

Computer Network Laboratory

CSN-361

Assignment 2

Name: Anshuman Shakya
Enrollment Number: 17114013
Class: 3rd year, B.Tech CSE

Problem Statement-

Question 1 : Write a socket program in C to connect two nodes on a network to communicate with each other, where one socket listens on a particular port at an IP, while other socket reaches out to the other to form a connection.

Sol->

Algorithm used:- Socket Programming

One socket(node) listens on a particular port at an IP, while other socket reaches out to the other to form a connection. Server forms the listener socket while client reaches out to the server.

1. Written a C program to create the server using TCP protocol and IPv4 address.
2. Written a C program to create the client using TCP protocol and IPv4 address and connects to the localhost.

Data structures used :

Server-

Socket creation:

sockfd: socket descriptor, an integer

struct sockaddr_in : structure to store internet addresses like IP address, port.

Bind:

```
int bind(int sockfd, const struct sockaddr *addr, socklen_t addrlen);
```

bind function binds the socket to the address and port number specified in addr.

Listen:

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

It puts the server socket in a passive mode, where it waits for the client to approach the server to make a connection. The backlog, defines the maximum length to which the queue of pending connections for sockfd may grow.

Accept:

```
int new_socket= accept(int sockfd, struct sockaddr *addr, socklen_t *addrlen);
```

It extracts the first connection request on the queue of pending connections for the listening socket, sockfd, creates a new connected socket, and returns a new file descriptor referring to that socket.

Client

Socket connection:

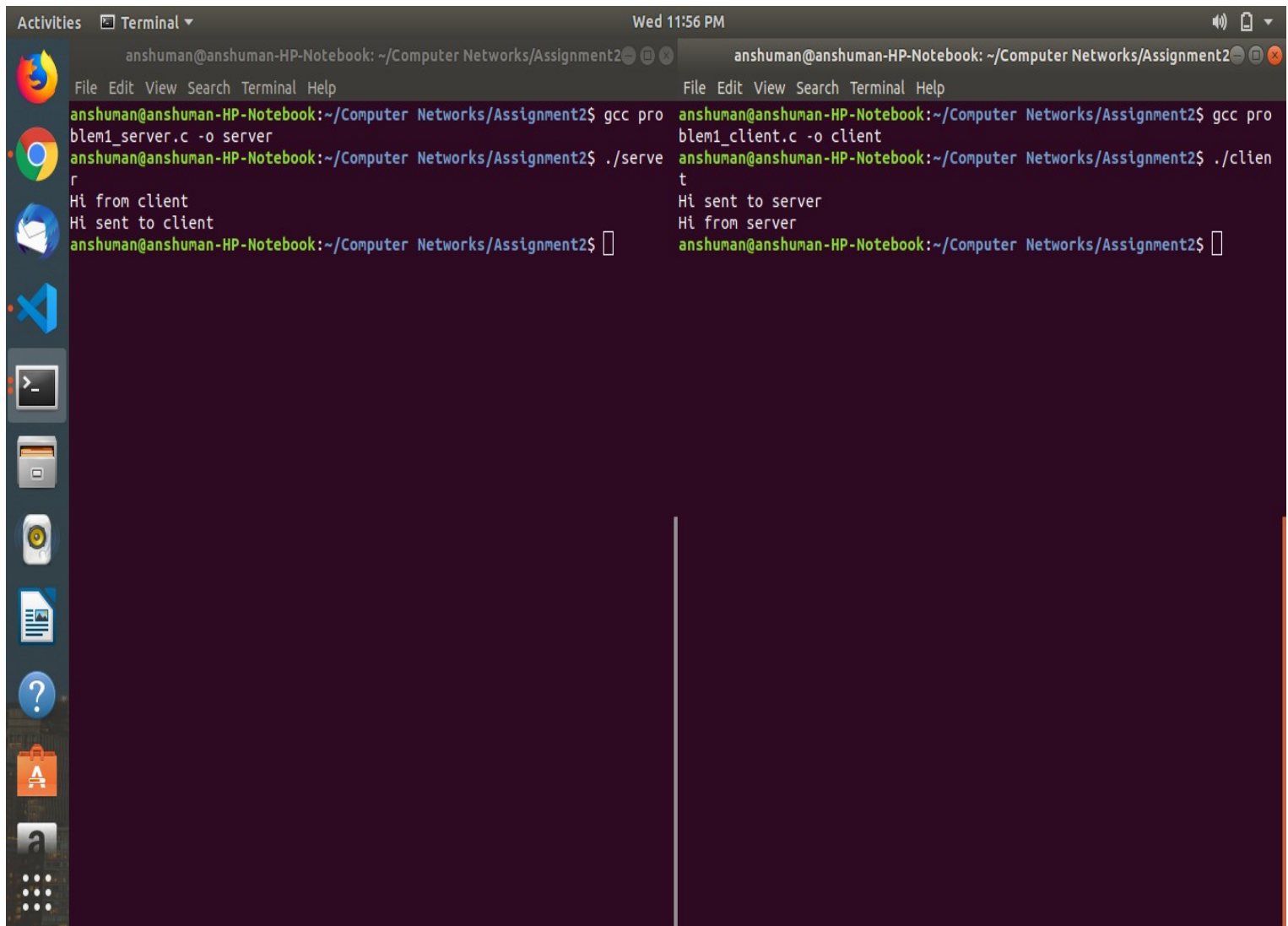
same as that of server's socket creation

Connect:

```
int connect(int sockfd, const struct sockaddr *addr, socklen_t addrlen);
```

The connect() system call connects the socket referred to by the file descriptor sockfd to the address specified by addr. Server's address and port is specified in addr.

Screenshots-



The screenshot displays two terminal windows side-by-side on a Linux desktop. The desktop environment includes a sidebar with icons for Firefox, Google Chrome, a mail client, and a file manager. The top bar shows the date and time as 'Wed 11:56 PM'. Both terminal windows are titled 'anshuman@anshuman-HP-Notebook: ~/Computer Networks/Assignment2'.

Left Terminal Window:

```
anshuman@anshuman-HP-Notebook:~/Computer Networks/Assignment2$ gcc problem1_server.c -o server
anshuman@anshuman-HP-Notebook:~/Computer Networks/Assignment2$ ./server
Hi from client
Hi sent to client
anshuman@anshuman-HP-Notebook:~/Computer Networks/Assignment2$
```

Right Terminal Window:

```
anshuman@anshuman-HP-Notebook:~/Computer Networks/Assignment2$ gcc problem1_client.c -o client
anshuman@anshuman-HP-Notebook:~/Computer Networks/Assignment2$ ./client
Hi sent to server
Hi from server
anshuman@anshuman-HP-Notebook:~/Computer Networks/Assignment2$
```

Question 2: Write a C program to demonstrate both Zombie and Orphan process.

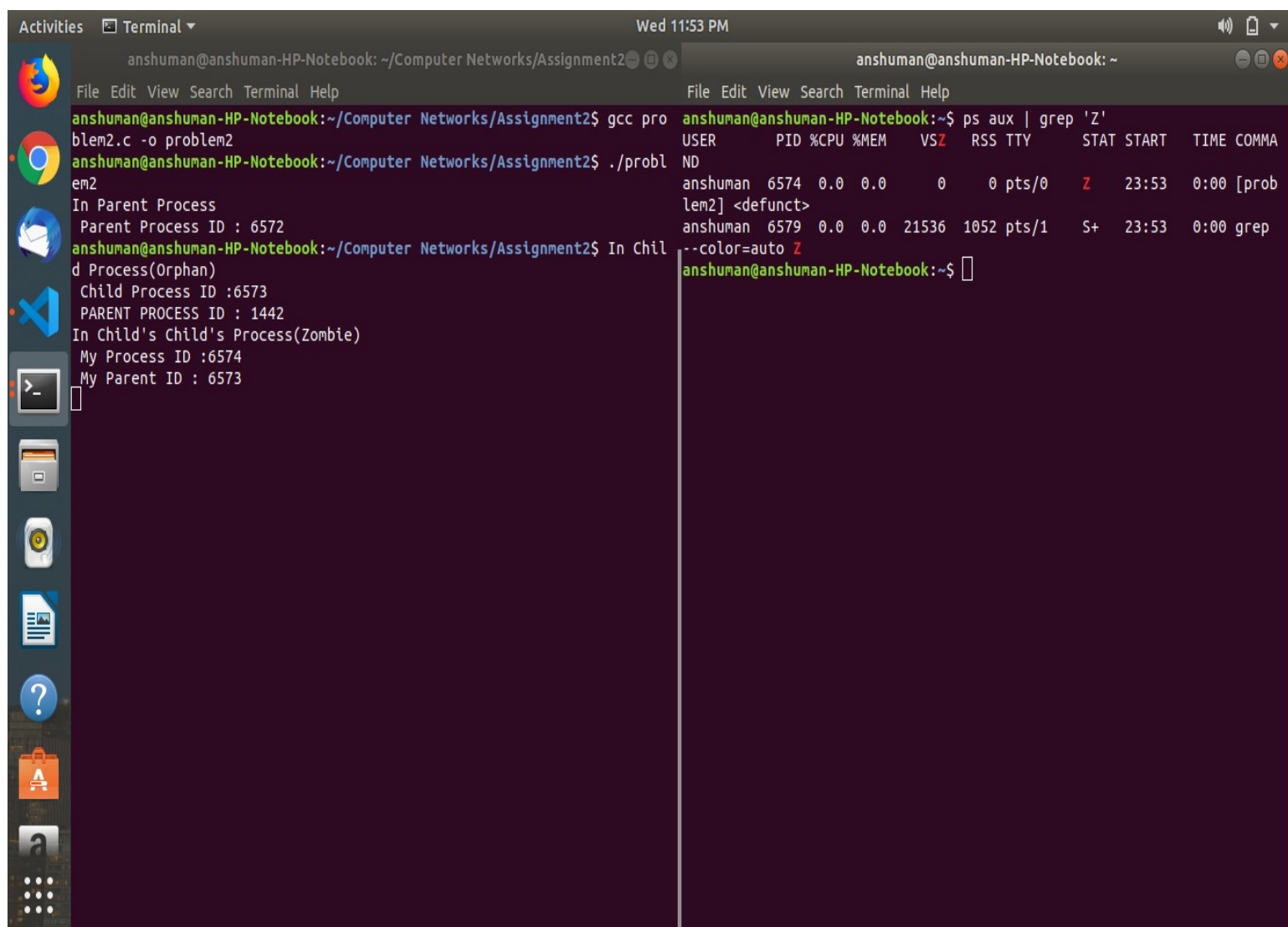
Sol->

Algorithm used:- Created a parent process and a child process and then a child's child process. The parent process exits and child process becomes orphan then the child's child process exits and child process keeps running, which makes child's child process a zombie process. The parent id of orphan process changes to the pid of init process.

Command to check the pid of zombie process : - **ps aux | grep 'Z'**

Data structures used : No data structures used.

Screenshots-



```
anshuman@anshuman-HP-Notebook: ~/Computer Networks/Assignment2$ gcc problem2.c -o problem2
anshuman@anshuman-HP-Notebook: ~/Computer Networks/Assignment2$ ./problem2
In Parent Process
Parent Process ID : 6572
anshuman@anshuman-HP-Notebook: ~/Computer Networks/Assignment2$ In Child Process(Orphan)
Child Process ID :6573
PARENT PROCESS ID : 1442
In Child's Child's Process(Zombie)
My Process ID :6574
My Parent ID : 6573

anshuman@anshuman-HP-Notebook:~$ ps aux | grep 'Z'
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
anshuman  6574  0.0  0.0      0     0 pts/0    Z   23:53   0:00 [problem2] <defunct>
anshuman  6579  0.0  0.0  21536  1052 pts/1    S+  23:53   0:00 grep --color=auto Z
```