(client_addr);
    client_socket = accept(server_socket, (struct sockaddr *)&client_addr, &client_len);

    recv(client_socket, buffer, sizeof(buffer), 0);
    send(client_socket, buffer, strlen(buffer), 0);

    close(client_socket);
    close(server_socket);
    return 0;
}
```

client.c

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

int main() {
    int client_socket;
    struct sockaddr_in server_addr;
    char buffer[1024];

    client_socket = socket(AF_INET, SOCK_STREAM, 0);
    server_addr.sin_family = AF_INET;
    server_addr.sin_port = htons(8080);
    server_addr.sin_addr.s_addr = inet_addr("127.0.0.1");

    connect(client_socket, (struct sockaddr *)&server_addr, sizeof(server_addr));

    printf("Enter message: ");
    fgets(buffer, sizeof(buffer), stdin);
    send(client_socket, buffer, strlen(buffer), 0);

    recv(client_socket, buffer, sizeof(buffer), 0);
    printf("Echo: %s", buffer);

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

Pradeep KM
3122225001092

Output



```
pradeepkmaran@DESKTOP-3Q9SOK8:/mnt/d/clg/sem5/networks/lab/tutudu/ex2
-echo-client-server$ gcc server.c -o ser
pradeepkmaran@DESKTOP-3Q9SOK8:/mnt/d/clg/sem5/networks/lab/tutudu/ex2
-echo-client-server$ ./ser
pradeepkmaran@DESKTOP-3Q9SOK8:/mnt/d/clg/sem5/networks/lab/tutudu/ex2
-echo-client-server$
```

```
pradeepkmaran@DESKTOP-3Q9SOK8:/mnt/d/clg/sem5/networks/lab/tutudu/ex2
-echo-client-server$ gcc client.c -o cli
pradeepkmaran@DESKTOP-3Q9SOK8:/mnt/d/clg/sem5/networks/lab/tutudu/ex2
-echo-client-server$ ./cli
Enter message: Hello, World!
Echo: Hello, World!
pradeepkmaran@DESKTOP-3Q9SOK8:/mnt/d/clg/sem5/networks/lab/tutudu/ex2
-echo-client-server$
```