Tcp client

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <unistd.h>

// input for socket programming

#include <arpa/inet.h>

int main()

{

char \*ip = "127.0.0.1";

int port = 5566;

int sock; //socket descriptor

struct sockaddr\_in addr; //structure to store socket address information

socklen\_t addr\_size; //size of the address structure

char buffer[1024]; //an array to store data

int n; //an integer to store the number of bytes received or sent

//socket function. The AF\_INET parameter specifies the address family (IPv4),

//SOCK\_STREAM specifies a TCP socket, and 0 is the protocol (default protocol for TCP)

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

if (sock < 0)

{

perror("[-]Socket error");

exit(1);// If the socket creation fails, an error message is printed, and the program exits.

}

printf("[+]TCP client socket created.\n");

//Initialization of the addr structure. The structure is cleared using memset, and then the address family is set to AF\_INET,

//the port is set to the specified port number, and the IP address is set to the specified IP address using inet\_addr

memset(&addr, '\0', sizeof(addr));

addr.sin\_family = AF\_INET;

addr.sin\_port = port;

addr.sin\_addr.s\_addr = inet\_addr(ip);

//connect function. It takes the socket descriptor, a pointer to the addr structure cast to struct sockaddr \*,

//and the size of the address structure as arguments. If the connection is successful, a message is printed.

connect(sock, (struct sockaddr \*)&addr, sizeof(addr));

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

//Clearing the buffer array, copying a message into it, printing the message,

// and then sending the message to the server using the send function.

bzero(buffer, 1024);

strcpy(buffer, "HELLO, THIS IS CLIENT .");

printf("Client: %s\n", buffer);

send(sock, buffer, strlen(buffer), 0);

//Clearing the buffer array again, receiving a message from the server using the recv function, and printing the received message.

bzero(buffer, 1024);

recv(sock, buffer, sizeof(buffer), 0);

printf("Server: %s\n", buffer);

close(sock);

printf("Disconnected from the server.\n");

return 0;

}