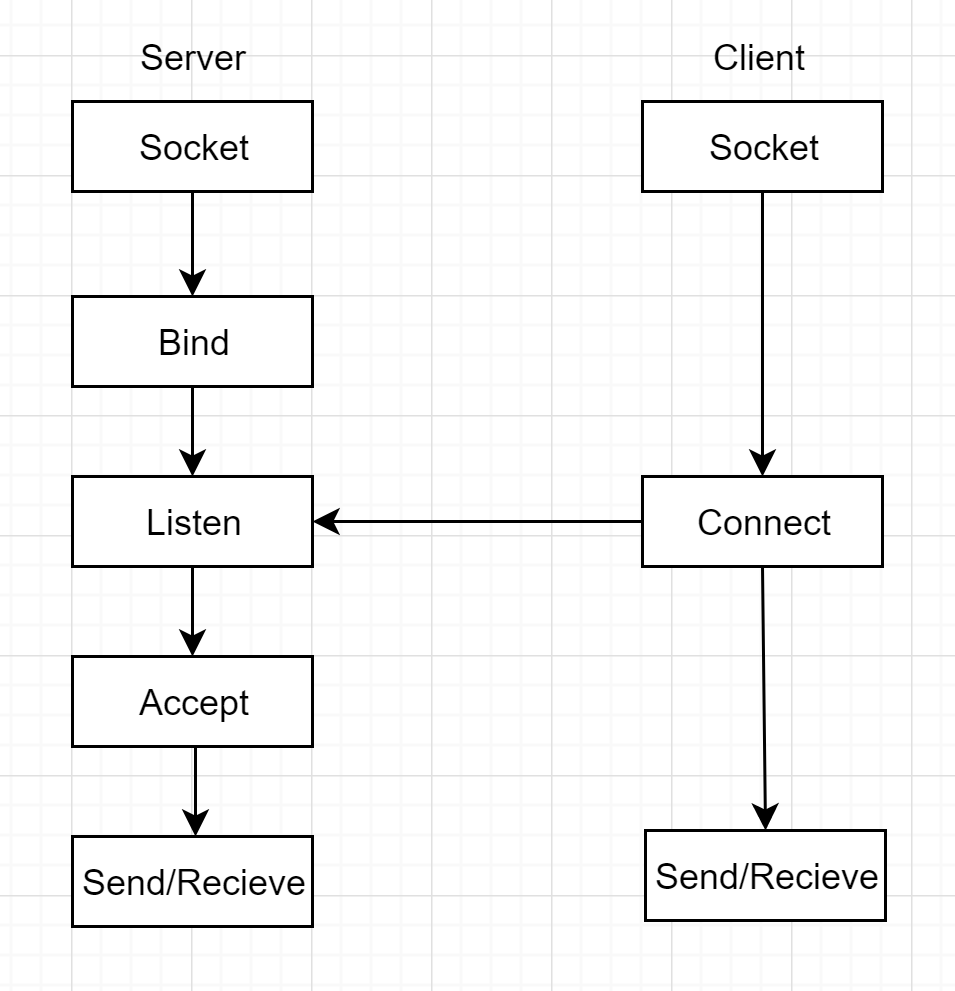
***Project Report***

***Authors***

***Shanmukha Praveen Madasu (801076693)***

***Narendra Kumar Vankayala (801081957)***



**Steps for Creating server:**

1) socket creation:

***int sockfd = socket (domain, type, protocol);***

Sockfd: socket file descriptor

Domain: AF\_INET (IPv4), AF\_INET6 (IPv6)

Type: SOCK\_STREAM (TCP), SOCK\_DGRAM (UDP)

Protocol: Internet protocol, value is 0

2) Bind a name to a socket:

***int bind (int sockfd, const struct sockaddr \*addr, socklen\_t addrlen);***

when a socket is created it exists in the namespace but has no address assigned to it. Bind() assigns the address specified by addr to the socket referred by the file descriptor.

3) Listen for connections on the socket:

***int listen(int sockfd, int backlog);***

This marks the socket referred by the sockfd as a passive socket. That is, as a socket that will be used to accept incoming connections.

4) Accept a connection on the socket:

***int accept(int sockfd, struct sockaddr \*addr, socklen\_t \*addrlen);***

It extracts the 1st connection request on the queue of pending connections for the listening socket.

**Steps for Creating client:**

1) socket creation:

***int sockfd = socket (domain, type, protocol);***

Sockfd: socket file descriptor

Domain: AF\_INET (IPv4), AF\_INET6 (IPv6)

Type: SOCK\_STREAM (TCP), SOCK\_DGRAM (UDP)

Protocol: Internet protocol, value is 0

2) Initiate a connection on a socket:

***int connect(int sockfd, const struct sockaddr \*addr, socklen\_t addrlen);***

This connects the socket referred by the file descriptor to the address specified by addr. Server’s address and port is specified in the addr.

**Compilation & Execution:**

***Client:***

gcc http\_client.c -w -o client

./client localhost 7000 GET data.txt OR

./client localhost 7000 PUT sample.txt

***Server:***

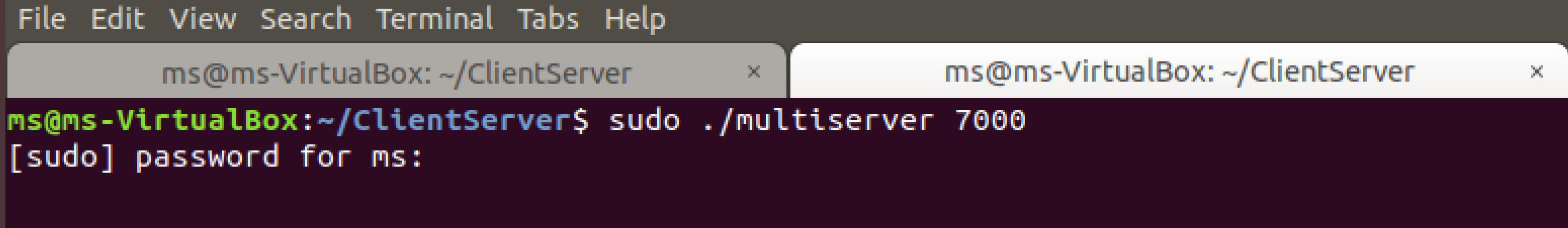
gcc multithreaded\_http\_server.c -w -o multiserver

./multiserver 7000

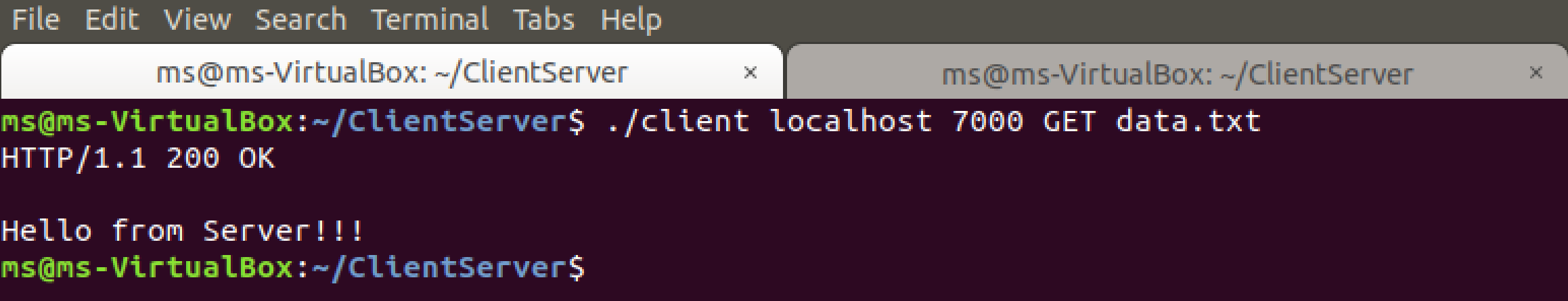
**Results:**

**GET command execution**

Server:

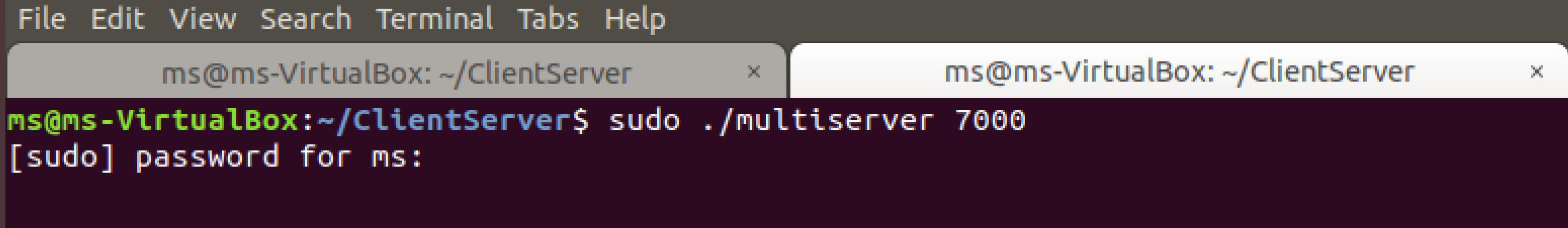


Client:

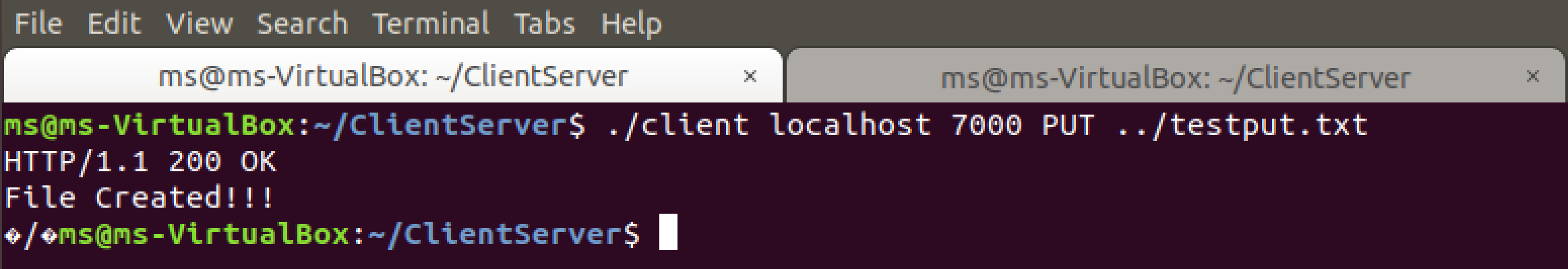


**PUT command execution**

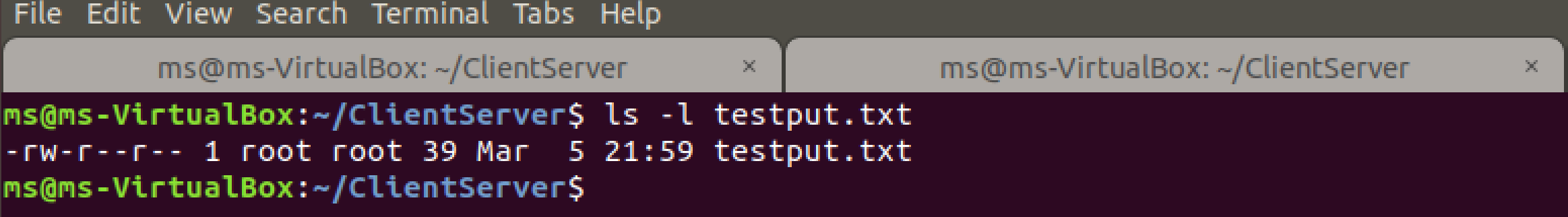
Server:



Client:

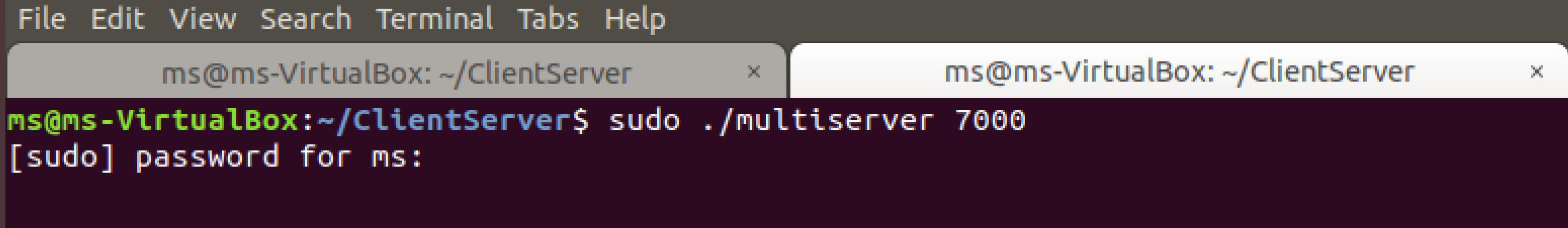


File in the server side

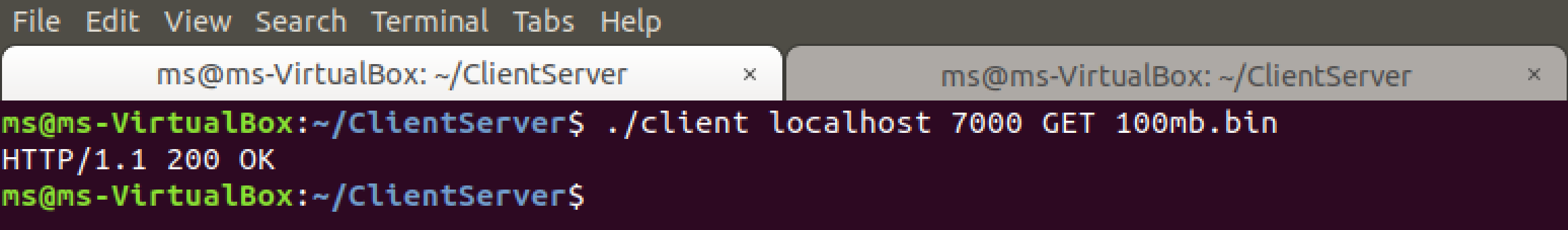


**Multi-Threaded server Test case:**

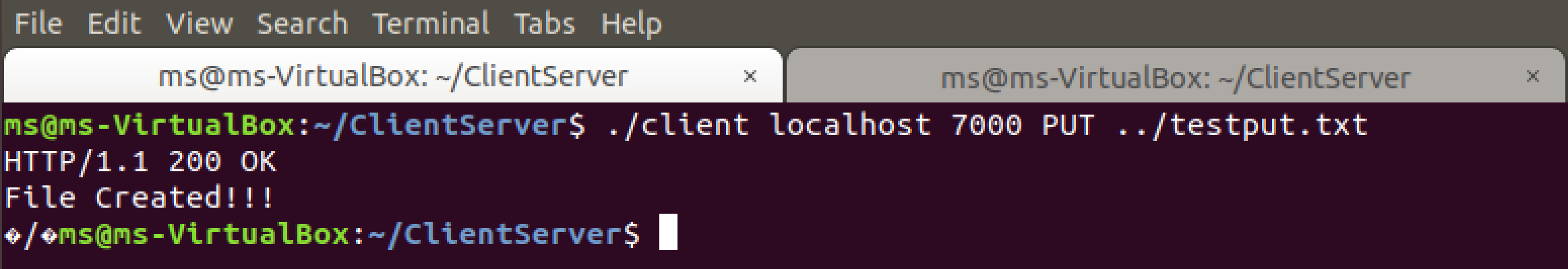
Server:



Client 1:

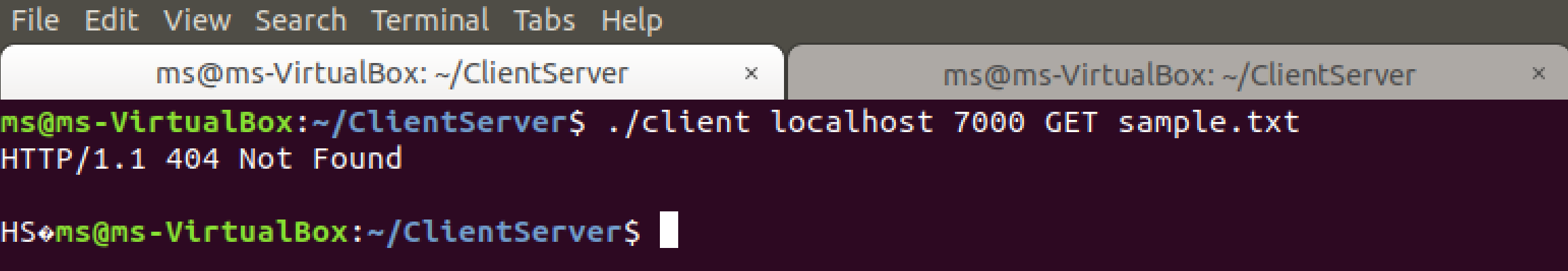


Client 2:

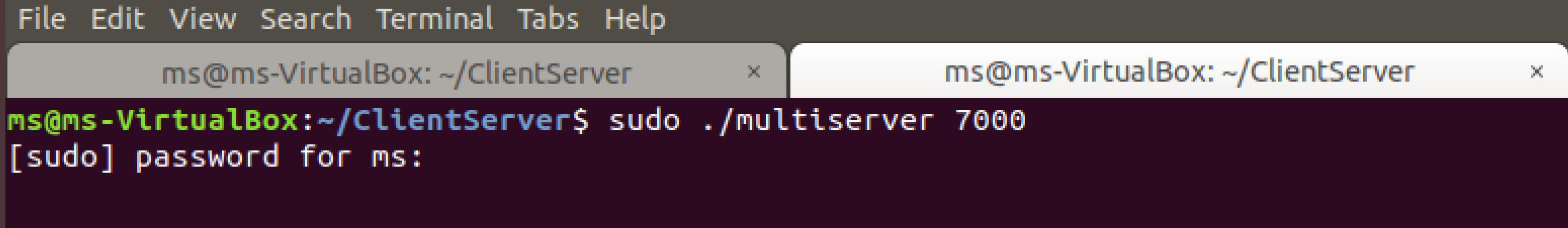


**Non-Existing file Test Case:**

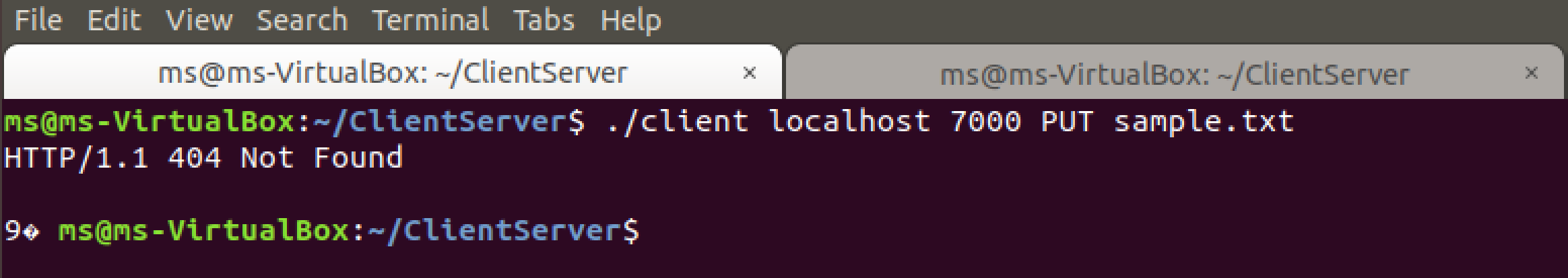
**Client GET:**



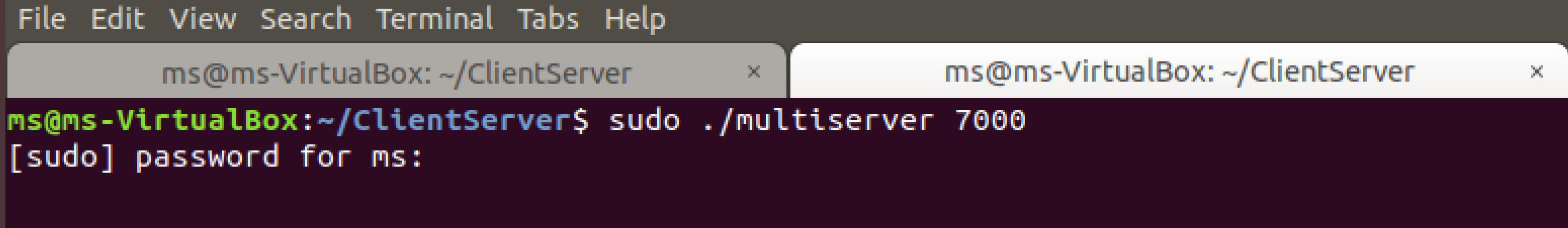
Server:



**Client PUT:**



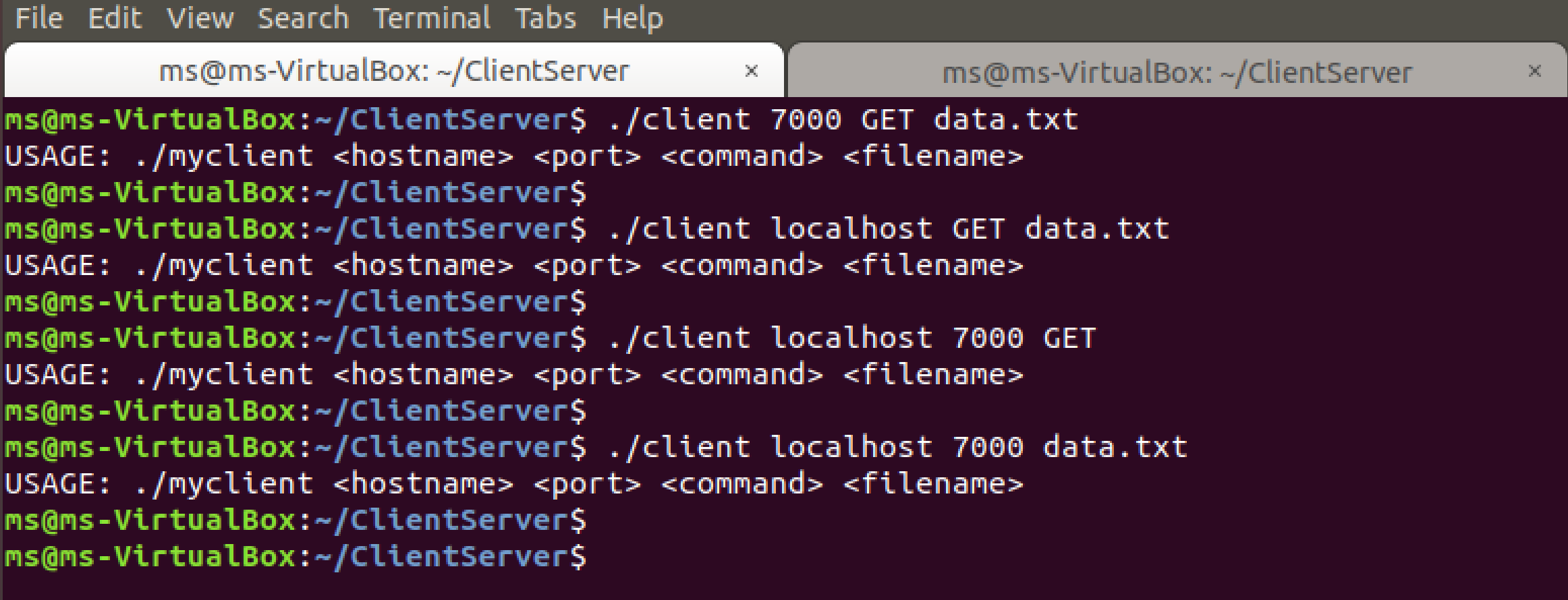
Server:



**Error Handling Test Case:**

Client:

When any of the argument is missing {localhost, port no, command, filename}, client will throw error.



Server:

When port number is missing, server will throw error and will not run.

