**Ex: 3 Computer Network Lab**

**Name:** Athithraja R

**Regno:** 2022503702

**UDP:**

**server.c:**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <unistd.h>

#include <sys/types.h>

#include <sys/socket.h>

#include <arpa/inet.h>

#include <netinet/in.h>

#define PORT 8080

#define MAXLINE 1024

int main() {

int sockfd;

char buffer[MAXLINE];

struct sockaddr\_in servaddr, cliaddr;

// Creating socket file descriptor

if ((sockfd = socket(AF\_INET, SOCK\_DGRAM, 0)) < 0) {

perror("Socket creation failed");

exit(EXIT\_FAILURE);

}

memset(&servaddr, 0, sizeof(servaddr));

// Filling server information

servaddr.sin\_family = AF\_INET; // IPv4

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

servaddr.sin\_port = htons(PORT);

// Bind the socket with the server address

if (bind(sockfd, (const struct sockaddr \*)&servaddr, sizeof(servaddr)) < 0) {

perror("Bind failed");

close(sockfd);

exit(EXIT\_FAILURE);

}

socklen\_t len;

int n;

len = sizeof(cliaddr); // len is value/result

while (1) {

n = recvfrom(sockfd, (char \*)buffer, MAXLINE, MSG\_WAITALL, (struct sockaddr \*)&cliaddr, &len);

buffer[n] = '\0';

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

// Send response back to client

sendto(sockfd, (const char \*)buffer, strlen(buffer), MSG\_CONFIRM, (const struct sockaddr \*)&cliaddr, len);

}

close(sockfd);

return 0;

}

Output:



**client.c:**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <unistd.h>

#include <sys/types.h>

#include <sys/socket.h>

#include <arpa/inet.h>

#include <netinet/in.h>

#define PORT 8080

#define MAXLINE 1024

int main() {

int sockfd;

char buffer[MAXLINE];

const char \*message;

struct sockaddr\_in servaddr;

// Creating socket file descriptor

if ((sockfd = socket(AF\_INET, SOCK\_DGRAM, 0)) < 0) {

perror("Socket creation failed");

exit(EXIT\_FAILURE);

}

memset(&servaddr, 0, sizeof(servaddr));

// Filling server information

servaddr.sin\_family = AF\_INET; // IPv4

servaddr.sin\_port = htons(PORT);

servaddr.sin\_addr.s\_addr = inet\_addr("127.0.0.1"); // Server IP

while (1) {

printf("Enter message: ");

fgets(buffer, MAXLINE, stdin);

message = buffer;

// Send message to server

sendto(sockfd, (const char \*)message, strlen(message), MSG\_CONFIRM, (const struct sockaddr \*)&servaddr, sizeof(servaddr));

// Receive response from server

socklen\_t len = sizeof(servaddr);

int n = recvfrom(sockfd, (char \*)buffer, MAXLINE, MSG\_WAITALL, (struct sockaddr \*)&servaddr, &len);

buffer[n] = '\0';

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

}

close(sockfd);

return 0;

}

**Output:**

