

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);`

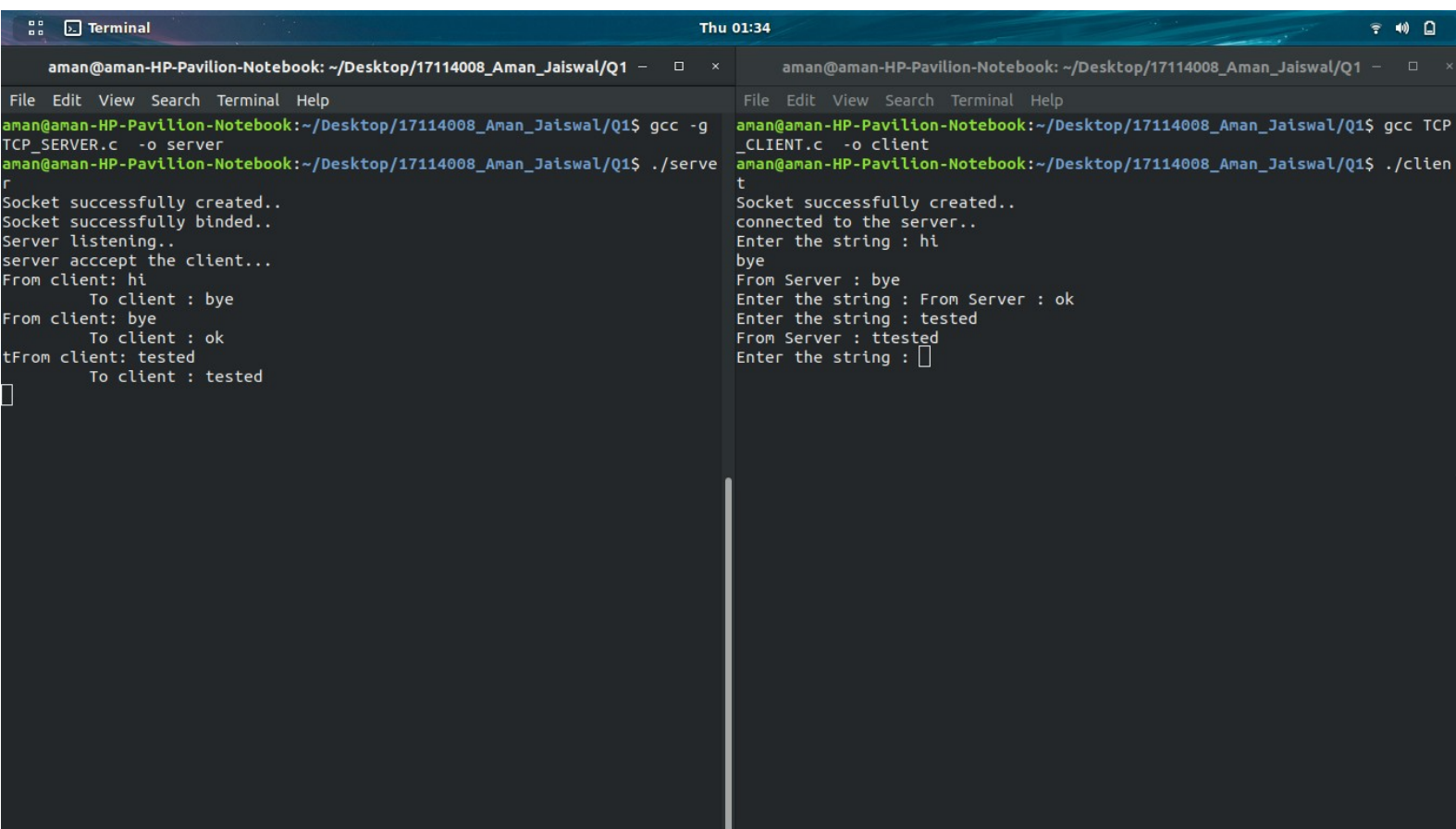
It extracts the first connection request on the queue of pending connections for the listening socket, sockfd, creates a new connected socket, and returns a new file descriptor referring to that socket. At this point, connection is established between client and server, and they are ready to transfer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



```
aman@aman-HP-Pavilion-Notebook: ~/Desktop/17114008_Aman_Jaiswal/Q1
File Edit View Search Terminal Help
aman@aman-HP-Pavilion-Notebook:~/Desktop/17114008_Aman_Jaiswal/Q1$ gcc -g TCP_SERVER.c -o server
aman@aman-HP-Pavilion-Notebook:~/Desktop/17114008_Aman_Jaiswal/Q1$ ./server
Socket successfully created..
Socket successfully binded..
Server listening..
server accept the client...
From client: hi
    To client : bye
From client: bye
    To client : ok
From client: tested
    To client : tested

aman@aman-HP-Pavilion-Notebook: ~/Desktop/17114008_Aman_Jaiswal/Q1
File Edit View Search Terminal Help
aman@aman-HP-Pavilion-Notebook:~/Desktop/17114008_Aman_Jaiswal/Q1$ gcc TCP_CLIENT.c -o client
aman@aman-HP-Pavilion-Notebook:~/Desktop/17114008_Aman_Jaiswal/Q1$ ./client
Socket successfully created..
connected to the server..
Enter the string : hi
bye
From Server : bye
Enter the string : From Server : ok
Enter the string : tested
From Server : ttested
Enter the string : 
```

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.

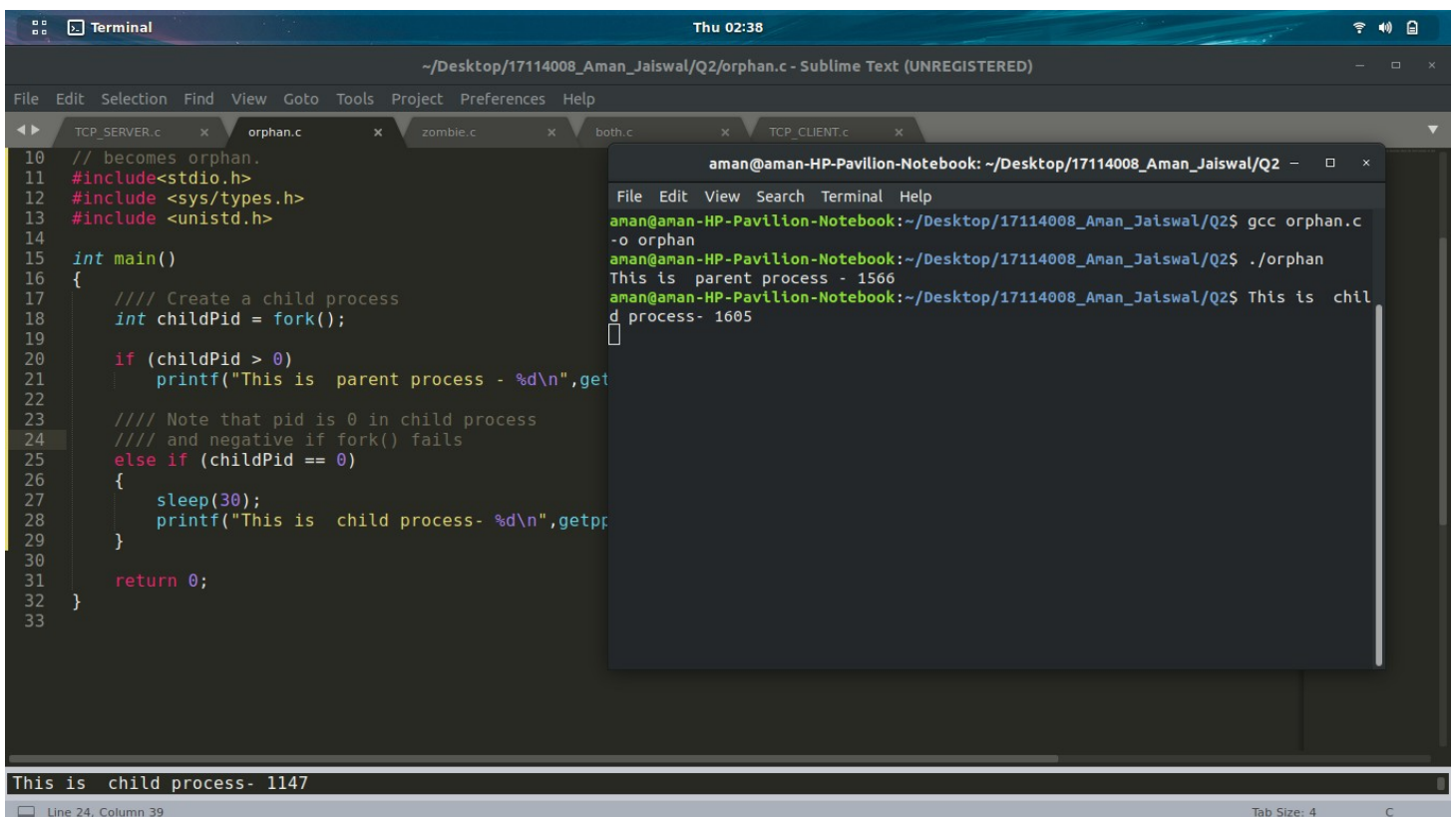
A zombie process is a process that has completed execution but still has an entry in the process table. This entry is still needed to allow the parent process to read its child's exit status.

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- child's parent id is not same as the original parent that proves that it is a orphan process.



```
10 // becomes orphan.
11 #include<stdio.h>
12 #include <sys/types.h>
13 #include <unistd.h>
14
15 int main()
16 {
17     /// Create a child process
18     int childPid = fork();
19
20     if (childPid > 0)
21         printf("This is parent process - %d\n",getpid());
22
23     /// Note that pid is 0 in child process
24     /// and negative if fork() fails
25     else if (childPid == 0)
26     {
27         sleep(30);
28         printf("This is child process- %d\n",getpid());
29     }
30
31     return 0;
32 }
33
```

aman@aman-HP-Pavilion-Notebook: ~/Desktop/17114008_Aman_Jaiswal/Q2 - x

File Edit View Search Terminal Help

aman@aman-HP-Pavilion-Notebook:~/Desktop/17114008_Aman_Jaiswal/Q2\$ gcc orphan.c

-o orphan

aman@aman-HP-Pavilion-Notebook:~/Desktop/17114008_Aman_Jaiswal/Q2\$./orphan

This is parent process - 1566

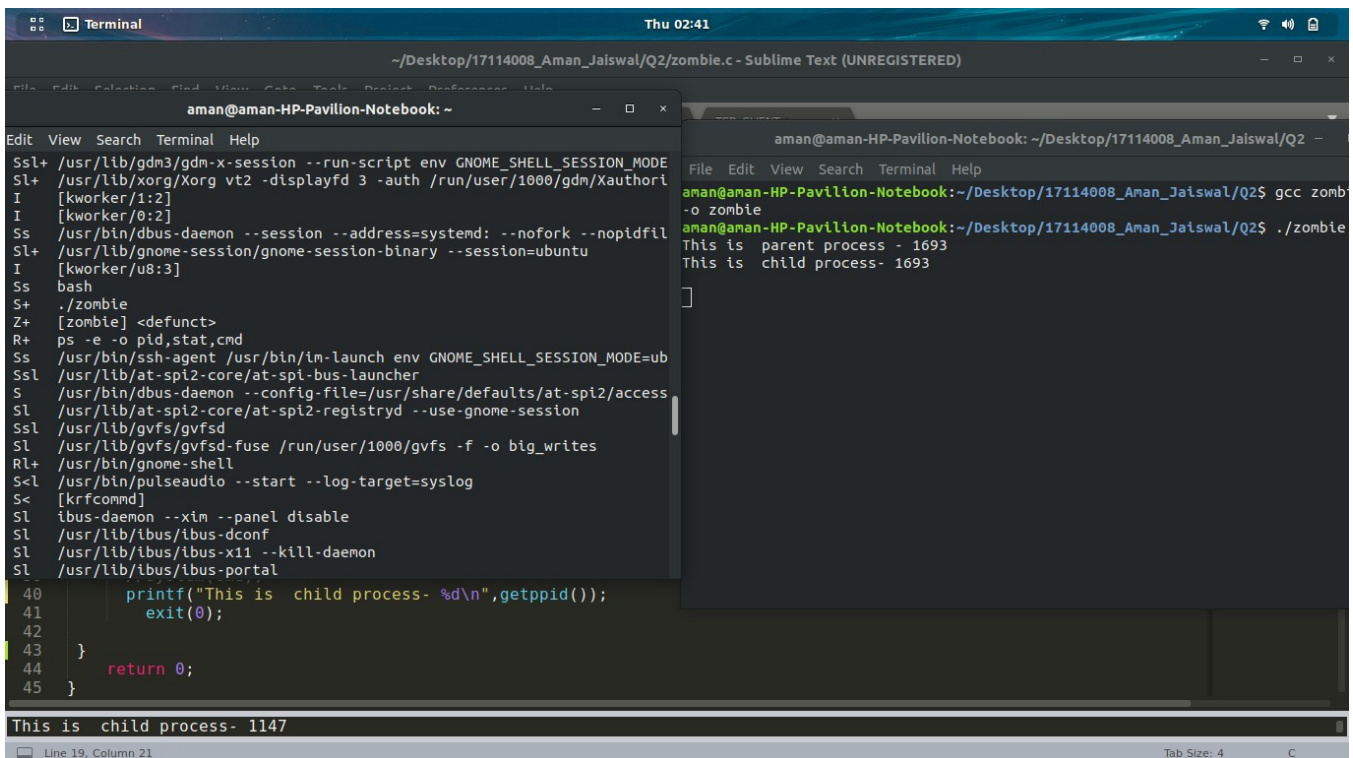
aman@aman-HP-Pavilion-Notebook:~/Desktop/17114008_Aman_Jaiswal/Q2\$ This is child process- 1605

This is child process- 1147

Line 24, Column 39

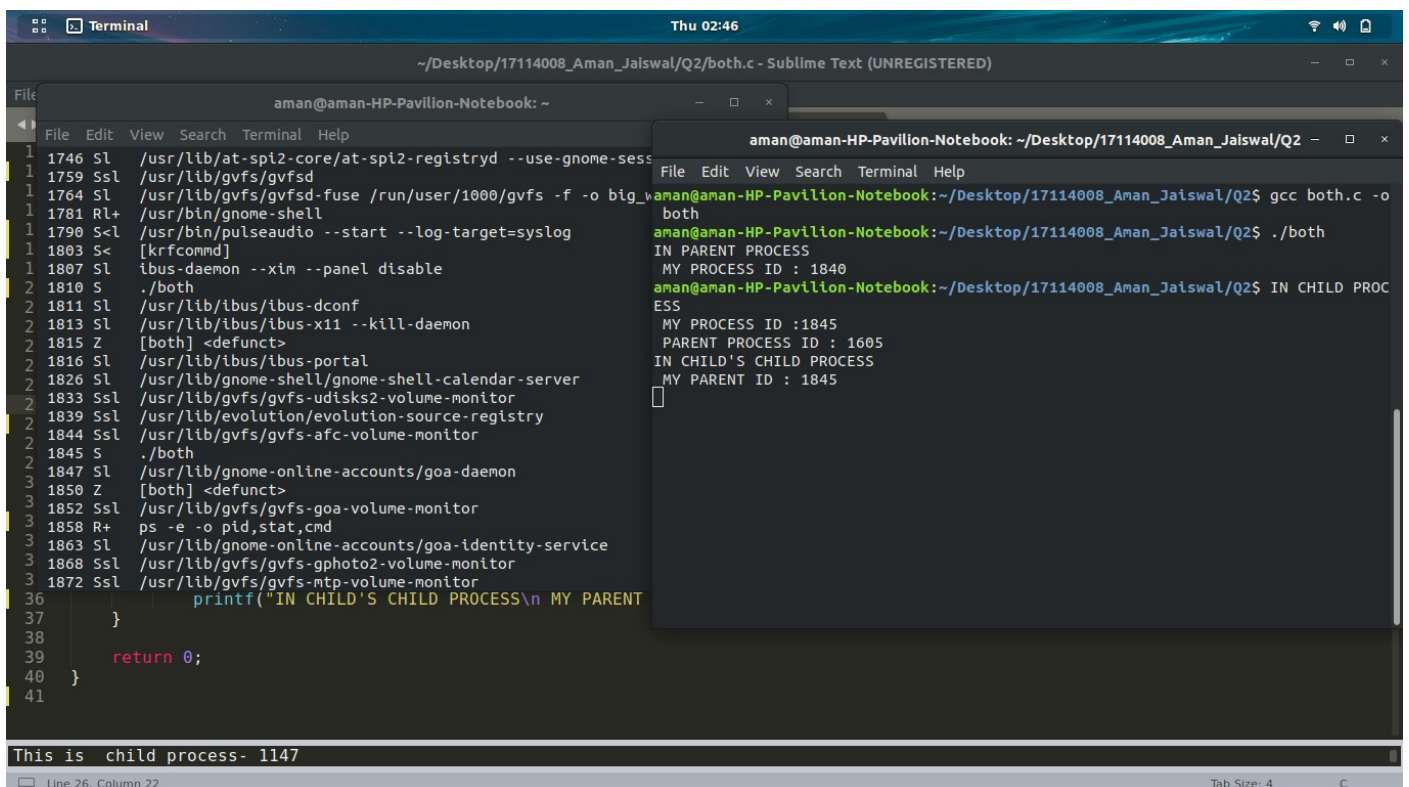
Tab Size: 4

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



```
aman@aman-HP-Pavilion-Notebook: ~  
Ssl+ /usr/lib/gdm3/gdm-x-session --run-script env GNOME_SHELL_SESSION_MODE  
Ssl+ /usr/lib/xorg/Xorg vt2 -displayfd 3 -auth /run/user/1000/gdm/Xauthort  
I [kworker/1:2]  
I [kworker/0:2]  
Ss /usr/bin/dbus-daemon --session --address=systemd: --nofork --nopidfil  
Ssl+ /usr/lib/gnome-session/gnome-session-binary --session=ubuntu  
I [kworker/u8:3]  
Ss bash  
S+ ./zombie  
Z+ [zombie] <defunct>  
R+ ps -e -o pid,stat,cmd  
Ss /usr/bin/ssh-agent /usr/bin/ln-launch env GNOME_SHELL_SESSION_MODE=ub  
Ssl /usr/lib/at-spi2-core/at-spi-bus-launcher  
S /usr/bin/dbus-daemon --config-file=/usr/share/defaults/at-spi2/access  
Ssl /usr/lib/at-spi2-core/at-spi2-registrd --use-gnome-session  
Ssl /usr/lib/gvfs/gvfsd  
Ssl /usr/lib/gvfs/gvfsd-fuse /run/user/1000/gvfs -f -o big_writes  
Rl+ /usr/bin/gnome-shell  
S<l /usr/bin/pulseaudio --start --log-target=syslog  
S< [krfcomm]  
Ssl ibus-daemon --xim --panel disable  
Ssl /usr/lib/ibus/ibus-dconf  
Ssl /usr/lib/ibus/ibus-x11 --kill-daemon  
Ssl /usr/lib/ibus/ibus-portal  
40 printf("This is child process- %d\n",getppid());  
41 exit(0);  
42  
43 }  
44 return 0;  
45 }  
This is child process- 1147
```

2. Both together:-



```
aman@aman-HP-Pavilion-Notebook: ~  
1746 Ssl /usr/lib/at-spi2-core/at-spi2-registrd --use-gnome-sess  
1759 Ssl /usr/lib/gvfs/gvfsd  
1764 Ssl /usr/lib/gvfs/gvfsd-fuse /run/user/1000/gvfs -f -o big_w  
1781 Rl+ /usr/bin/gnome-shell  
1790 S<l /usr/bin/pulseaudio --start --log-target=syslog  
1803 S< [krfcomm]  
1807 Ssl ibus-daemon --xim --panel disable  
2 1810 S ./both  
2 1811 Ssl /usr/lib/ibus/ibus-dconf  
2 1813 Ssl /usr/lib/ibus/ibus-x11 --kill-daemon  
2 1815 Z [both] <defunct>  
2 1816 Ssl /usr/lib/ibus/ibus-portal  
2 1826 Ssl /usr/lib/gnome-shell/gnome-shell-calendar-server  
2 1833 Ssl /usr/lib/gvfs/gvfs-udisks2-volume-monitor  
2 1839 Ssl /usr/lib/evolution/evolution-source-registry  
2 1844 Ssl /usr/lib/gvfs/gvfs-afc-volume-monitor  
2 1845 S ./both  
2 1847 Ssl /usr/lib/gnome-online-accounts/goa-daemon  
3 1850 Z [both] <defunct>  
3 1852 Ssl /usr/lib/gvfs/gvfs-go-a-volume-monitor  
3 1858 R+ ps -e -o pid,stat,cmd  
3 1863 Ssl /usr/lib/gnome-online-accounts/goa-identity-service  
3 1868 Ssl /usr/lib/gvfs/gvfs-gphoto2-volume-monitor  
3 1872 Ssl /usr/lib/gvfs/gvfs-mtp-volume-monitor  
36 printf("IN CHILD'S CHILD PROCESS\n MY PARENT  
37 }  
38  
39 return 0;  
40 }  
41  
This is child process- 1147
```

- Doxygen File :-

Assignment-2 version 1
Computer Networks Labratory Assignments

Main Page Files

Assignment-2

Detailed Description

Create TCP socket.connect newly created client socket to server.

Author
Aman Jaiswal

Date
01/08/2019

Definition in file TCP_CLIENT.c.

Macro Definition Documentation

MAX

```
#define MAX 80
```

Definition at line 13 of file TCP_CLIENT.c.

17114008_Aman_Jaiswal Q1 TCP_CLIENT.c Generated by doxygen 1.8.13

Assignment-2 version 1
Computer Networks Labratory Assignments

Main Page Files

Assignment-2

File List

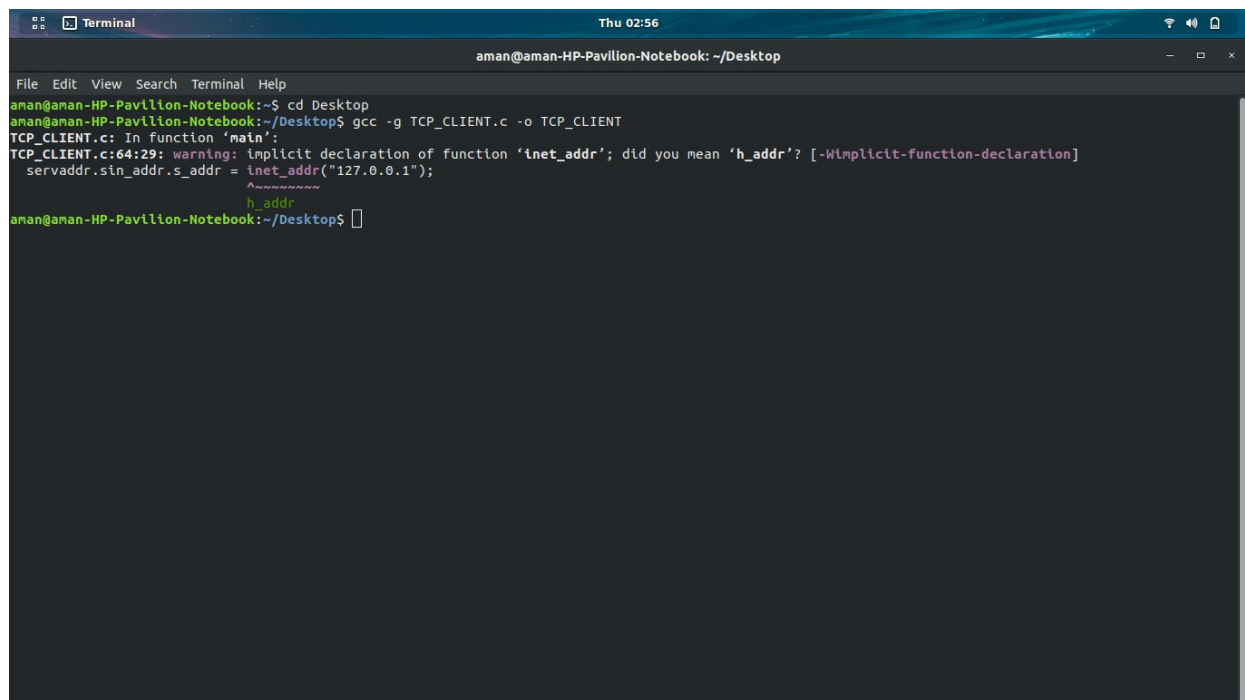
Here is a list of all files with brief descriptions:

[detail level 1 2 3]

- 17114008_Aman_Jaiswal
 - Q1
 - TCP_CLIENT.c Create TCP socket.connect newly created client socket to server
 - TCP_SERVER.c Create TCP socket.connect newly created client socket to server
 - Q2
 - both.c A C program to demonstrate Orphan and zombie Process
 - orphan.c A C program to demonstrate Orphan Process.Parent process finishes execution while the child process is running. The child process becomes orphan
 - zombie.c A C program to demonstrate Zombie Process.Child becomes Zombie as parent is sleeping when child process exits

Generated by doxygen 1.8.13

- GNU Debugger:-



```
aman@aman-HP-Pavillon-Notebook: ~/Desktop
File Edit View Search Terminal Help
aman@aman-HP-Pavillon-Notebook:~$ cd Desktop
aman@aman-HP-Pavillon-Notebook:~/Desktop$ gcc -g TCP_CLIENT.c -o TCP_CLIENT
TCP_CLIENT.c: In function 'main':
TCP_CLIENT.c:64:29: warning: implicit declaration of function 'inet_addr'; did you mean 'h_addr'? [-Wimplicit-function-declaration]
    servaddr.sin_addr.s_addr = inet_addr("127.0.0.1");
                              ^
                              h_addr
aman@aman-HP-Pavillon-Notebook:~/Desktop$
```