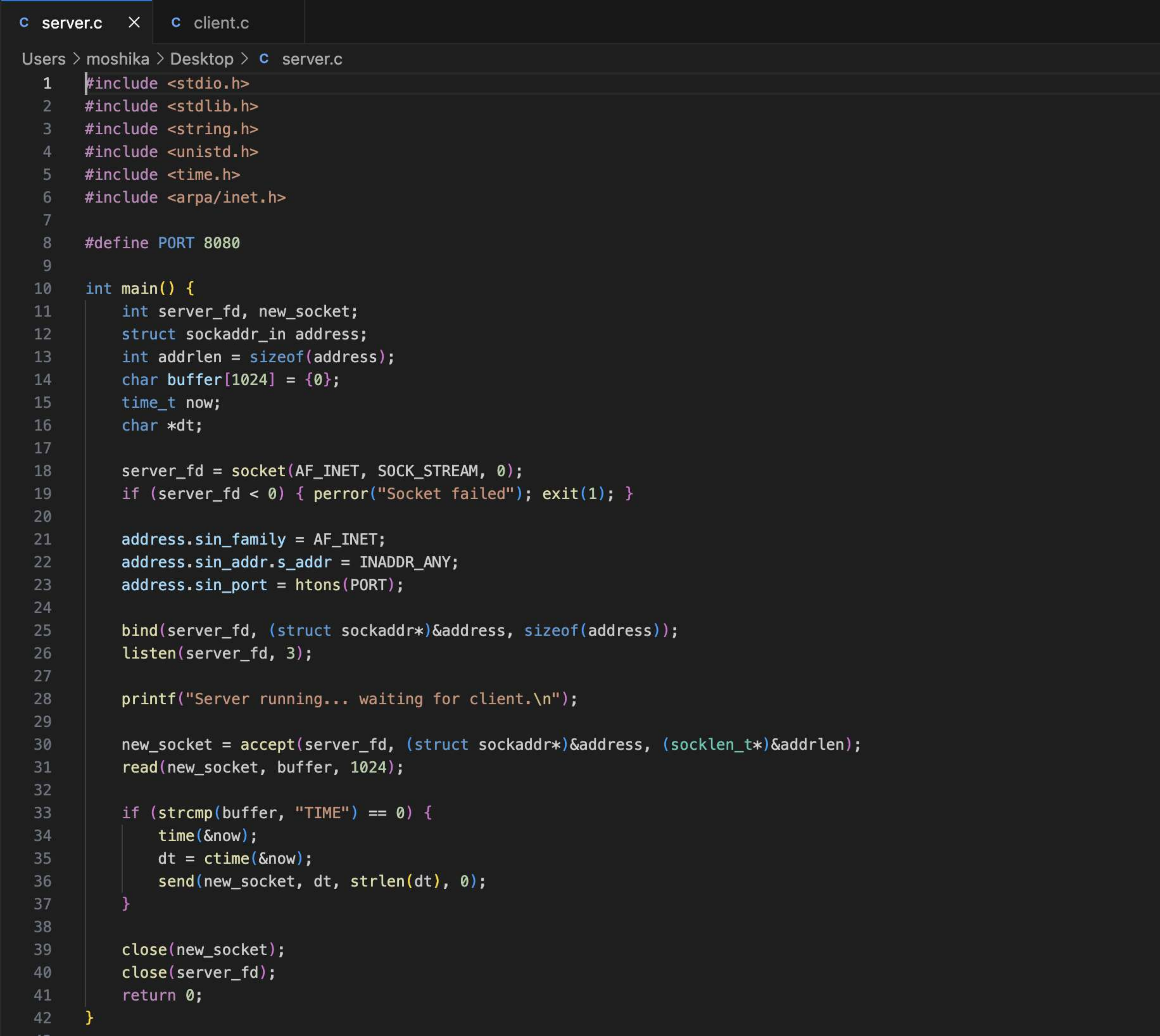
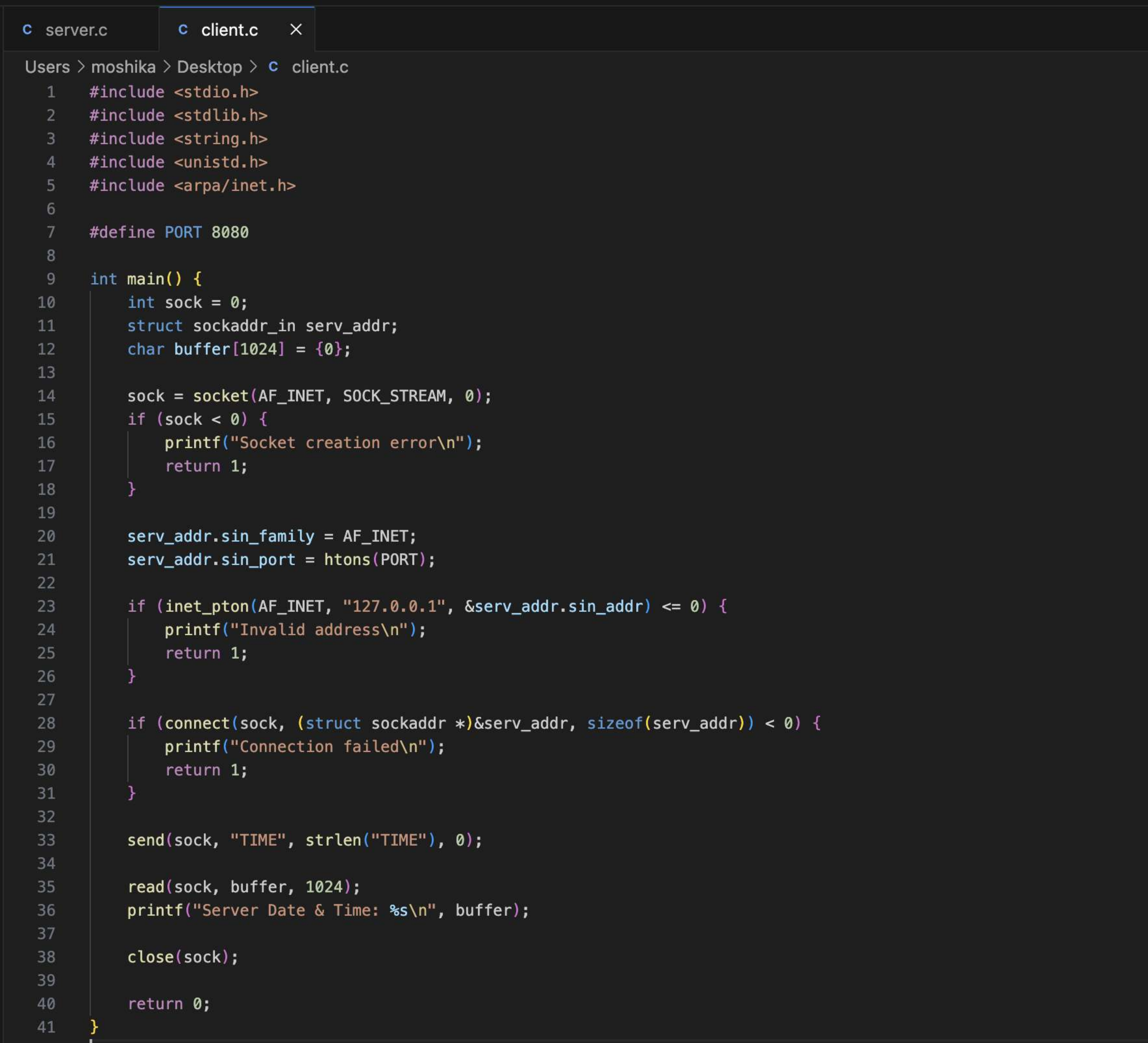
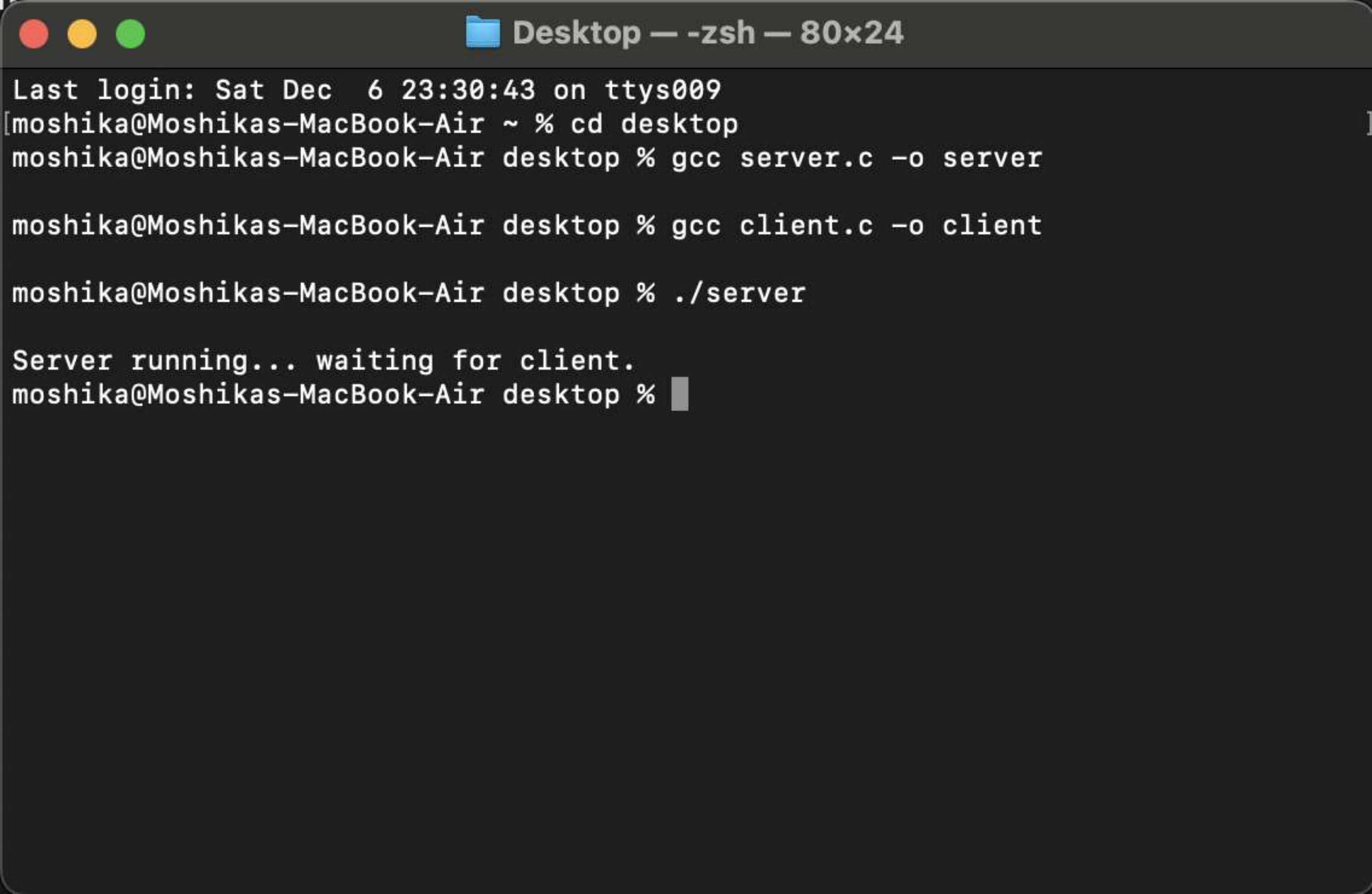


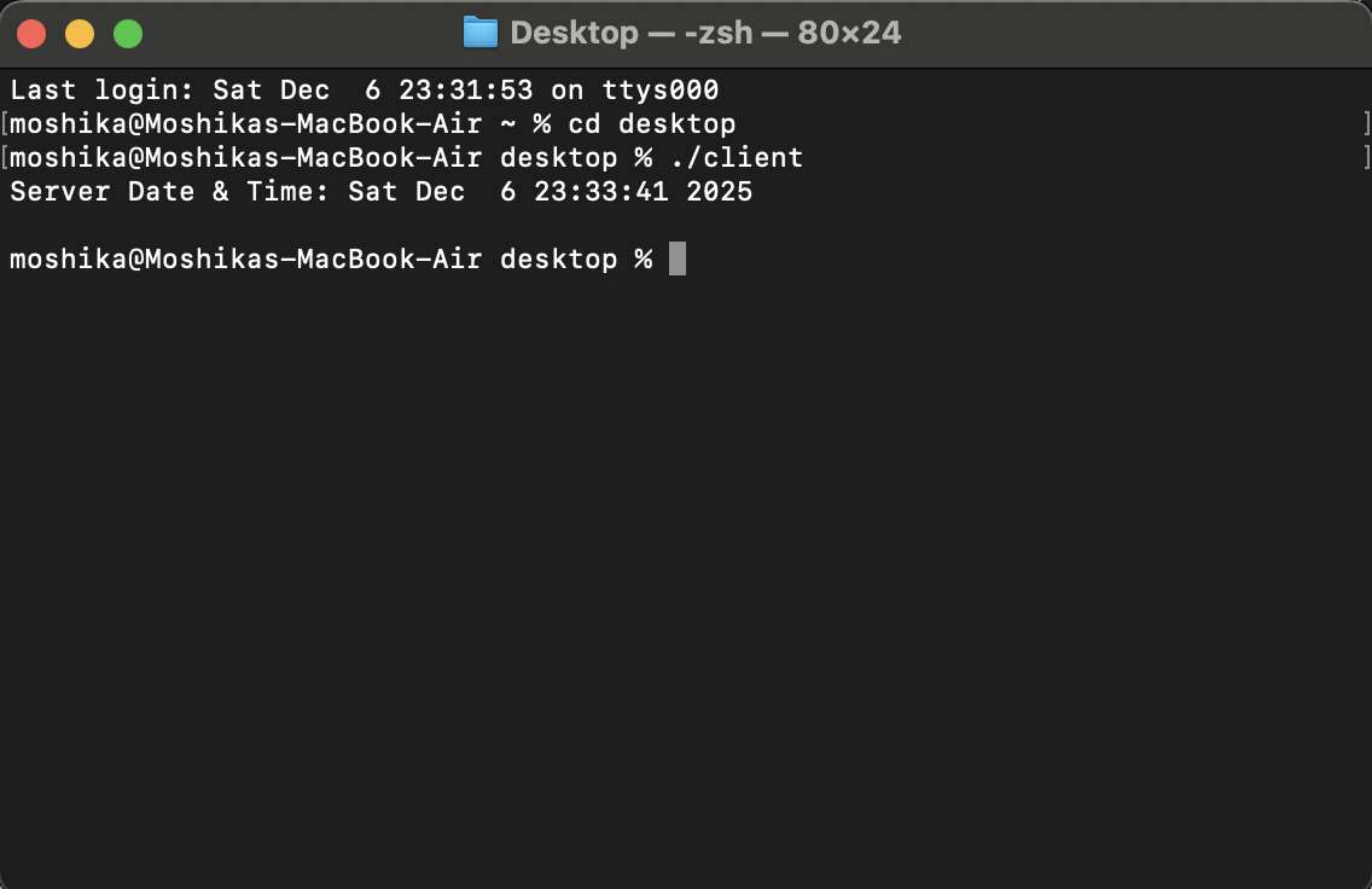
```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <time.h>
#include <arpa/inet.h>

#define PORT 8080

int main() {
    int server_fd, new_socket;
    struct sockaddr_in address;
    int addrlen = sizeof(address);
    char buffer[1024] = {0};
    time_t now;
    char *dt;

    server_fd = socket(AF_INET, SOCK_STREAM, 0);
    if (server_fd < 0) { perror("Socket failed"); exit(1); }

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

    bind(server_fd, (struct sockaddr*)&address, sizeof(address));
    listen(server_fd, 3);

    printf("Server running... waiting for client.\n");

    new_socket = accept(server_fd, (struct sockaddr*)&address, (socklen_t*)&addrlen);
    read(new_socket, buffer, 1024);

    if (strcmp(buffer, "TIME") == 0) {
        time(&now);
        dt = ctime(&now);
        send(new_socket, dt, strlen(dt), 0);
    }

    close(new_socket);
    close(server_fd);
    return 0;
}
```



```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <arpa/inet.h>

#define PORT 8080

int main() {
    int sock = 0;
    struct sockaddr_in serv_addr;
    char buffer[1024] = {0};

    sock = socket(AF_INET, SOCK_STREAM, 0);
    if (sock < 0) {
        printf("Socket creation error\n");
        return 1;
    }

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

    if (inet_pton(AF_INET, "127.0.0.1", &serv_addr.sin_addr) <= 0) {
        printf("Invalid address\n");
        return 1;
    }

    if (connect(sock, (struct sockaddr *)&serv_addr, sizeof(serv_addr)) < 0) {
        printf("Connection failed\n");
        return 1;
    }

    send(sock, "TIME", strlen("TIME"), 0);

    read(sock, buffer, 1024);
    printf("Server Date & Time: %s\n", buffer);

    close(sock);

    return 0;
}
```



```
Last login: Sat Dec  6 23:30:43 on ttys009
[moshika@Moshikas-MacBook-Air ~ % cd desktop
moshika@Moshikas-MacBook-Air desktop % gcc server.c -o server

moshika@Moshikas-MacBook-Air desktop % gcc client.c -o client

moshika@Moshikas-MacBook-Air desktop % ./server

Server running... waiting for client.
moshika@Moshikas-MacBook-Air desktop % 
```



```
Last login: Sat Dec  6 23:31:53 on ttys000
moshika@Moshikas-MacBook-Air ~ % cd desktop
moshika@Moshikas-MacBook-Air desktop % ./client
Server Date & Time: Sat Dec  6 23:33:41 2025

moshika@Moshikas-MacBook-Air desktop % 
```