**09/09/2021 CN LAB 1 2019103573**

**EXCERSISE 1**

**SOCKET PROGRAMMING**

**SERVER.C**

#include <unistd.h>

#include <stdio.h>

#include <sys/socket.h>

#include <stdlib.h>

#include <netinet/in.h>

#include <string.h>

#define MAXSIZE 1024

int main(int argc, char \*argv[])

{

    if (argc < 3)

    {

        printf("Enter the port number and the string to send\n");

        return 1;

    }

    int server\_fd, new\_socket, valread;

    struct sockaddr\_in address;

    int opt = 1;

    int addrlen = sizeof(address);

    char buffer[MAXSIZE] = {0};

    int PORT = atoi(argv[1]);

    char \*msg = argv[2];

    // Creating socket file descriptor

    if ((server\_fd = socket(AF\_INET, SOCK\_STREAM, 0)) == 0)

    {

        perror("socket failed");

        exit(EXIT\_FAILURE);

    }

    // Forcefully attaching socket to the given port

    if (setsockopt(server\_fd, SOL\_SOCKET, SO\_REUSEADDR | SO\_REUSEPORT,

                   &opt, sizeof(opt)))

    {

        perror("setsockopt");

        exit(EXIT\_FAILURE);

    }

    address.sin\_family = AF\_INET;

    address.sin\_addr.s\_addr = INADDR\_ANY;

    address.sin\_port = htons(PORT);

    // binding socket to the port

    if (bind(server\_fd, (struct sockaddr \*)&address,

             sizeof(address)) < 0)

    {

        perror("bind failed");

        exit(EXIT\_FAILURE);

    }

    if (listen(server\_fd, 3) < 0)

    {

        perror("listen");

        exit(EXIT\_FAILURE);

    }

    if ((new\_socket = accept(server\_fd, (struct sockaddr \*)&address,

                             (socklen\_t \*)&addrlen)) < 0)

    {

        perror("accept");

        exit(EXIT\_FAILURE);

    }

    valread = read(new\_socket, buffer, MAXSIZE);

    printf("Message from client: %s\n", buffer);

    send(new\_socket, msg, strlen(msg), 0);

    printf("Message has been sent to the client\n");

    return 0;

}

**CLIENT.C**

#include <stdio.h>

#include <sys/socket.h>

#include <arpa/inet.h>

#include <unistd.h>

#include <string.h>

#include <stdlib.h>

#define MAXSIZE 1024

int main(int argc, char \*argv[])

{

    if (argc < 3)

    {

        printf("Enter the port number and the string to send\n");

        return 1;

    }

    int sock = 0, valread;

    struct sockaddr\_in serv\_addr;

    int PORT = atoi(argv[1]);

    char \*msg = argv[2];

    char buffer[MAXSIZE] = {0};

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

    {

        printf("\n Socket creation error \n");

        return -1;

    }

    serv\_addr.sin\_family = AF\_INET;

    serv\_addr.sin\_port = htons(PORT);

    // Converting addresses from text to binary form

    if (inet\_pton(AF\_INET, "127.0.0.1", &serv\_addr.sin\_addr) <= 0)

    {

        printf("\nInvalid address/ Address not supported \n");

        return -1;

    }

    if (connect(sock, (struct sockaddr \*)&serv\_addr, sizeof(serv\_addr)) < 0)

    {

        printf("\nConnection Failed \n");

        return -1;

    }

    send(sock, msg, strlen(msg), 0);

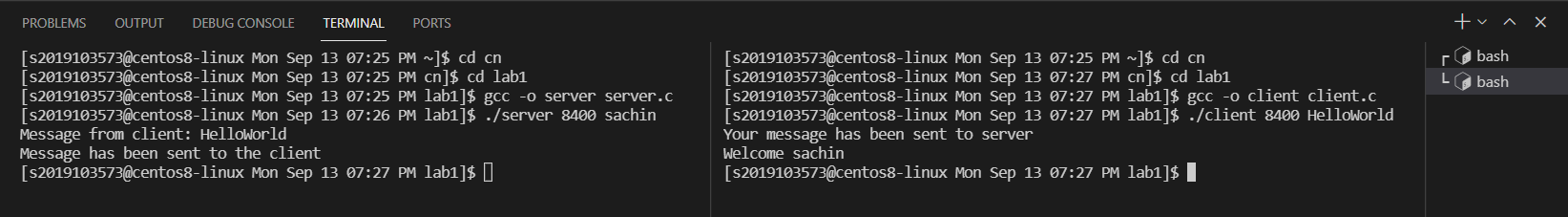
    printf("Your message has been sent to server\n");

    valread = read(sock, buffer, MAXSIZE);

    printf("Welcome %s\n", buffer);

    return 0;

}

**OUTPUT :**

