

Program

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.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 argc, char *argv[]) {

    int len, n;
    int sockfd;
    char buffer[1024];
    char *hello = "Hello from server";
    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);
    }

    // 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");
        exit(EXIT_FAILURE);
    }

    len = sizeof(cliaddr); //len is value/result
    n = recvfrom(sockfd, (char *)buffer, MAXLINE, MSG_WAITALL, ( struct sockaddr *) &cliaddr, &len);
    buffer[n] = '\0';
    printf("Message Length=%d \n Client : %s \n", n, buffer);
    sendto(sockfd, (const char *)hello, strlen(hello), MSG_CONFIRM, (const struct sockaddr *) &cliaddr, len);
    printf("Hello message sent.\n");

    return 0;
} //-----client-----//
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
```

```

#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <netinet/in.h>

#define PORT    8080
#define MAXLINE 1020
int main(int argc,char *argv[]) {

    int n, len;
    int sockfd;
    char buffer[1024];
    char *hello = "Hello from client";
    struct sockaddr_in    servaddr;

    // Creating socket file descriptor
    if ( (sockfd = socket(AF_INET, SOCK_DGRAM, 0)) < 0 ) {
        perror("socket creation failed");
        exit(EXIT_FAILURE);
    }

    // Filling server information
    servaddr.sin_family = AF_INET;
    servaddr.sin_port = htons(PORT);

    if (inet_pton(AF_INET,"172.16.2.83", &servaddr.sin_addr)<= 0) {
printf("\nInvalid address/ Address not supported \n");
return -1;
    }

    sendto(sockfd, (const char *)hello, strlen(hello),MSG_CONFIRM, (const struct sockaddr *) &servaddr,sizeof(servaddr));
    printf("Hello message sent.\n");
    n = recvfrom(sockfd, (char *)buffer, MAXLINE,MSG_WAITALL, (struct sockaddr *) &servaddr,&len);
    printf("Message Length:%d",n);
    buffer[n] = '\0';
    printf("Server : %s\n", buffer);

    close(sockfd);
    return 0;
}

```

Output

- Recent
- Starred
- Home
- Desktop
- Documents
- Downloads
- Music
- Pictures
- Videos
- Trash
- Documents
- Music
- Pictures
- Videos
- Downloads
- Other Location

Name



a.out



client.c



udpserver.c



cek21cs049@ltsp595: ~/s6networklab/

```
cek21cs049@ltsp595:~/s6networklab/udp$ gcc client.c
cek21cs049@ltsp595:~/s6networklab/udp$ ./a.out
Hello message sent.
Message Length:17Server : Hello from server
cek21cs049@ltsp595:~/s6networklab/udp$
```



```
cek21cs049@ltsp595:~
(gedit:5456): dbind-
lity bus address: or
ice is masked.
(gedit:5456): IBUS-W
ig/ibus/bus is not c
cek21cs049@ltsp595:~
(gedit:5493): dbind-
lity bus address: or
ice is masked.
(gedit:5493): IBUS-W
ig/ibus/bus is not c
cek21cs049@ltsp595:~
cek21cs049@ltsp595:~
Message Length=17
Client : Hello from
Hello message sent.
cek21cs049@ltsp595:~
```