Name: Shivani Suresh

Register Number:2021503050

Semester:5

Batch:2

Course: CS6111- Computer Networks

Lab – Experiment 4- Chat Application using UDP

1. Using UDP

Code:

Client:

#include <iostream>

#include <cstdlib>

#include <cstdio>

#include <cstring>

#include <string>

#include <sys/types.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <arpa/inet.h>

#include <unistd.h>

#define PORT 12024

#define MAXLINE 1024

#ifndef EXIT\_FAILURE

#define EXIT\_FAILURE 1

#endif

int main() {

int sockfd;

char buffer[MAXLINE];

struct sockaddr\_in servaddr, cliaddr;

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

perror("socket creation failed");

return EXIT\_FAILURE;

}

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

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

servaddr.sin\_family = AF\_INET;

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

servaddr.sin\_port = htons(PORT);

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

perror("bind failed");

return EXIT\_FAILURE;

}

socklen\_t len;

len = sizeof(cliaddr);

while (true) {

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

buffer[n] = '\0';

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

std::string response;

std::cout << "Server: ";

std::getline(std::cin, response);

sendto(sockfd, response.c\_str(), response.length(), MSG\_CONFIRM, (const struct sockaddr \*)&cliaddr, len);

}

return 0;

}

Server:

#include <iostream>

#include <cstring>

#include <sys/types.h>

#include <sys/socket.h>

#include <netinet/in.h>

#include <unistd.h>

#include <arpa/inet.h>

#include <cstdio>

#define PORT 12024

#define MAXLINE 1024

int main() {

int sockfd;

char buffer[MAXLINE];

struct sockaddr\_in servaddr;

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

perror("socket creation failed");

return 1;

}

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

servaddr.sin\_family = AF\_INET;

servaddr.sin\_port = htons(PORT);

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

socklen\_t len;

while (true) {

char message[MAXLINE];

std::cout << "Client: ";

std::cin.getline(message, MAXLINE);

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

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

buffer[n] = '\0';

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

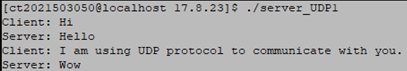
}

return 0;

}

Output:

Server side:



Client side:

