



Computer Networks Laboratory

Lab Assignment-3 (L3)

22.08.2019

Rahul Dugar

Enr. no. 17114061

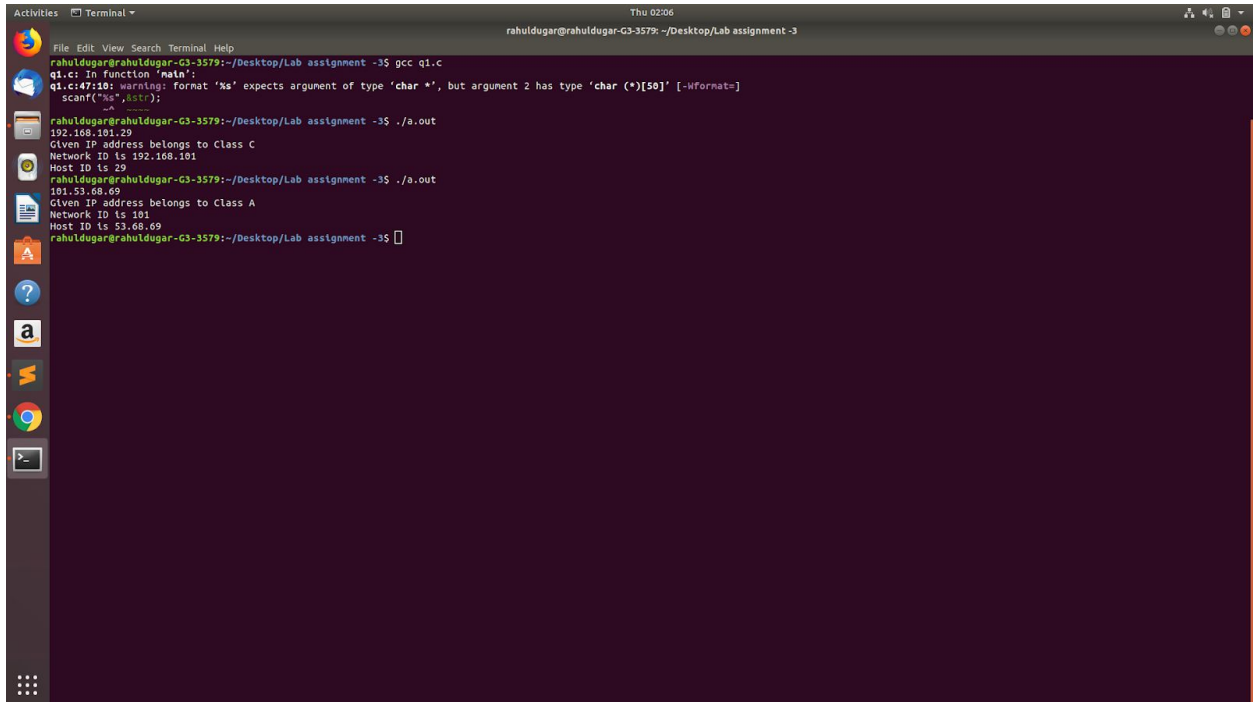
CS-2

Problem Statement 1:

Write a socket program in C to determine class, Network and Host ID of an IPv4 address.

Algorithms and Data Structures Used:

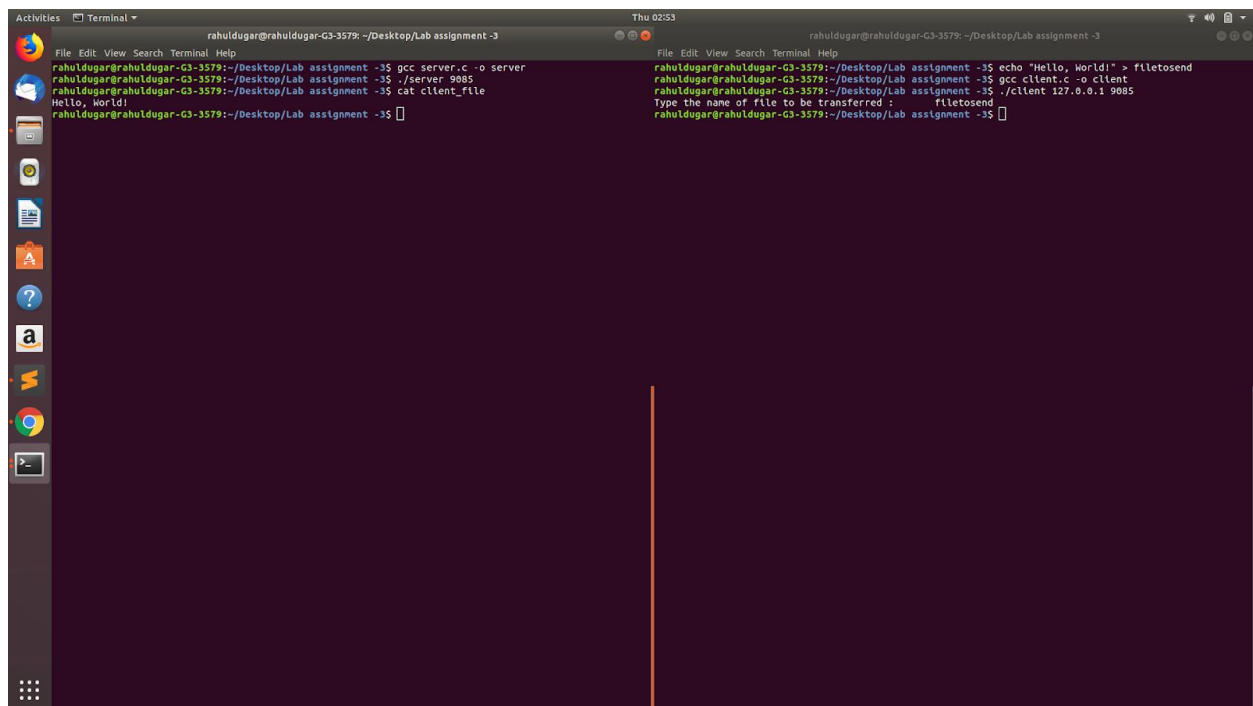
Direct conditional statements were used.



```
Activities Terminal
rahuldugar@rahuldugar-G3-3579: ~/Desktop/Lab assignment -3
File Edit View Search Terminal Help
rahuldugar@rahuldugar-G3-3579:~/Desktop/Lab assignment -3$ gcc q1.c
q1.c: In function 'main':
q1.c:47:10: warning: format '%s' expects argument of type 'char *', but argument 2 has type 'char (*)[50]' [-Wformat=]
scanf("%s", &str);
          ^
rahuldugar@rahuldugar-G3-3579:~/Desktop/Lab assignment -3$ ./a.out
192.168.101.29
Given IP address belongs to Class C
Network ID is 192.168.101
Host ID is 29
rahuldugar@rahuldugar-G3-3579:~/Desktop/Lab assignment -3$ ./a.out
101.53.68.69
Given IP address belongs to Class A
Network ID is 101
Host ID is 53.68.69
rahuldugar@rahuldugar-G3-3579:~/Desktop/Lab assignment -3$
```

Problem Statement 2:

Write a C program to demonstrate File Transfer using UDP.



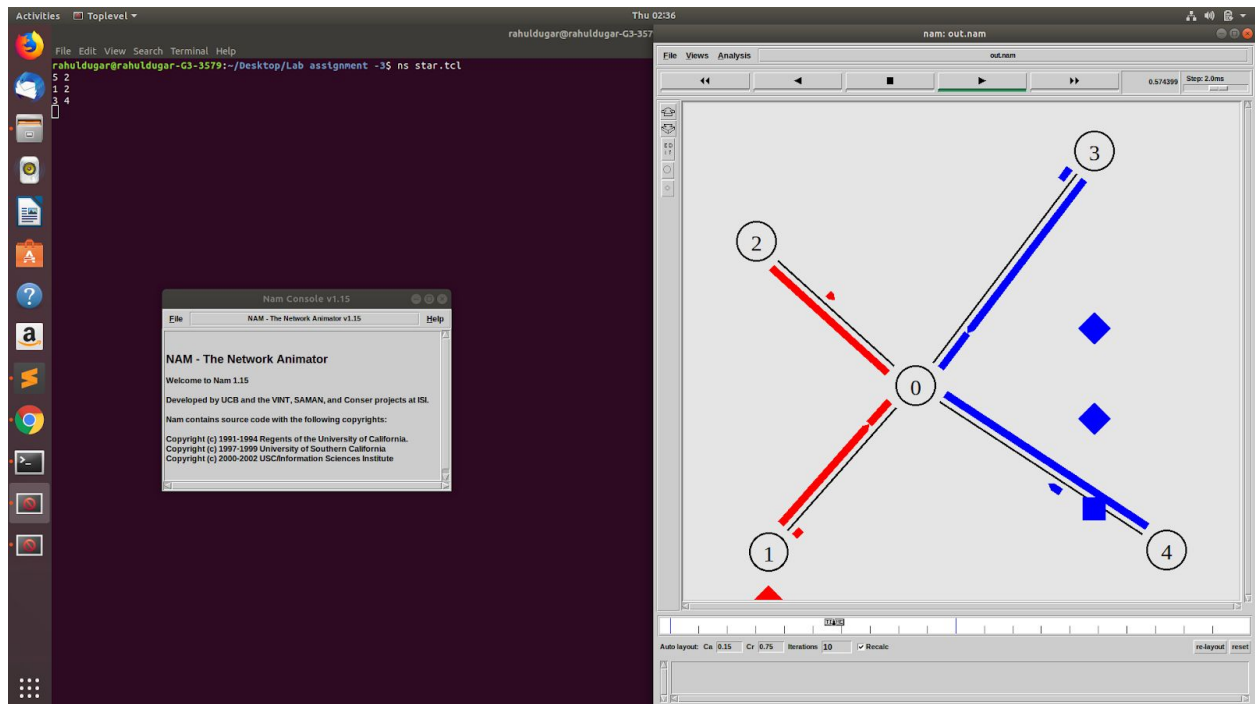
The image shows two terminal windows side-by-side. The left window shows the compilation of a server program. The right window shows the compilation of a client program and its execution, which sends a file named 'filetosend' to the server.

```
rahuldugar@rahuldugar-G3-3579: ~/Desktop/Lab assignment-3
File Edit View Search Terminal Help
rahuldugar@rahuldugar-G3-3579:~/Desktop/Lab assignment-3$ gcc server.c -o server
rahuldugar@rahuldugar-G3-3579:~/Desktop/Lab assignment-3$ ./server 9085
rahuldugar@rahuldugar-G3-3579:~/Desktop/Lab assignment-3$ cat client_file
Hello, World!
rahuldugar@rahuldugar-G3-3579:~/Desktop/Lab assignment-3$
```

```
rahuldugar@rahuldugar-G3-3579: ~/Desktop/Lab assignment-3
File Edit View Search Terminal Help
rahuldugar@rahuldugar-G3-3579:~/Desktop/Lab assignment-3$ echo "Hello, World!" > filetosend
rahuldugar@rahuldugar-G3-3579:~/Desktop/Lab assignment-3$ gcc client.c -o client
rahuldugar@rahuldugar-G3-3579:~/Desktop/Lab assignment-3$ ./client 127.0.0.1 9085
Type the name of file to be transferred : filetosend
rahuldugar@rahuldugar-G3-3579:~/Desktop/Lab assignment-3$
```

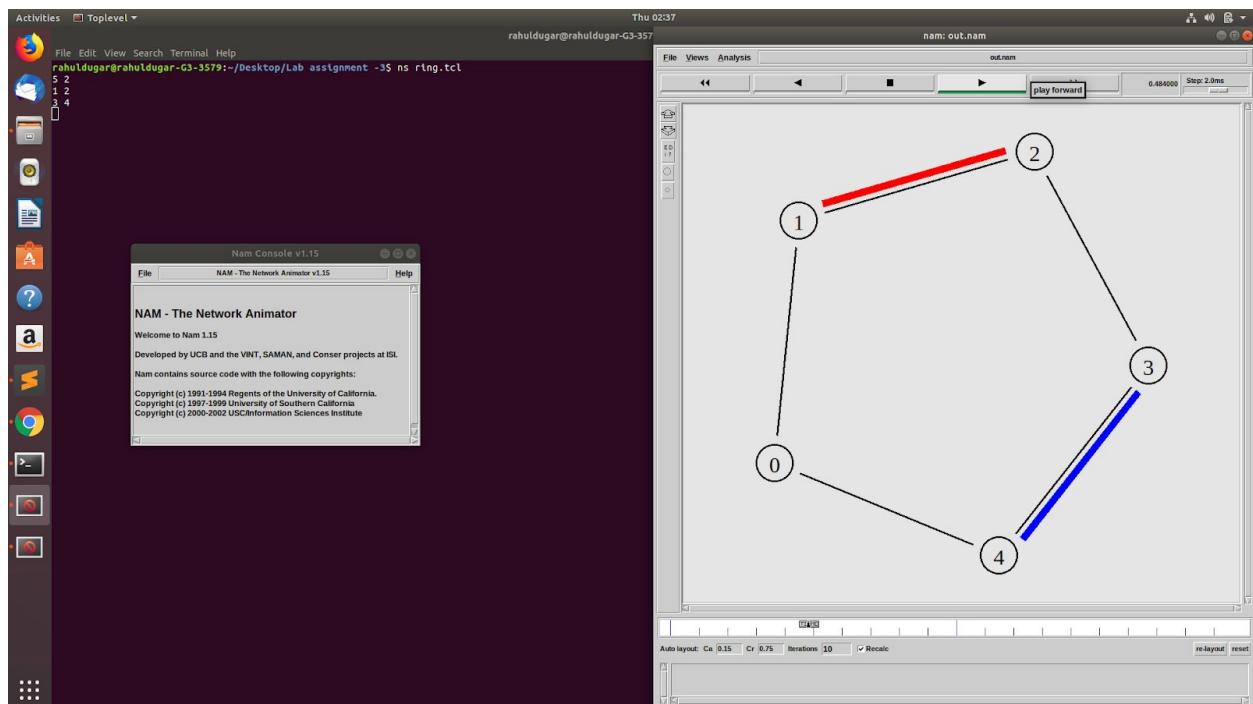
Problem Statement 3:

Write a TCL code for network simulator NS2 to demonstrate the star topology among a set of computer nodes. Given N nodes, one node will be assigned as the central node and the other nodes will be connected to it to form the star. You have to set up a TCP connection between k pairs of nodes and demonstrate the packet transfer between them using Network Animator (NAM). Use File Transfer protocol (FTP) for the same. Each link should have different color of packets to differentiate the packets transferred between each pair of nodes. The program should take the number of nodes (N) as input followed by k pairs of nodes.



Problem Statement 4:

Write a TCL code for network simulator NS2 to demonstrate the ring topology among a set of computer nodes. Given N nodes, each node will be connected to two other nodes in the form of a ring. You have to set up a TCP connection between k pairs of nodes and demonstrate packet transfer between them using Network Animator (NAM). Use File Transfer protocol (FTP) for the same. Each link should have different color of packets to differentiate the packets transferred between each pair of nodes. The program should take the number of nodes (N) as input followed by k pairs of nodes.



Problem Statement 5:

Write a TCL code for network simulator NS2 to demonstrate the bus topology among a set of computer nodes. Given N nodes, each node will be connected to a common link. You have to set up a TCP connection between k pairs of nodes and demonstrate packet transfer between them using Network Animator (NAM). Use File Transfer protocol (FTP) for the same. Each link should have different color of packets to differentiate the packets transferred between each pair of nodes. The program should take the number of nodes (N) as input followed by k pairs of nodes.

