Lab Assignment-2
CSN-361: Computer Networks Laboratory

THEEGALA LOKESH REDDY
Enrollment No:-17114073
B.tech CSE 3 rd 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:

struct sockaddr_in servaddr : sockaddr_in structure for storing server address

sockfd = socket(AF_INET, SOCK_STREAM, 0);

sockfd: socket descriptor, an integer (like a file-handle)

AF_INET (eg IPv4 protocol) is domain

SOCK_STREAM: TCP(reliable, connection oriented)

SOCK_DGRAM: UDP(unreliable, connectionless)

protocol: Protocol value for Internet Protocol(IP), which is 0 , confirms it is working

2. Bind:

(bind(sockfd, (SA*)&servaddr, sizeof(servaddr)))

After creation of the socket, bind function binds the socket to the address and port number

specified in addr. In this we bind the server to the localhost, hence we use INADDR_ANY to specify the IP address of our system .

3. Listen:

(listen(sockfd, n))_

It puts the server socket in waiting for the client to approach the server to make a connection. Here n defines the no of clients u can connect.

4. Accept:

connfd = accept(sockfd, (SA*)&cli, &len);

It extracts the first connection request 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.

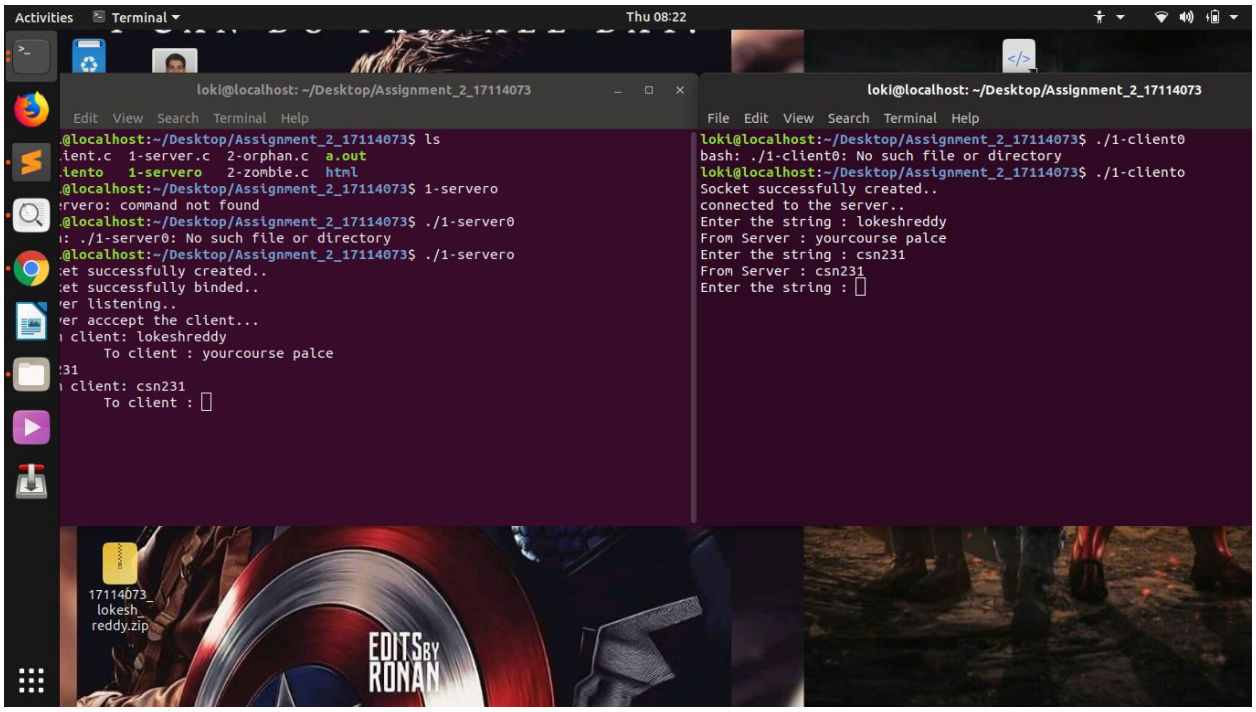
For Client:-

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

2. Connect:

connect(sockfd, (SA*)&servaddr, sizeof(servaddr);

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:

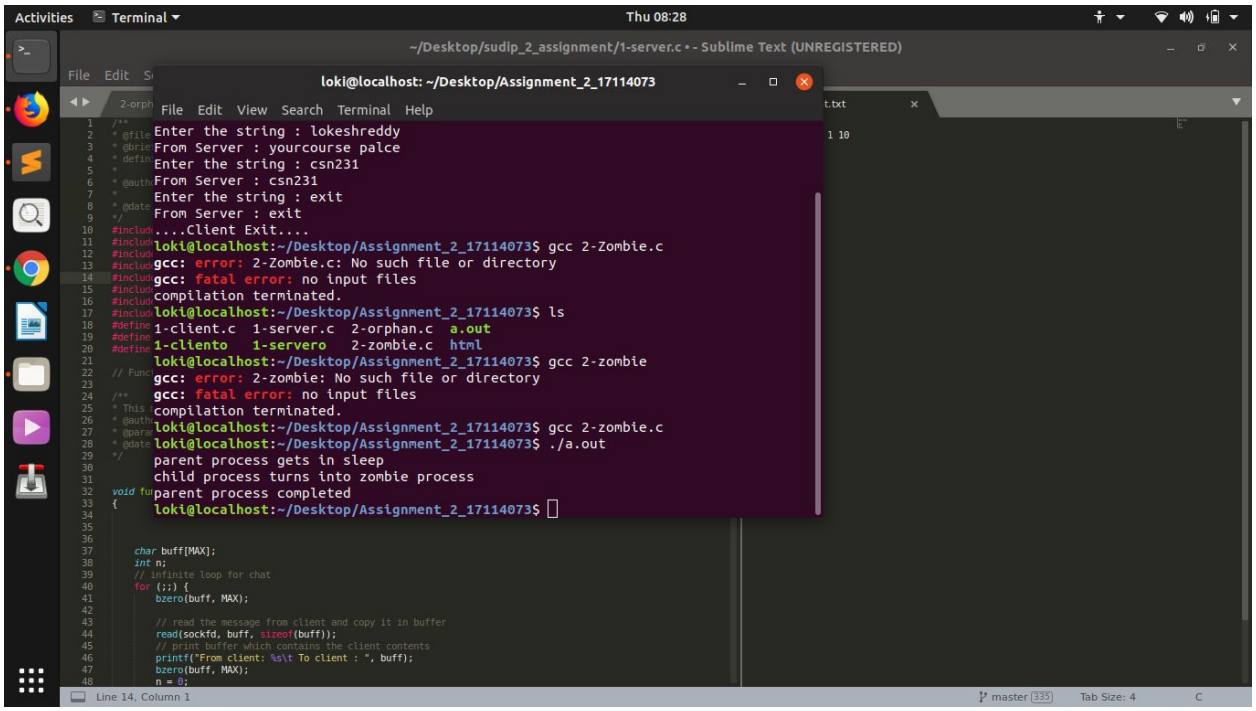
Write a C program to demonstrate both Zombie and Orphan process

Zombie:

Data Structure and Functions used:-

1. fork() is used for creating new child of parent.
2.child finishes its execution using exit() .
3.int Child_pid is used for storing child id.
4.char * cmd= ps -e -o pid,stat,cmd command for printing all process such that we can see zombie process .
5.system() system call to execute cmd.

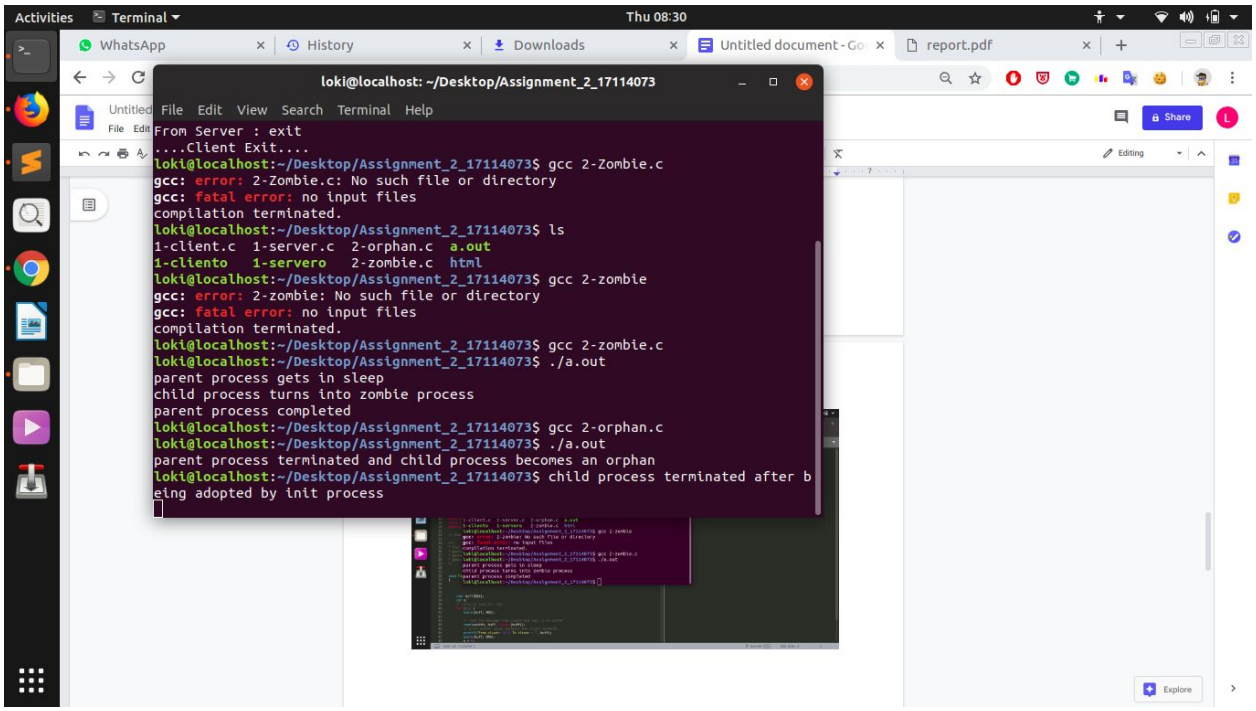
Brief:
when child completes its completion of process, but if parent didnt call wait() to get exit status child procees will still present in process talble even it's process is completed.



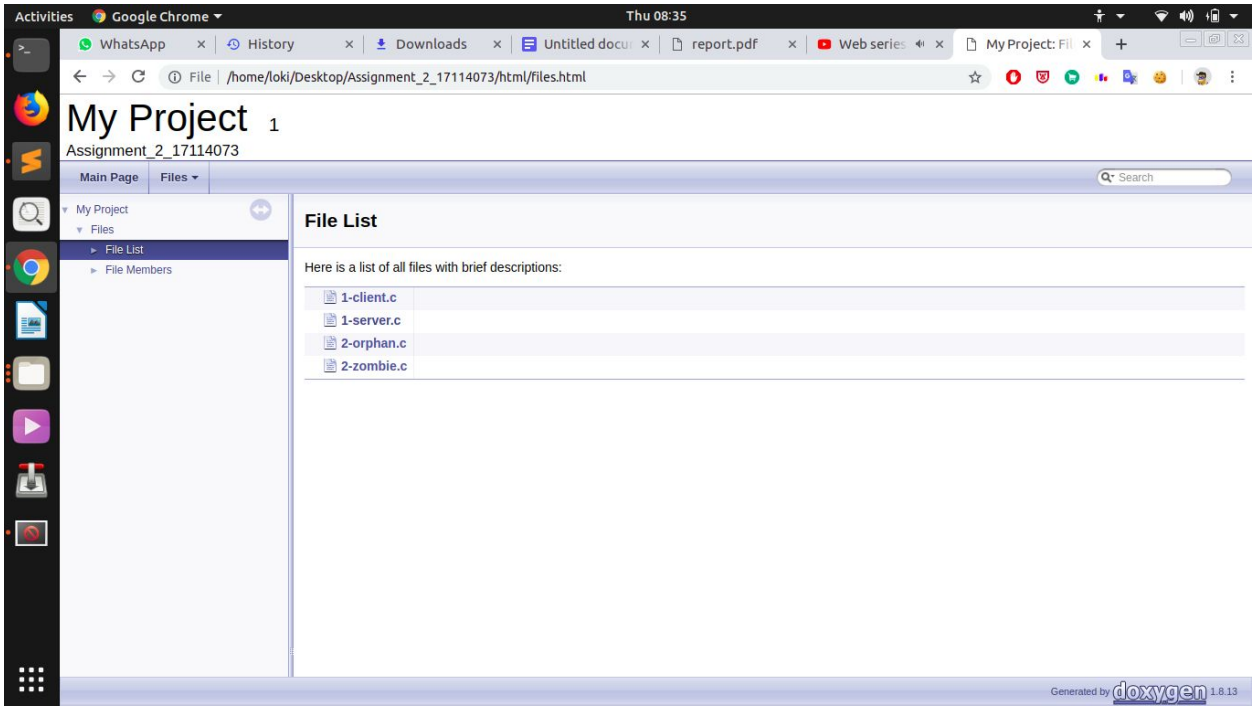
Orphan:
Data Structures and functions

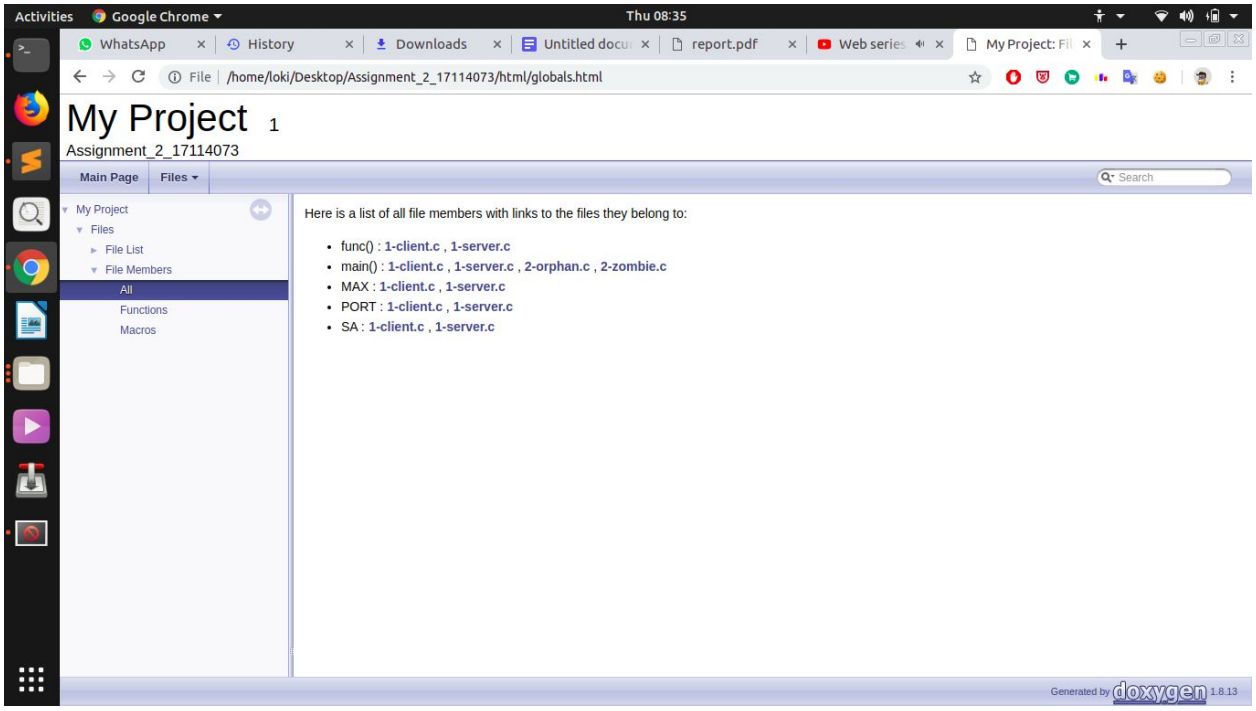
1.fork() is used for creating new child of parent.
2. exit() parent finishes its execution using it .
3.int Child_pid is used for storing child id.
4. sleep() used for child process to wait

Brief: when parent completes it process and exit without taking exit status from child, child doesn't have parent after that init will adopt the child .



sreen shots of doxygen:


Activities
Google Chrome
Thu 08:35
WhatsApp
History
Downloads
Untitled docu
report.pdf
Web series
My Project: Fil
File
/home/loki/Desktop/Assignment_2_17114073/html/files.html
My Project 1
Assignment_2_17114073
Main Page
Files
Search
My Project
Files
File List
File Members
File List
Here is a list of all files with brief descriptions:
1-client.c
1-server.c
2-orphan.c
2-zombie.c
Generated by doxygen 1.8.13


Activities
Google Chrome
Thu 08:35
WhatsApp
History
Downloads
Untitled docu
report.pdf
Web series
My Project: Fil
File
/home/loki/Desktop/Assignment_2_17114073/html/globals.html
My Project 1
Assignment_2_17114073
Main Page
Files
Search
My Project
Files
File List
File Members
All
Functions
Macros
Here is a list of all file members with links to the files they belong to:
func() : 1-client.c , 1-server.c
main() : 1-client.c , 1-server.c , 2-orphan.c , 2-zombie.c
MAX : 1-client.c , 1-server.c
PORT : 1-client.c , 1-server.c
SA : 1-client.c , 1-server.c
Generated by doxygen 1.8.13