Program

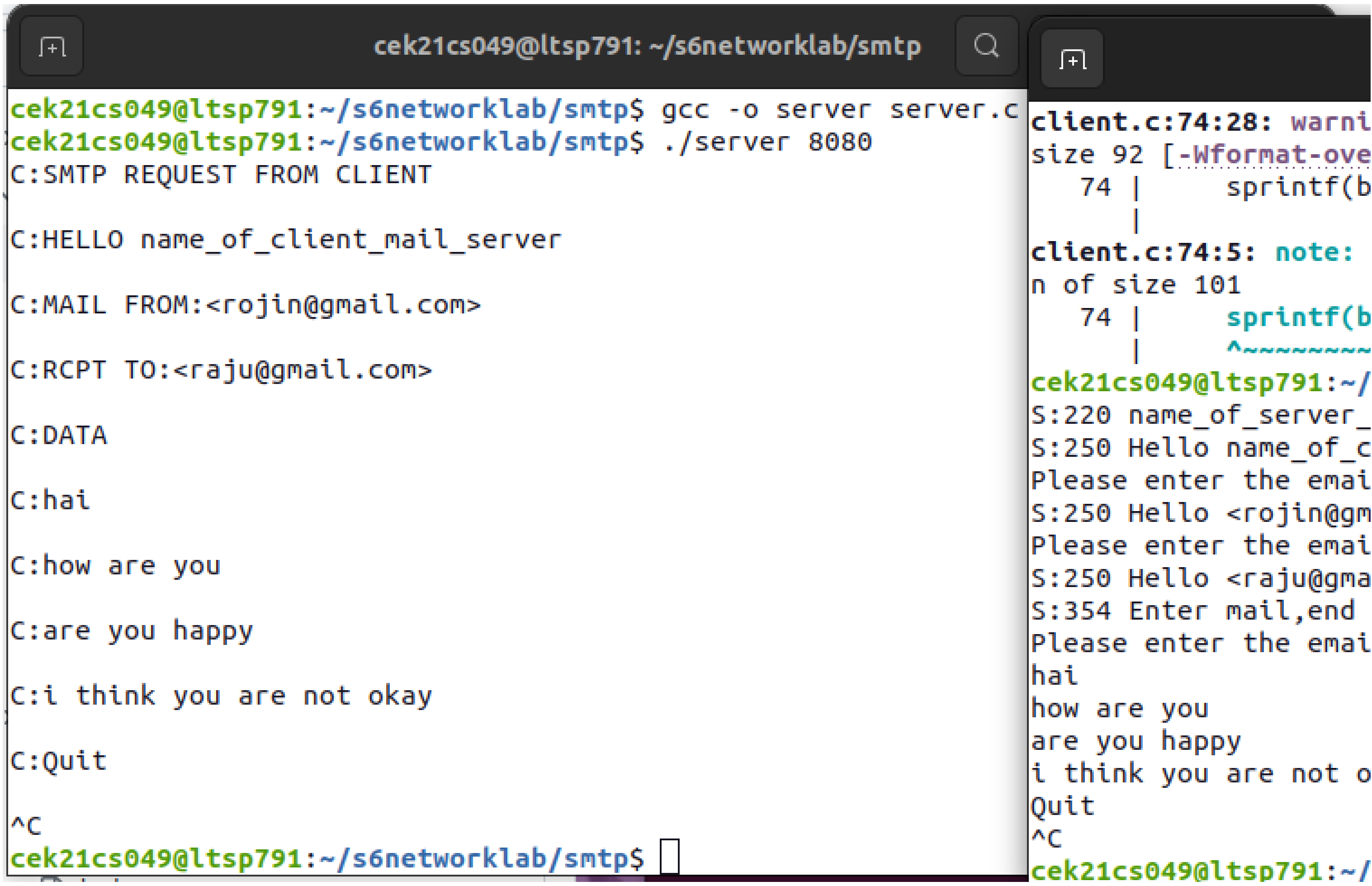
```
//-------------------server-------------------------------------------
#include <stdio.h>
#include <string.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <fcntl.h>
#include <stdlib.h>
#include <ctype.h>

#define MAXLINE 100

int main(int argc, char *argv[]) {
    int n, sock_fd;
    struct sockaddr_in servaddr, cliaddr;
    char mesg[MAXLINE + 1];
    socklen_t len;
    char *str_ptr, *buf_ptr, *str;

    if (argc != 2) {
        fprintf(stderr, "Usage: %s port\n", argv[0]);
        exit(1);
    }

    len = sizeof(cliaddr);
    if ((sock_fd = socket(AF_INET, SOCK_DGRAM, 0)) < 0) {
        printf("Cannot create socket\n");
        exit(1);
    }

    bzero((char *)&servaddr, sizeof(servaddr));
    servaddr.sin_family = AF_INET;
    servaddr.sin_port = htons(atoi(argv[1]));
    servaddr.sin_addr.s_addr = htonl(INADDR_ANY);
    if (bind(sock_fd, (struct sockaddr *)&servaddr, sizeof(servaddr)) < 0) {
        perror("bind failed:");
        exit(1);
    }

    if ((n = recvfrom(sock_fd, mesg, MAXLINE, 0, (struct sockaddr *)&cliaddr, &len)) == -1) {
        perror("size not received:");
        exit(1);
    }
    mesg[n] = '\0';
    printf("C:%s\n", mesg);
    sprintf(mesg, "220 name_of_server_mail_server\n");
    sendto(sock_fd, mesg, strlen(mesg), 0, (struct sockaddr *)&cliaddr, sizeof(cliaddr));

    if ((n = recvfrom(sock_fd, mesg, MAXLINE, 0, (struct sockaddr *)&cliaddr, &len)) == -1) {
        perror("UDP read error");
    exit(1);
}
mesg[n] = '\0';
printf("C:%s\n", mesg);
str_ptr = strdup(mesg);
buf_ptr = strsep(&str_ptr, " ");
sprintf(mesg, "250 Hello %s", str_ptr);
free(buf_ptr);
sendto(sock_fd, mesg, strlen(mesg), 0, (struct sockaddr *)&cliaddr, sizeof(cliaddr));

if ((n = recvfrom(sock_fd, mesg, MAXLINE, 0, (struct sockaddr *)&cliaddr, &len)) == -1) {
    perror("UDP read error");
    exit(1);
}
mesg[n] = '\0';
printf("C:%s\n", mesg);
str_ptr = strdup(mesg);
buf_ptr = strsep(&str_ptr, ":");
str_ptr[strlen(str_ptr) - 1] = '\0';

sprintf(mesg, "250 Hello %s.......sender ok\n", str_ptr);
free(buf_ptr);
sendto(sock_fd, mesg, strlen(mesg), 0, (struct sockaddr *)&cliaddr, sizeof(cliaddr));

if ((n = recvfrom(sock_fd, mesg, MAXLINE, 0, (struct sockaddr *)&cliaddr, &len)) == -1) {
    perror("UDP read error");
    exit(1);
}
mesg[n] = '\0';
printf("C:%s\n", mesg);
str_ptr = strdup(mesg);
buf_ptr = strsep(&str_ptr, ":");
str_ptr[strlen(str_ptr) - 1] = '\0';
sprintf(mesg, "250 Hello %s.......Recepient ok\n", str_ptr);
free(buf_ptr);
sendto(sock_fd, mesg, strlen(mesg), 0, (struct sockaddr *)&cliaddr, sizeof(cliaddr));

if ((n = recvfrom(sock_fd, mesg, MAXLINE, 0, (struct sockaddr *)&cliaddr, &len)) == -1) {
    perror("UDP read error");
    exit(1);
}
mesg[n] = '\0';
printf("C:%s\n", mesg);
sprintf(mesg, "354 Enter mail,end with \".\" on a line by itself \n");
sendto(sock_fd, mesg, strlen(mesg), 0, (struct sockaddr *)&cliaddr, sizeof(cliaddr));

while (1) {
    n = recvfrom(sock_fd, mesg, MAXLINE, 0, (struct sockaddr *)&cliaddr, &len);
    mesg[n] = '\0';
    printf("C:%s\n", mesg);
    mesg[strlen(mesg) - 1] = '\0';

    str = mesg;
```

```
        while (isspace(*str++));
        if (strcmp(--str, ".") == 0)
            break;
        sprintf(mesg, "250 messages accepted for delivery \n");
        sendto(sock_fd, mesg, strlen(mesg), 0, (struct sockaddr *)&cliaddr, sizeof(cliaddr));
        n = recvfrom(sock_fd, mesg, MAXLINE, 0, (struct sockaddr *)&cliaddr, &len);
        mesg[n] = '\0';
        printf("C:%s\n", mesg);
    }

    sprintf(mesg, "221 servers mail server closing connection\n");
    sendto(sock_fd, mesg, strlen(mesg), 0, (struct sockaddr *)&cliaddr, sizeof(cliaddr));

    return 0;
}




//------------------client---------------------------------------------
#include <stdio.h>
#include <string.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <fcntl.h>
#include <stdlib.h>
#include <ctype.h>

#define MAXLINE 100

int main(int argc, char *argv[]) {
    int n;
    int sock_fd;
    struct sockaddr_in servaddr;
    char buf[MAXLINE + 1];
    char address_buf[MAXLINE], message_buf[MAXLINE];
    char *str;

    if (argc != 3) {
        fprintf(stderr, "Command is: ./client address port\n");
        exit(1);
    }

    if ((sock_fd = socket(AF_INET, SOCK_DGRAM, 0)) < 0) {
        printf("Cannot create socket\n");
        exit(1);
    }

    bzero((char *)&servaddr, sizeof(servaddr));
    servaddr.sin_family = AF_INET;
    servaddr.sin_port = htons(atoi(argv[2]));
```

```
inet_pton(AF_INET, argv[1], &servaddr.sin_addr);

sprintf(buf, "SMTP REQUEST FROM CLIENT\n");
n = sendto(sock_fd, buf, strlen(buf), 0, (struct sockaddr *)&servaddr, sizeof(servaddr));
if (n < 0) {
    perror("ERROR");
    exit(1);
}

if ((n = recvfrom(sock_fd, buf, MAXLINE, 0, NULL, NULL)) == -1) {
    perror("UDP read error");
    exit(1);
}
buf[n] = '\0';
printf("S:%s", buf);

sprintf(buf, "HELLO name_of_client_mail_server\n");
n = sendto(sock_fd, buf, strlen(buf), 0, (struct sockaddr *)&servaddr, sizeof(servaddr));
if ((n = recvfrom(sock_fd, buf, MAXLINE, 0, NULL, NULL)) == -1) {
    perror("UDP read error");
    exit(1);
}
buf[n] = '\0';
printf("S:%s", buf);

printf("Please enter the email address of the sender: ");
fgets(address_buf, sizeof(address_buf), stdin);
address_buf[strlen(address_buf) - 1] = '\0';
sprintf(buf, "MAIL FROM:<%s>\n", address_buf);
sendto(sock_fd, buf, strlen(buf), 0, (struct sockaddr *)&servaddr, sizeof(servaddr));
if ((n = recvfrom(sock_fd, buf, MAXLINE, 0, NULL, NULL)) == -1) {
    perror("UDP read error");
    exit(1);
}
buf[n] = '\0';
printf("S:%s", buf);

printf("Please enter the email address of the receiver: ");
fgets(address_buf, sizeof(address_buf), stdin);
address_buf[strlen(address_buf) - 1] = '\0';
sprintf(buf, "RCPT TO:<%s>\n", address_buf);
sendto(sock_fd, buf, strlen(buf), 0, (struct sockaddr *)&servaddr, sizeof(servaddr));
if ((n = recvfrom(sock_fd, buf, MAXLINE, 0, NULL, NULL)) == -1) {
    perror("UDP read error");
    exit(1);
}
buf[n] = '\0';
printf("S:%s", buf);

sprintf(buf, "DATA\n");
sendto(sock_fd, buf, strlen(buf), 0, (struct sockaddr *)&servaddr, sizeof(servaddr));
if ((n = recvfrom(sock_fd, buf, MAXLINE, 0, NULL, NULL)) == -1) {
    perror("UDP read error");
```

```
        exit(1);
    }
    buf[n] = '\0';
    printf("S:%s", buf);

    printf("Please enter the email message:\n");
    do {
        fgets(message_buf, sizeof(message_buf), stdin);
        sprintf(buf, "%s", message_buf);
        sendto(sock_fd, buf, strlen(buf), 0, (struct sockaddr *)&servaddr, sizeof(servaddr));
        message_buf[strlen(message_buf) - 1] = '\0';
        str = message_buf;
        while (isspace(*str++));
        if (strcmp(--str, ".") == 0)
            break;
    } while (1);

    if ((n = recvfrom(sock_fd, buf, MAXLINE, 0, NULL, NULL)) == -1) {
        perror("UDP read error");
        exit(1);
    }
    buf[n] = '\0';

    sprintf(buf, "QUIT\n");
    printf("S:%s", buf);
    sendto(sock_fd, buf, strlen(buf), 0, (struct sockaddr *)&servaddr, sizeof(servaddr));

    if ((n = recvfrom(sock_fd, buf, MAXLINE, 0, NULL, NULL)) == -1) {
        perror("UDP read error");
        exit(1);
    }
    buf[n] = '\0';
    printf("S:%s", buf);

    return 0;
}
```

Output



```
cek21cs049@ltsp791:~/s6networklab/smtp$ gcc -o server server.c
cek21cs049@ltsp791:~/s6networklab/smtp$ ./server 8080
C:SMTP REQUEST FROM CLIENT

C:HELLO name_of_client_mail_server

C:MAIL FROM:<rojin@gmail.com>

C:RCPT TO:<raju@gmail.com>

C:DATA

C:hai

C:how are you

C:are you happy

C:i think you are not okay

C:Quit

^C
cek21cs049@ltsp791:~/s6networklab/smtp$ 
```

```
client.c:74:28: warni
size 92 [-Wformat-ove
   74 |     sprintf(b
      |
client.c:74:5: note: 
n of size 101
   74 |     sprintf(b
      |     ^~~~~~~~~
cek21cs049@ltsp791:~/
S:220 name_of_server_
S:250 Hello name_of_c
Please enter the emai
S:250 Hello <rojin@gm
Please enter the emai
S:250 Hello <raju@gma
S:354 Enter mail,end 
Please enter the emai
hai
how are you
are you happy
i think you are not o
Quit
^C
cek21cs049@ltsp791:~/
```