Lab Assignment-2

Indian Institute of Technology Roorkee Department of Computer Science and Engineering

CSN-361: Computer Networks Laboratory (Autumn 2019-2020)

Aman jaiswal
Enrollment No:-17114008
B.tech CSE 3rd Year.

Problem Statement 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.

- Data Structure and Functions used:-

For Server -

1. Socket creation:

   int sockfd = socket(domain, type, protocol)

- sockfd: socket descriptor, an integer (like a file-handle)
- domain: integer, communication domain e.g., AF_INET (IPv4 protocol) , AF_INET6 (IPv6 protocol)
- type: communication type
- SOCK_STREAM: TCP(reliable, connection oriented)
- SOCK_DGRAM: UDP(unreliable, connectionless)
- protocol: Protocol value for Internet Protocol(IP), which is 0. This is the same number which appears on protocol field in the IP header of a packet.(man protocols for more details)

2. Setsockopt:

   *int setsockopt(int sockfd, int level, int optname, const void *optval, socklen_t optlen);*

   This helps in manipulating options for the socket referred by the file descriptor sockfd. This is completely optional, but it helps in reuse of address and port. Prevents error such as: “address already in use”.

3. Bind:

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

   After creation of the socket, bind function binds the socket to the address and port number specified in addr(custom data structure). In the example code, we bind the server to the localhost, hence we use INADDR_ANY to specify the IP address.

4. 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. If a connection request arrives when the queue is full, the client may receive an error with an indication of ECONNREFUSED.

5. Accept:

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

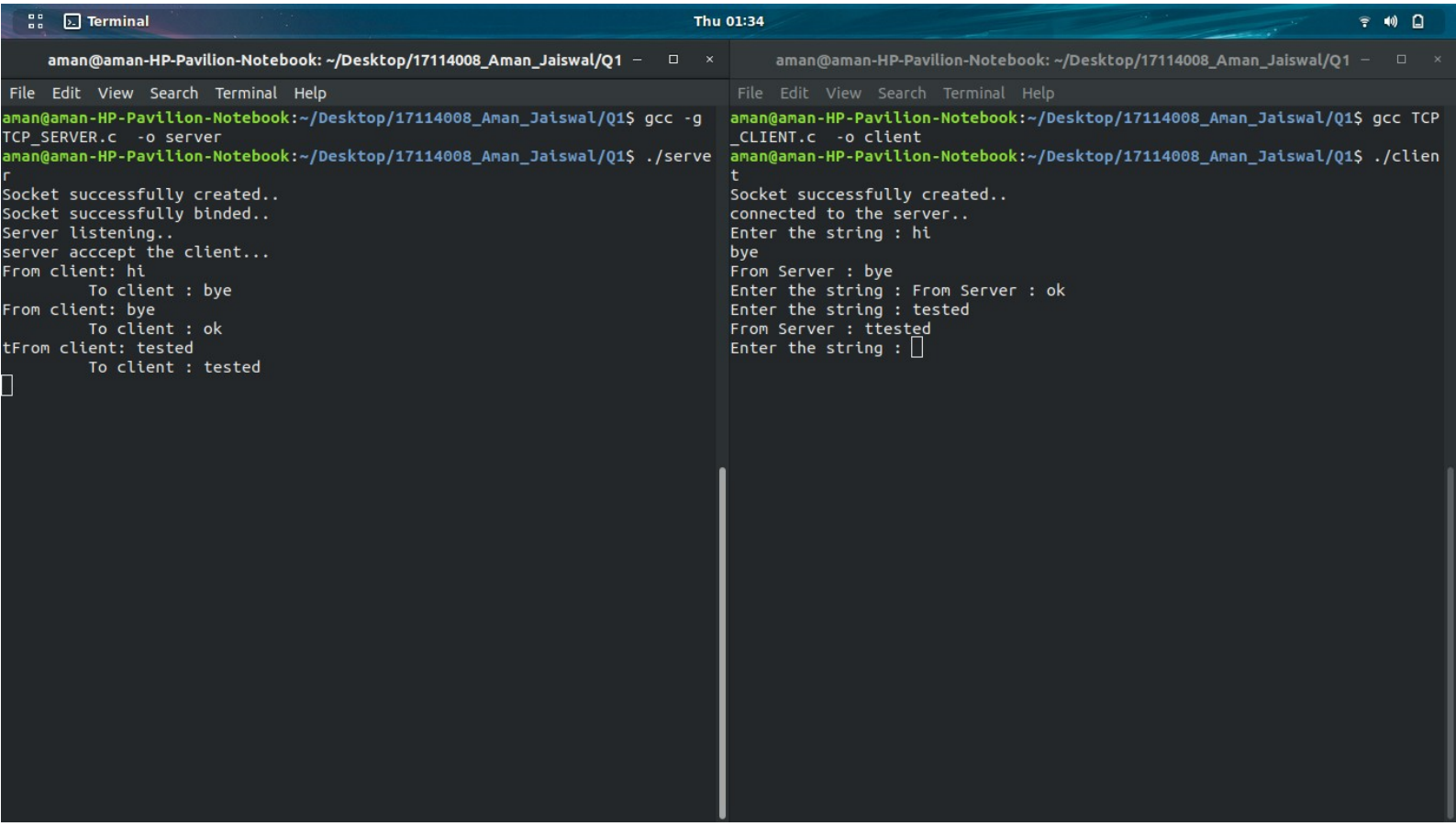
   It extracts the first connection request on the queue of pending connections for the listening socket, sockfd, creates a new connected socket, and returns a new file descriptor referring to that socket. At this point, connection is established between client and server, and they are ready to transfer data.

For Client:-

1. Socket connection: Exactly same as that of server's socket creation
2. 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



Problem Statement 2:

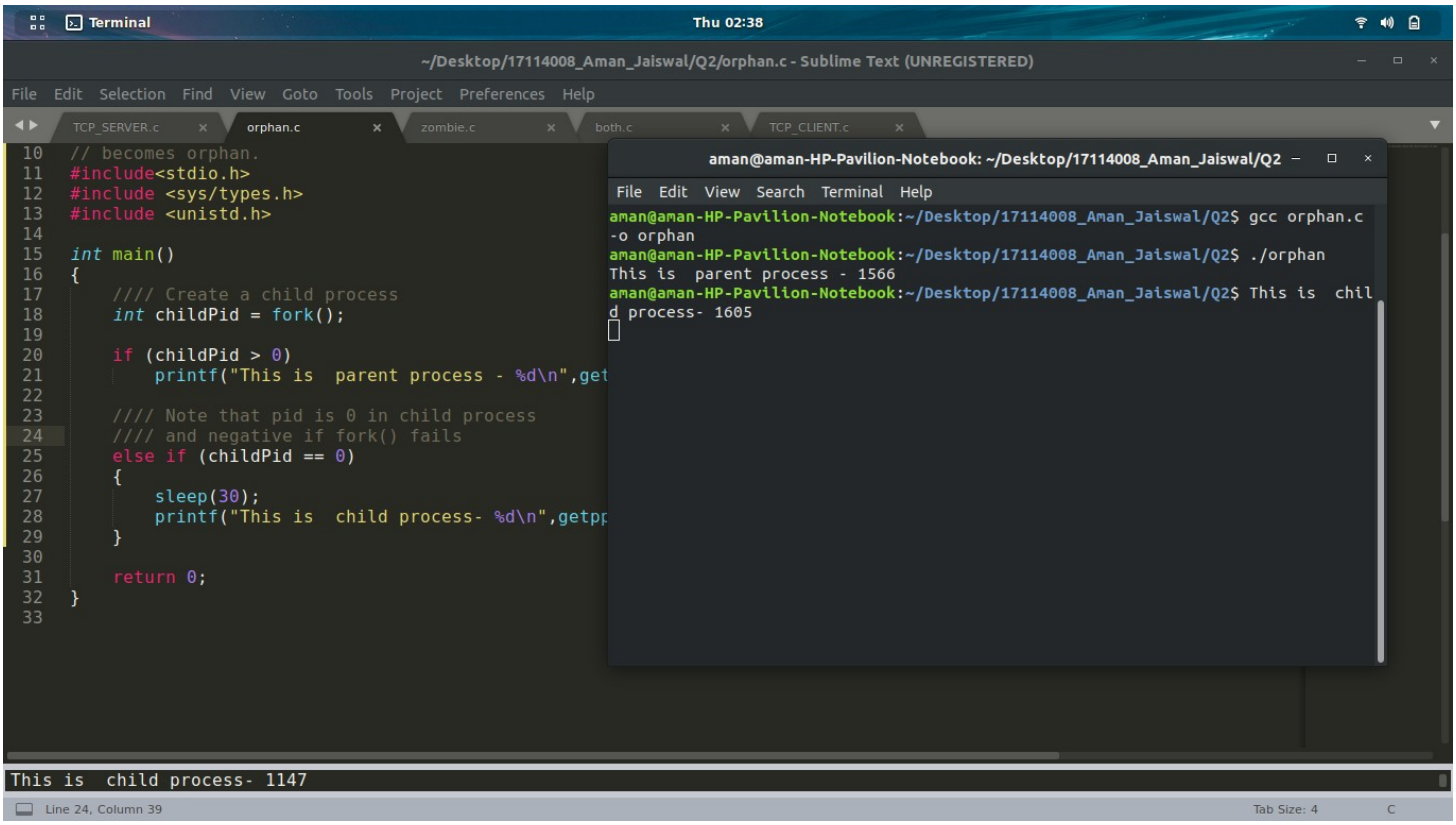
Write a C program to demonstrate both Zombie and Orphan process

An orphan process is a process that is still executing, but whose parent has died. They do not become zombie processes; instead, they are adopted by init (process ID 1), which waits on its children.
A zombie process is a process that has completed execution but still has an entry in the process table. This entry is still needed to allow the parent process to read its child's exit status.
A process that terminates cannot leave the system until its parent accepts its return code. If its parent process is already dead, it'll already have been adopted by the "init" process, which always accepts its children's return codes. However, if a process's parent is alive but never executes a wait ( ), the process's return code will never be accepted and the process will remain a zombie

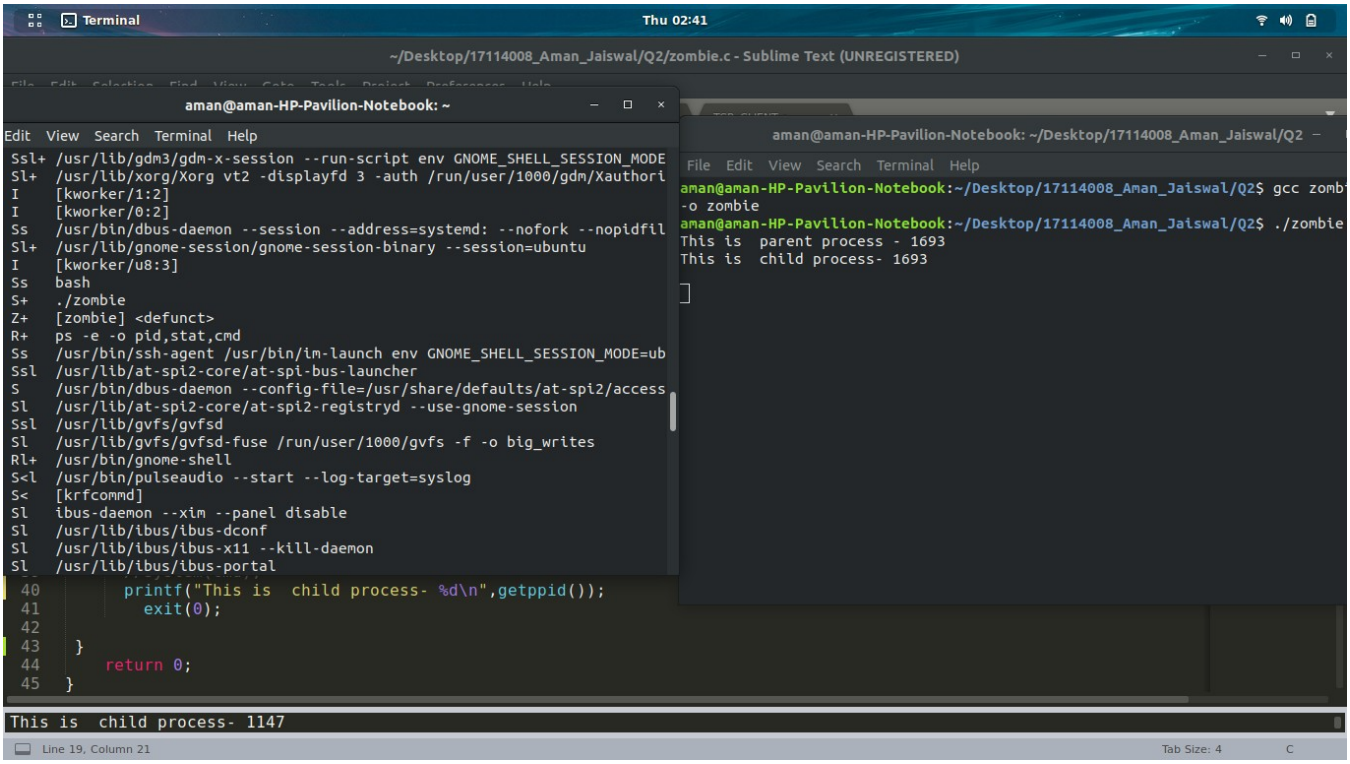
- Data Structure and Functions used:-

1. child finishes its execution using exit()
2. fork() is used for creating new child of parent.
3. ChildPid is used for storing child id.
4. getpid(),getppid()

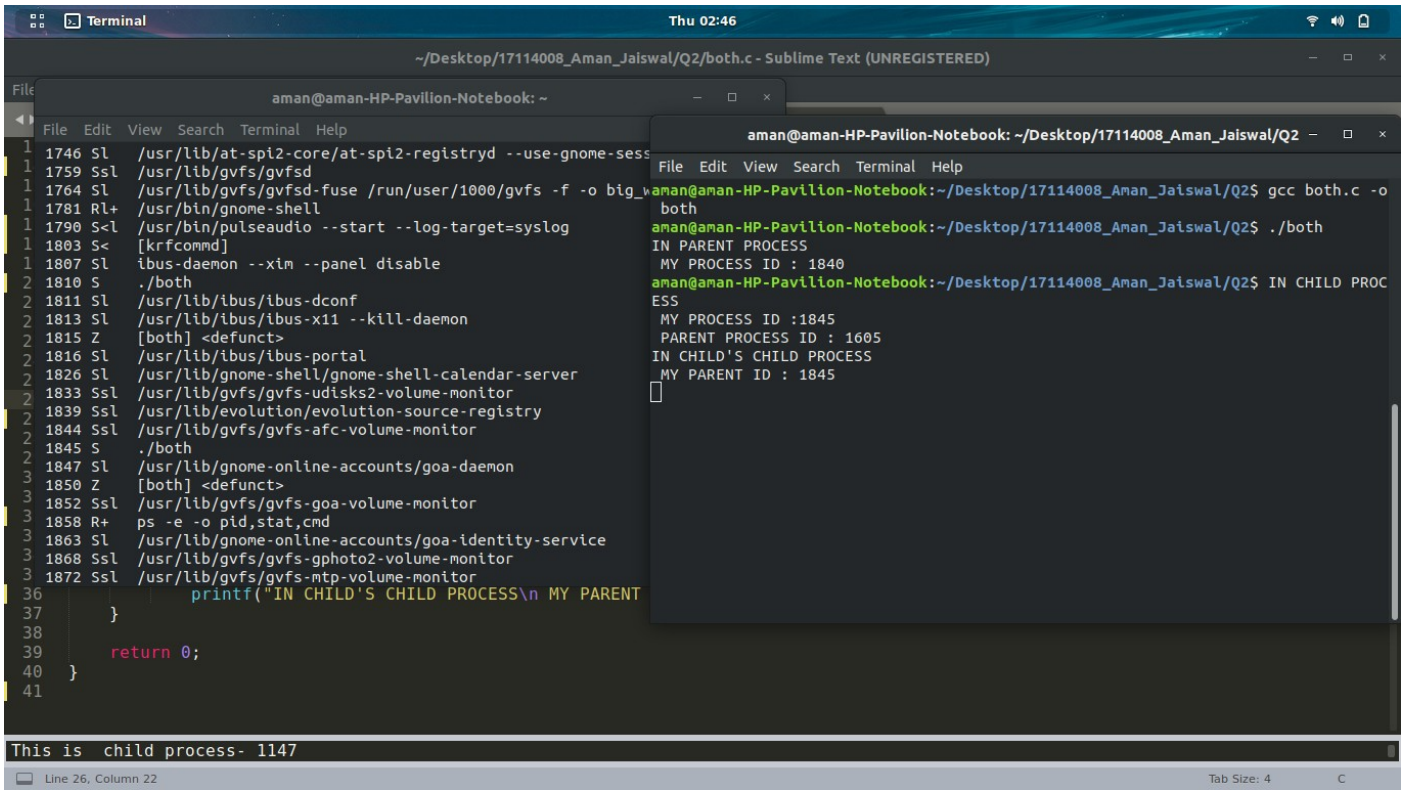
1.Orphan- childs parent id is not same as the original parent that proves that it is a oprphan process.



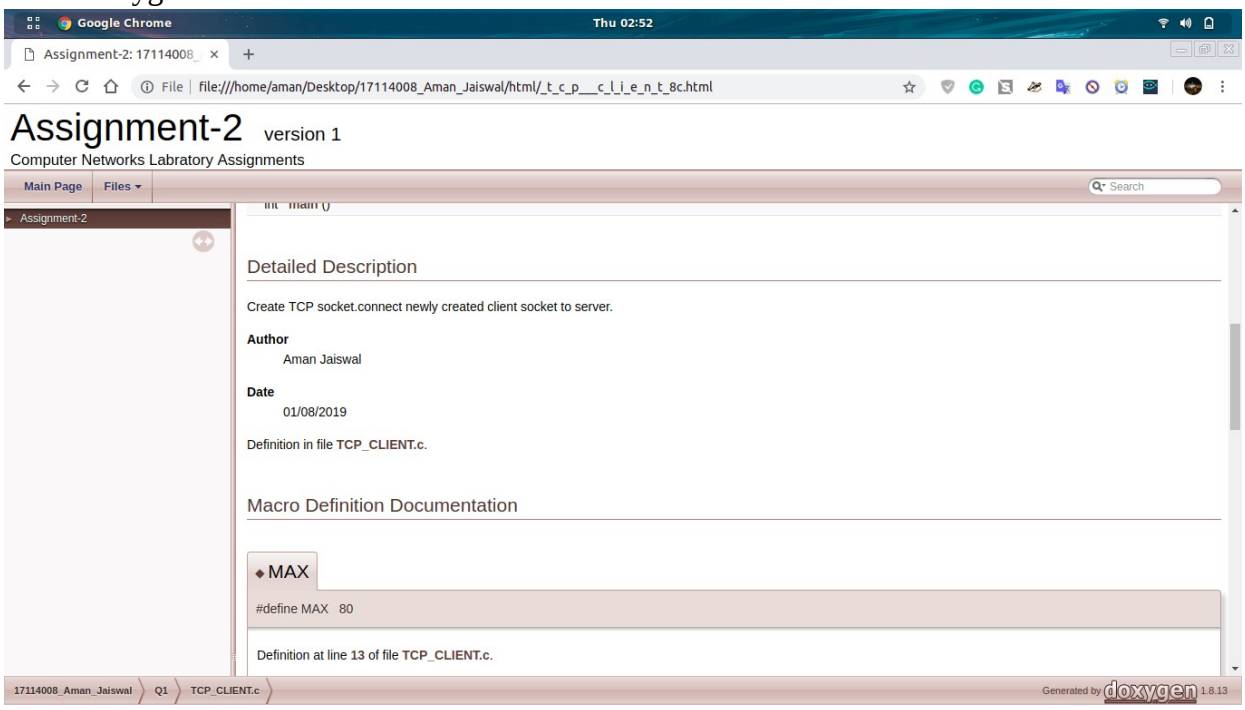
2.Zombie process:- Z+ symbol can be seen process chart for zombie process.

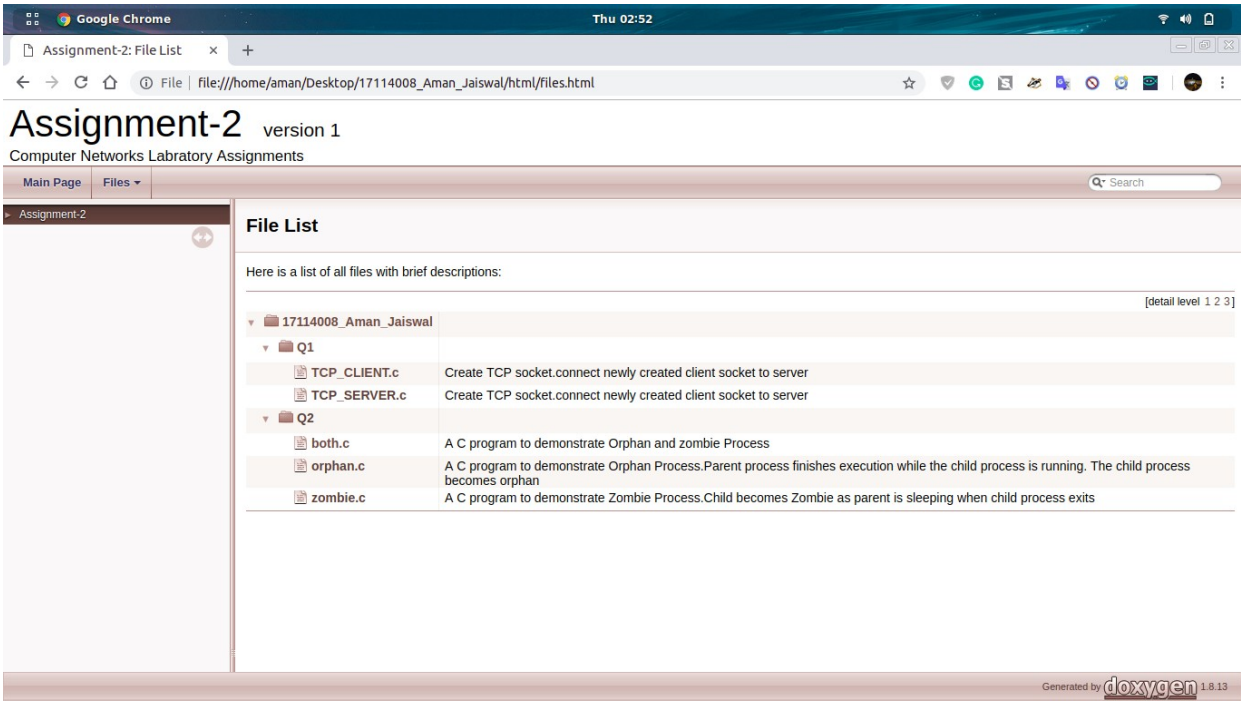


2. Both together:-



- Doxygen File :-





- GNU Debbuger:-