

```

/*udp broadcast*/

//udp broadcast server
#include <iostream>
#include <string>
#include <cstring>
#include <unistd.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>

#define PORT 8080
#define BUFFER_SIZE 1024

int main() {
    int sockfd;

    struct sockaddr_in servaddr, cliaddr;

    char buffer[BUFFER_SIZE];

    socklen_t len = sizeof(cliaddr);

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

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

    // Filling server information
    servaddr.sin_family = AF_INET;
    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) {
    std::cerr << "bind failed" << std::endl;
    exit(EXIT_FAILURE);
}

std::cout << "Server listening for broadcast on port " << PORT << std::endl;

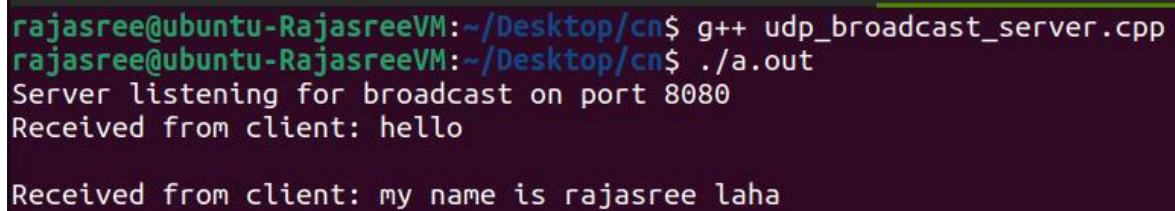
while (1) {
    int n = recvfrom(sockfd, (char *)buffer, BUFFER_SIZE, MSG_WAITALL, (struct sockaddr *)&cliaddr,
    &len);

    buffer[n] = '\0';

    std::cout << "Received from client: " << buffer << std::endl;
}

return 0;
}

```



```

rajasree@ubuntu-RajasreeVM:~/Desktop/cn$ g++ udp_broadcast_server.cpp
rajasree@ubuntu-RajasreeVM:~/Desktop/cn$ ./a.out
Server listening for broadcast on port 8080
Received from client: hello

Received from client: my name is rajasree laha

```

```

//udp broadcast client
#include <iostream>
#include <string>
#include <cstring>
#include <unistd.h>
#include <sys/socket.h>

```

```
#include <netinet/in.h>

#include <arpa/inet.h>

#define PORT 8080

#define BUFFER_SIZE 1024

int main() {
    int sockfd;

    struct sockaddr_in servaddr;

    char buffer[BUFFER_SIZE];

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

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

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

    int broadcastEnable = 1;
    if (setsockopt(sockfd, SOL_SOCKET, SO_BROADCAST, &broadcastEnable, sizeof(broadcastEnable)) <
0) {
        std::cerr << "setsockopt failed" << std::endl;
        close(sockfd);
        exit(EXIT_FAILURE);
    }
```

```
while (1) {  
    std::cout << "Enter message to broadcast: ";  
    fgets(buffer, BUFFER_SIZE, stdin);  
  
    sendto(sockfd, (const char *)buffer, strlen(buffer), MSG_CONFIRM, (const struct sockaddr  
*)&servaddr, sizeof(servaddr));  
}  
  
close(sockfd);  
return 0;  
}
```

```
rajasree@ubuntu-RajasreeVM:~/Desktop/cn$ g++ udp_broadcast_client.cpp  
rajasree@ubuntu-RajasreeVM:~/Desktop/cn$ ./a.out  
Enter message to broadcast: hello  
Enter message to broadcast: my name is rajasree laha
```