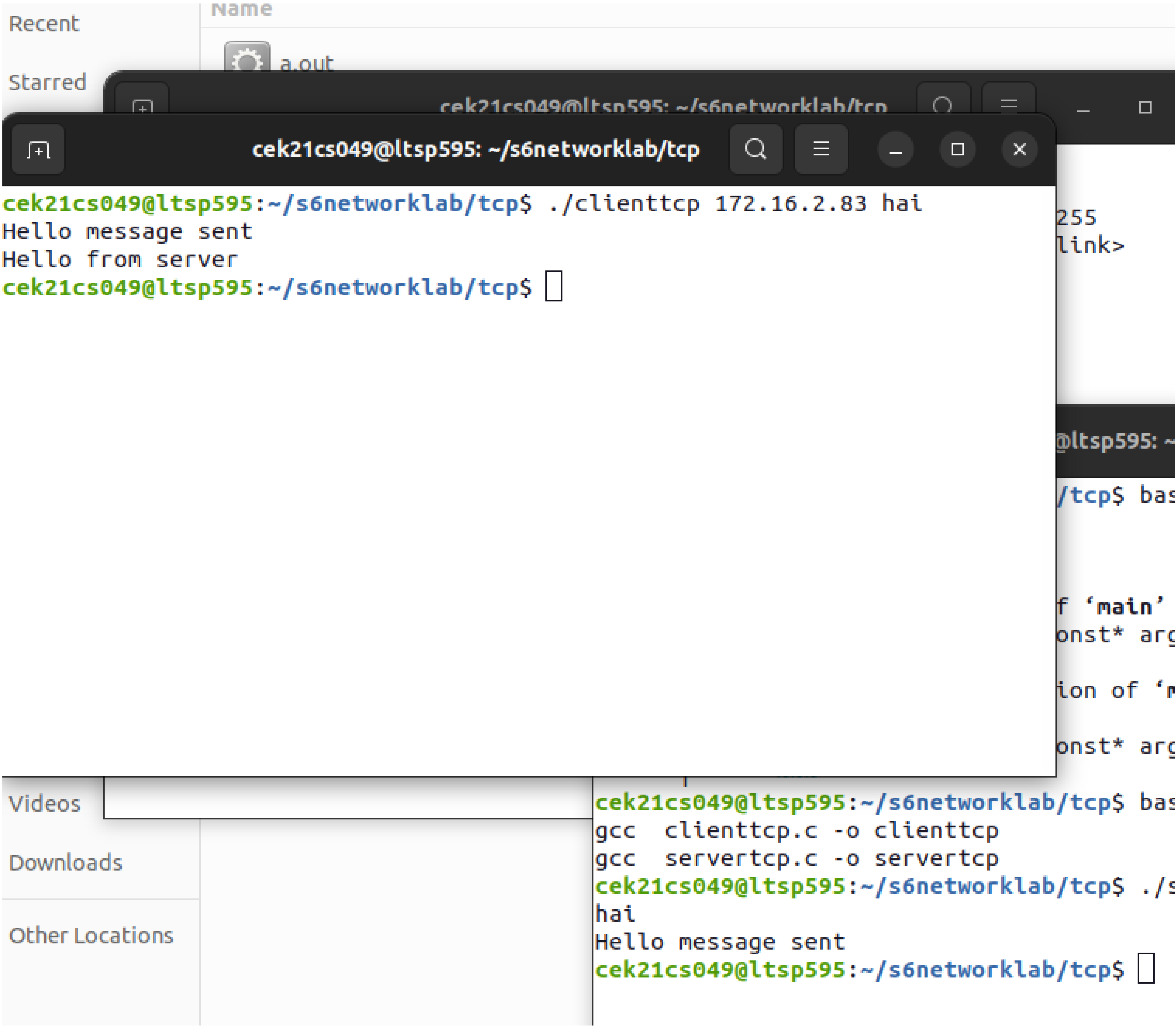
Program

```
#include <arpa/inet.h>
#include <stdio.h>
#include <string.h>
#include <sys/socket.h>
#include <unistd.h>
#define PORT 8080

int main(int argc, char const* argv[])
{
int sock = 0, valread, client_fd;
struct sockaddr_in serv_addr;
//char* hello = "Hello";
char buffer[1024] = { 0 };
char const *ip=argv[1];
char const *msg=argv[2];
if ((sock = socket(AF_INET, SOCK_STREAM, 0)) < 0) {
printf("\n Socket creation error \n");
return -1;
}

serv_addr.sin_family = AF_INET;
serv_addr.sin_port = htons(PORT);

// Convert IPv4 and IPv6 addresses from text to binary
// form
if (inet_pton(AF_INET,ip, &serv_addr.sin_addr)<= 0) {
printf("\nInvalid address/ Address not supported \n");
return -1;
}

if ((client_fd= connect(sock, (struct sockaddr*)&serv_addr,sizeof(serv_addr)))< 0) {
printf("\nConnection Failed \n");
return -1;
}

send(sock,msg, strlen(msg), 0);
printf("Hello message sent\n");
valread = read(sock, buffer, 1024);
printf("%s\n", buffer);

// closing the connected socket
close(client_fd);
return 0;
}
//-------------------------------------------------------server--------------------------------------------------------------------//
#include <netinet/in.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/socket.h>
#include <unistd.h>
```

```
#define PORT 8080
int main(int argc, char const* argv[])
{
int server_fd, new_socket, valread;
struct sockaddr_in address;
int opt = 1;
int addrlen = sizeof(address);
char buffer[1024] = { 0 };
char* hello = "Hello from server";

// Creating socket file descriptor
if ((server_fd = socket(AF_INET, SOCK_STREAM, 0))== 0) {
perror("socket failed");
exit(EXIT_FAILURE);
}

address.sin_family = AF_INET;
address.sin_addr.s_addr = INADDR_ANY;
address.sin_port = htons(PORT);

// Forcefully attaching socket to the port 8080
if (bind(server_fd, (struct sockaddr*)&address,sizeof(address))< 0) {
perror("bind failed");
exit(EXIT_FAILURE);
}

if (listen(server_fd, 3) < 0) {
perror("listen");
exit(EXIT_FAILURE);
}

if ((new_socket= accept(server_fd, (struct sockaddr*)&address,(socklen_t*)&addrlen))< 0) {
perror("accept");
exit(EXIT_FAILURE);
}

valread = read(new_socket, buffer, 1024);
printf("%s\n", buffer);
send(new_socket, hello, strlen(hello), 0);
printf("Hello message sent\n");

// closing the connected socket
close(new_socket);
// closing the listening socket
shutdown(server_fd, SHUT_RDWR);
return 0;
}
```

Output


Recent
Starred
Name
a.out
cek21cs049@ltsp595: ~/s6networklab/tcp
cek21cs049@ltsp595: ~/s6networklab/tcp
@ltsp595: ~
Videos
Downloads
Other Locations

```
cek21cs049@ltsp595:~/s6networklab/tcp$ ./clienttcp 172.16.2.83 hai
Hello message sent
Hello from server
cek21cs049@ltsp595:~/s6networklab/tcp$ 
```

```
255
link>

/tcp$ bas

f ‘main’
onst* arg

ion of ‘m

onst* arg

cek21cs049@ltsp595:~/s6networklab/tcp$ bas
gcc  clienttcp.c -o clienttcp
gcc  servertcp.c -o servertcp
cek21cs049@ltsp595:~/s6networklab/tcp$ ./s
hai
Hello message sent
cek21cs049@ltsp595:~/s6networklab/tcp$ 
```