

Server.cpp

```
#include <iostream>
```

```
#include <cstring>
```

```
#include <unistd.h>
```

```
#include <arpa/inet.h>
```

```
using namespace std;
```

```
#define PORT 8080
```

```
#define MAXLINE 1024
```

```
int main() {
```

```
    int sockfd;
```

```
    char buffer[MAXLINE];
```

```
    struct sockaddr_in servaddr, cliaddr;
```

```
    const char *hello = "Hello from server";
```

```
    // Create socket
```

```
    if ((sockfd = socket(AF_INET, SOCK_DGRAM, 0)) < 0) {
```

```
        perror("Socket creation failed");
```

```
        return 1;
```

```
    }
```

```
    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 1;
    }

    socklen_t len = sizeof(cliaddr);
    int n = recvfrom(sockfd, buffer, MAXLINE, 0, (struct sockaddr *)&cliaddr, &len);
    buffer[n] = '\0';

    cout << "Client: " << buffer << endl;

    sendto(sockfd, hello, strlen(hello), 0, (const struct sockaddr *)&cliaddr, len);
    cout << "Server: Hello message sent to client." << endl;

    close(sockfd);
    return 0;
}

```

Client.cpp

```

#include <iostream>
#include <cstring>
#include <unistd.h>
#include <arpa/inet.h>
using namespace std;

#define PORT 8080
#define MAXLINE 1024

int main() {
    int sockfd;
    char buffer[MAXLINE];
    struct sockaddr_in servaddr;
    const char *hello = "Hello from client";

```

```

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);
inet_pton(AF_INET, "127.0.0.1", &servaddr.sin_addr);

sendto(sockfd, hello, strlen(hello), 0, (const struct sockaddr *)&servaddr, sizeof(servaddr));
cout << "Client: Hello message sent to server." << endl;

socklen_t len = sizeof(servaddr);
int n = recvfrom(sockfd, buffer, MAXLINE, 0, (struct sockaddr *)&servaddr, &len);
buffer[n] = '\0';
cout << "Server: " << buffer << endl;

close(sockfd);
return 0;
}

```

Output:

```

ise@ise-VirtualBox:~/Desktop$ g++ server.cpp -o server
ise@ise-VirtualBox:~/Desktop$ g++ client.cpp -o client
ise@ise-VirtualBox:~/Desktop$ ./server
Client: Hello from client
Server: Hello message sent to client.
ise@ise-VirtualBox:~/Desktop$ 
ise@ise-VirtualBox:~/Desktop$ ./client
Client: Hello message sent to server.
Server: Hello from server
ise@ise-VirtualBox:~/Desktop$ 

```