



# TCP Client-Server Program to Check if a Given String is Palindrome



thakur\_aman

Read

Discuss

Courses

Practice

Video

## Prerequisites:

- [Socket Programming in C/C++](#),
- [TCP and UDP server using select](#),
- [UDP Server-Client implementation in C](#)
- [TCP Client-Server Implementation in C](#)

This article describes a Client and Server setup where a Client connects, sends a string to the server and the server shows the original string and sends confirmation whether the string is a palindrome or not, to the client using socket connection.

## Examples:

**Input:** naman

**Output:** Palindrome

**Input:** geek

**Output:** Not Palindrome

Recommended: Please try your approach on ***{IDE}*** first, before moving on to the solution.

## Approach:



- In this, first set up a client-server connection.
- When the connection will setup, the client will send the user input string to the server by the send system call.
- At the server-side, the server will wait for a string sent by the client.
- Server reads the string by the reading system call.
- After this, the server will check if the string is a palindrome or not and sends the confirmation back to the client.

### Compiling:

1. First, run the server program as

```
gcc server.c -o server
./server
```

2. Run the client program on another terminal

```
gcc client.c -o client
./client
```

3. Server program is waiting for the string sent by the client.
4. Input the string in client-side.
5. Server program will print original string.
6. Client program will print result.

Below is the implementation of the above approach:

---

## TCP Server

```
// defines in_addr structure
#include <arpa/inet.h>

// contains constants and structures
// needed for internet domain addresses
```

```
#include <netinet/in.h>

// standard input and output library
#include <stdio.h>

// contains string functions
#include <string.h>

// for socket creation
#include <sys/socket.h>

// contains constructs that facilitate getting
// information about files attributes.
#include <sys/stat.h>

// contains a number of basic derived types
// that should be used whenever appropriate
#include <sys/types.h>

main()
{
    struct sockaddr_in client, server;
    int s, n, sock, g, j, left, right, flag;
    char b1[20], b2[10], b3[10], b4[10];

    // creating socket
    s = socket(AF_INET, SOCK_STREAM, 0);

    // assign IP, PORT
    server.sin_family = AF_INET;

    // this is the port number of running server
    server.sin_port = 2000;
    server.sin_addr.s_addr = inet_addr("127.0.0.1");

    // Binding newly created socket
    // to given IP and verification
    bind(s, (struct sockaddr*)&server, sizeof server);
    listen(s, 1);
    n = sizeof client;

    sock = accept(s, (struct sockaddr*)&client, &n);
    for (;;) {
        recv(sock, b1, sizeof(b1), 0);

        // whenever a request from a client came.
        // It will be processed here.
        printf("\nThe string received is:%s\n", b1);
        if (strlen(b1) == 0)
            flag = 1;
        else {
            left = 0;
            right = strlen(b1) - 1;
            flag = 1;
            while (left < right && flag) {
                if (b1[left] != b1[right])
                    flag = 0;
            }
        }
    }
}
```

```

        else {
            left++;
            right--;
        }
    }
}
send(sock, &flag, sizeof(int), 0);
break;
}
close(sock);

// close the socket
close(s);
}

```

[Job Fair For Students](#)
[Trending Now](#)
[DSA](#)
[Data Structures](#)
[Algorithms](#)
[Interview Preparation](#)
[Data Sci](#)

```

// defines in_addr structure
#include <arpa/inet.h>

// contains constants and structures
// needed for internet domain addresses
#include <netinet/in.h>

// standard input and output library
#include <stdio.h>

// contains string functions
#include <string.h>

// for socket creation
#include <sys/socket.h>

// contains constructs that facilitate getting
// information about files attributes.
#include <sys/stat.h>

// contains a number of basic derived types
// that should be used whenever appropriate
#include <sys/types.h>

main()
{
    struct sockaddr_in client;
    int s, flag;
    char buffer[20];

    // socket create
    s = socket(AF_INET, SOCK_STREAM, 0);

    // assign IP, PORT
    client.sin_family = AF_INET;
    client.sin_port = 2000;
    client.sin_addr.s_addr = inet_addr("127.0.0.1");
}

```

```
// connect the client socket to server socket
connect(s, (struct sockaddr*)&client, sizeof client);

for (;;) {
    printf("\nEnter a string to check palindrome: ");
    scanf("%s", buffer);

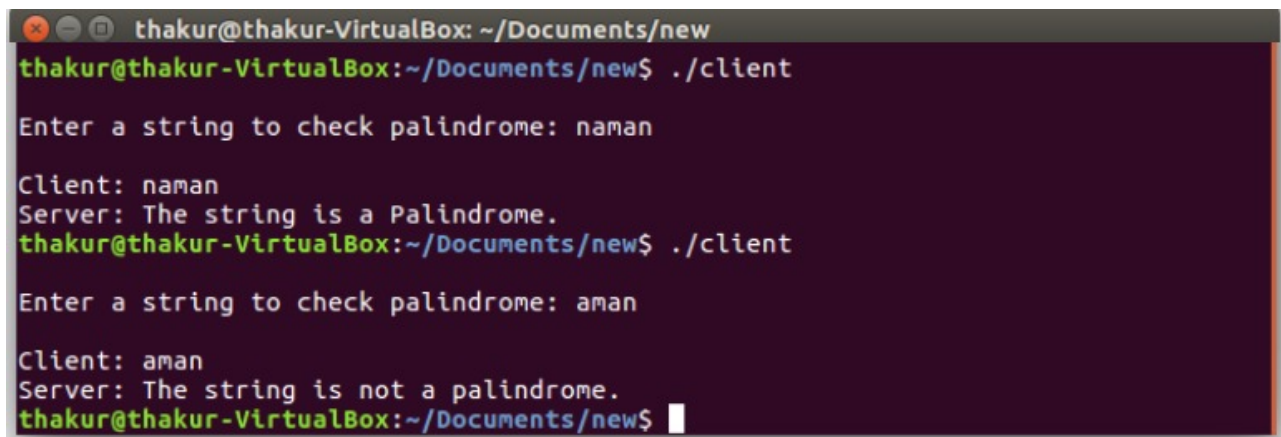
    printf("\nClient: %s", buffer);
    send(s, buffer, sizeof(buffer), 0);
    recv(s, &flag, sizeof(int), 0);

    if (flag == 1) {
        printf("\nServer: The string is a Palindrome.\n");
        break;
    }
    else {
        printf("\nServer: The string is not a palindrome.\n");
        break;
    }
}

// close the socket
close(s);
}
```

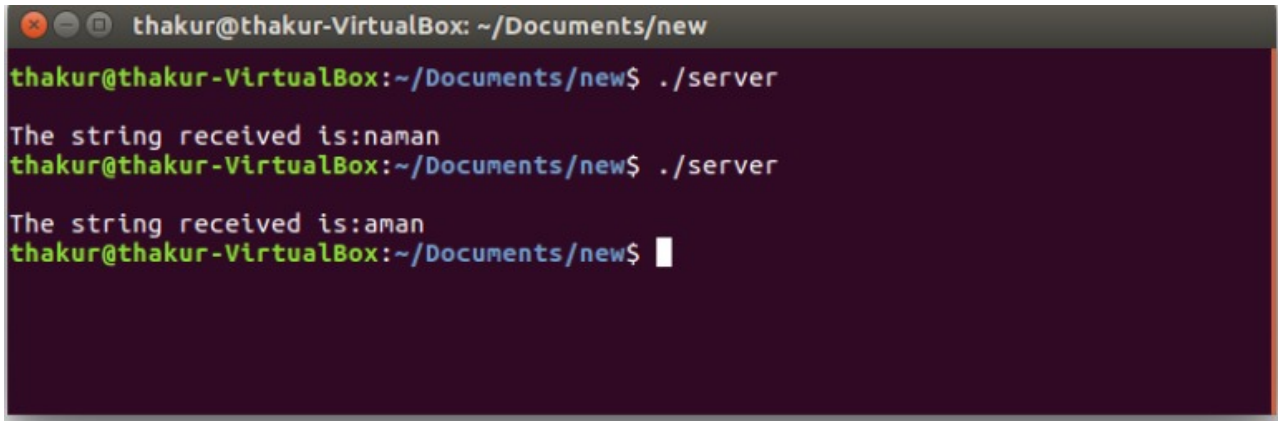
### Output:

- Server Side:



```
thakur@thakur-VirtualBox: ~/Documents/new
thakur@thakur-VirtualBox:~/Documents/new$ ./client
Enter a string to check palindrome: naman
Client: naman
Server: The string is a Palindrome.
thakur@thakur-VirtualBox:~/Documents/new$ ./client
Enter a string to check palindrome: aman
Client: aman
Server: The string is not a palindrome.
thakur@thakur-VirtualBox:~/Documents/new$
```

- Client Side:



```
thakur@thakur-VirtualBox: ~/Documents/new
thakur@thakur-VirtualBox:~/Documents/new$ ./server
The string received is:naman
thakur@thakur-VirtualBox:~/Documents/new$ ./server
The string received is:aman
thakur@thakur-VirtualBox:~/Documents/new$
```

Last Updated : 02 Aug, 2019

3

## Similar Reads

1. TCP Server-Client implementation in C
2. Setting up local DNS server between client-server machines
3. TCP Tahoe and TCP Reno
4. TCP with explicit link failure notification (TCP-ELFN)
5. UDP Client Server using connect | C implementation
6. Java Implementation of Diffie-Hellman Algorithm between Client and Server
7. Two way communication between Client and Server using Win32 Threads
8. Distributed Objects Computing: The next generation of client-server computing
9. Difference between Server OS and Client OS
10. Difference between Client /Server and Distributed DBMS

[Previous](#)

[Next](#)

## Article Contributed By :



**thakur\_aman**

@thakur\_aman

## Vote for difficulty

Current difficulty : [Hard](#)

Easy

Normal

Medium

Hard

Expert

**Article Tags :** [cpp-multithreading](#), [palindrome](#), [C Programs](#), [Computer Networks](#)

**Practice Tags :** [Computer Networks](#), [palindrome](#), [Palindrome](#)

Improve Article

Report Issue



A-143, 9th Floor, Sovereign Corporate Tower,  
Sector-136, Noida, Uttar Pradesh - 201305

[feedback@geeksforgeeks.org](mailto:feedback@geeksforgeeks.org)

### Company

[About Us](#)

[Careers](#)

[In Media](#)

[Contact Us](#)

[Terms and Conditions](#)

[Privacy Policy](#)

[Copyright Policy](#)

[Third-Party Copyright Notices](#)

[Advertise with us](#)

### Languages

[Python](#)

### Trending @GfG

[Job Fair For Students](#)

[GfG Weekly #100](#)

[POTD: Revamped](#)

[Python Backend LIVE](#)

[Android App Development](#)

[DevOps LIVE](#)

[DSA in JavaScript](#)

### Data Structures

[Array](#)

[Java](#)[String](#)[C++](#)[Linked List](#)[GoLang](#)[Stack](#)[SQL](#)[Queue](#)[R Language](#)[Tree](#)[Android Tutorial](#)[Graph](#)

## Algorithms

[Sorting](#)[HTML](#)[Searching](#)[CSS](#)[Greedy](#)[JavaScript](#)[Dynamic Programming](#)[Bootstrap](#)[Pattern Searching](#)[ReactJS](#)[Recursion](#)[AngularJS](#)[Backtracking](#)[NodeJS](#)

## Web Development

## Data Science & ML

## Interview Corner

[Data Science With Python](#)[Company Preparation](#)[Data Science For Beginner](#)[Preparation for SDE](#)[Machine Learning Tutorial](#)[Company Interview Corner](#)[Maths For Machine Learning](#)[Experienced Interview](#)[Pandas Tutorial](#)[Internship Interview](#)[NumPy Tutorial](#)[Competitive Programming](#)[NLP Tutorial](#)[Aptitude](#)

## Python

## GfG School

[Python Tutorial](#)[CBSE Notes for Class 8](#)[Python Programming Examples](#)[CBSE Notes for Class 9](#)[Django Tutorial](#)[CBSE Notes for Class 10](#)[Python Projects](#)[CBSE Notes for Class 11](#)[Python Tkinter](#)[CBSE Notes for Class 12](#)[OpenCV Python Tutorial](#)[English Grammar](#)

## UPSC/SSC/BANKING

## Write & Earn

[SSC CGL Syllabus](#)[Write an Article](#)[SBI PO Syllabus](#)[Improve an Article](#)[IBPS PO Syllabus](#)[Pick Topics to Write](#)



[UPSC Ethics Notes](#)

[Write Interview Experience](#)

[UPSC Economics Notes](#)

[Internships](#)

[UPSC History Notes](#)

[Video Internship](#)

@geeksforgeeks , Some rights reserved