

```
//CSM19031
```

```
#include <stdio.h>
#include <string.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <time.h>
int init(int socket_desc, struct sockaddr_in server_addr, int
server_struct_length)
{

    char server_message[2000], client_message[2000];
    time_t t1, t2;
    do
    {
        memset(server_message, '\0', sizeof(server_message));
        memset(client_message, '\0', sizeof(client_message));
        // Get input from the user:
        printf("Enter message: ");
        gets(client_message);
        ctime(&t1);
        // Send the message to server:
        if (sendto(socket_desc, client_message,
strlen(client_message), 0,
(struct sockaddr *)&server_addr,
server_struct_length) < 0)
        {
            printf("Unable to send message\n");
            return -1;
        }
        memset(server_message, '\0', sizeof(server_message));
        memset(client_message, '\0', sizeof(client_message));

        // Receive the server's response:
        if (recvfrom(socket_desc, server_message,
sizeof(server_message), 0,
(struct sockaddr *)&server_addr,
&server_struct_length) < 0)
        {
            printf("Error while receiving server's msg\n");
            return -1;
        }
        printf("\n\nServer's response: %s\n", server_message);
        memset(server_message, '\0', sizeof(server_message));
        memset(client_message, '\0', sizeof(client_message));
        // Receive the server's response:
        if (recvfrom(socket_desc, (time_t)&t2, sizeof(&t2), 0,
(struct sockaddr *)&server_addr,
&server_struct_length) < 0)
        {
            printf("Error while receiving server's msg\n");
            return -1;
        }
        printf("\n-----\n");
        printf("Time stamp generated at Server -[%s] \nServer's
```

```

response time: %f\n", ctime(&t2), difftime(t2, t1));
    printf("\n-----\n");
}

    } while (strcmp(server_message, "exit") != 0);
    return 0;
}

int main(int argc, char *argv[])
{
    int socket_desc;
    struct sockaddr_in server_addr;
    char server_message[2000], client_message[2000];
    int server_struct_length = sizeof(server_addr);

    // Clean buffers:
    memset(server_message, '\0', sizeof(server_message));
    memset(client_message, '\0', sizeof(client_message));

    // Create socket:
    socket_desc = socket(AF_INET, SOCK_DGRAM, IPPROTO_UDP);

    if (socket_desc < 0)
    {
        printf("Error while creating socket\n");
        return -1;
    }
    printf("Socket created successfully\n");

    // Set port and IP:
    server_addr.sin_family = AF_INET;
    server_addr.sin_port = htons(2000);
    server_addr.sin_addr.s_addr = inet_addr(argv[1]);

    if (init(socket_desc, server_addr, server_struct_length) == 0)
    {
        printf("D END");
        // Close the socket:
        close(socket_desc);
    }

    return 0;
}

```