

Lab Assignment-2

CSN-361: Computer Networks Laboratory

SURESH BABU GANGAVRAPU
Enrollment No:-17114030
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:

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.

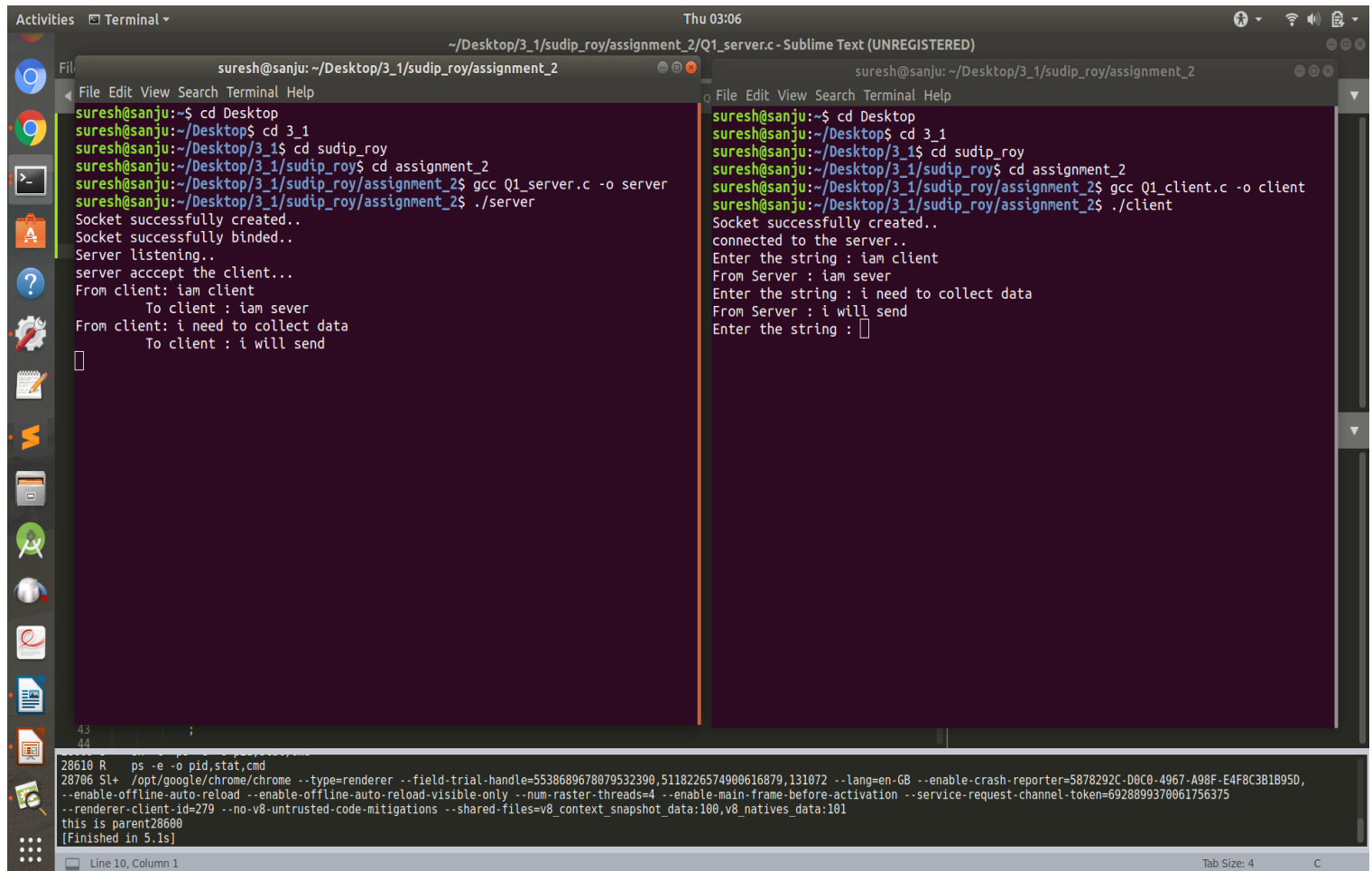
For Client:-

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`



```
suresh@sanju: ~/Desktop/3_1/sudip_roy/assignment_2
File Edit View Search Terminal Help
suresh@sanju:~$ cd Desktop
suresh@sanju:~/Desktop$ cd 3_1
suresh@sanju:~/Desktop/3_1$ cd sudip_roy
suresh@sanju:~/Desktop/3_1/sudip_roy$ cd assignment_2
suresh@sanju:~/Desktop/3_1/sudip_roy/assignment_2$ gcc Q1_server.c -o server
suresh@sanju:~/Desktop/3_1/sudip_roy/assignment_2$ ./server
Socket successfully created..
Socket successfully binded..
Server listening..
server accept the client...
From client: iam client
To client : iam sever
From client: i need to collect data
To client : i will send
[]

suresh@sanju:~/Desktop/3_1/sudip_roy/assignment_2
File Edit View Search Terminal Help
suresh@sanju:~$ cd Desktop
suresh@sanju:~/Desktop$ cd 3_1
suresh@sanju:~/Desktop/3_1$ cd sudip_roy
suresh@sanju:~/Desktop/3_1/sudip_roy$ cd assignment_2
suresh@sanju:~/Desktop/3_1/sudip_roy/assignment_2$ gcc Q1_client.c -o client
suresh@sanju:~/Desktop/3_1/sudip_roy/assignment_2$ ./client
Socket successfully created..
connected to the server..
Enter the string : iam client
From Server : iam sever
Enter the string : i need to collect data
From Server : i will send
Enter the string : []

28610 R ps -e -o pid,stat,cmd
28706 Sl+ /opt/google/chrome/chrome --type=renderer --field-trial-handle=5538689678879532390,5118226574900616879,131072 --lang=en-GB --enable-crash-reporter=5878292C-D0C0-4967-A98F-E4F8C3B1B95D,
--enable-offline-auto-reload --enable-offline-auto-reload-visible-only --num-raster-threads=4 --enable-main-frame-before-activation --service-request-channel-token=6928899370061756375
--renderer-client-id=279 --no-v8-untrusted-code-mitigations --shared-files=v8_context_snapshot_data:100,v8_natives_data:101
this is parent28600
[Finished in 5.1s]
```

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.

We can we verify this by printing all the process

```
Activities Terminal Thu 02:40
~/Desktop/3_1/sudip_roy/assignment_2/Q2_zombie.c - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
hello.cpp x example.cpp x Q2_zombie.c x Q2_orphan.c x Q1
11 // A C program to demonstrate Zombie Process.
12 // Child becomes Zombie as parent is sleeping
13 // when child process exits.
14
15 // ps -e -o pid,stat,cmd command to check all running process
16
17 #include <stdlib.h>
18 #include <sys/types.h>
19 #include <unistd.h>
20 #include <stdio.h>
21
22 int main()
23 {
24     char * cmd=" ps -e -o pid,stat,cmd ";
25
26     // Fork returns process id
27     // in parent process
28     int child_pid = fork();
29
30     // Parent process
31     if (child_pid > 0)
32     {
33         printf("this is parent""%d\n", getpid());
34         sleep(5);
35         system(cmd);
36         sleep(5);
37     }
38
39     // Child process
40     else
41     {
42         printf("this is child""%d\n", getpid());
43         exit(0);
44     }
45 }
46
47
48
28610 R ps -e -o pid,stat,cmd
28706 Sl+ /opt/google/chrome/chrome --type=renderer --field-trial-handle=5538689678079532390,5118226574900616879,131072 --lang=en-GB --enable-crash-reporter=5878292C-D0C0-4967-A98F-E4F8C3B1B95D,
--enable-offline-auto-reload --enable-offline-auto-reload-visible-only --num-raster-threads=4 --enable-main-frame-before-activation --service-request-channel-token=6928899370061756375
--renderer-client-id=279 --no-v8-untrusted-code-mitigations --shared-files=v8_context_snapshot_data:100,v8_natives_data:101
this is parent28600
[Finished in 5.1s]
```

```
Activities Terminal Thu 02:40
~/Desktop/3_1/sudip_roy/assignment_2/Q2_zombie.c - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
hello.cpp x example.cpp x Q2_zombie.c x Q2_orphan.c x Q1
11 // A C program to demonstrate Zombie Process.
12 // Child becomes Zombie as parent is sleeping
13 // when child process exits.
14
15 // ps -e -o pid,stat,cmd command to check all running process
16
17 #include <stdlib.h>
18 #include <sys/types.h>
19 #include <unistd.h>
20 #include <stdio.h>
21
22 int main()
23 {
24     char * cmd=" ps -e -o pid,stat,cmd ";
25
26     // Fork returns process id
27     // in parent process
28     int child_pid = fork();
29
30     // Parent process
31     if (child_pid > 0)
32     {
33         printf("this is parent""%d\n", getpid());
34         sleep(5);
35         system(cmd);
36         sleep(5);
37     }
38
39     // Child process
40     else
41     {
42         printf("this is child""%d\n", getpid());
43         exit(0);
44     }
45 }
46
47
48
28610 R ps -e -o pid,stat,cmd
28706 Sl+ /opt/google/chrome/chrome --type=renderer --field-trial-handle=5538689678079532390,5118226574900616879,131072 --lang=en-GB --enable-crash-reporter=5878292C-D0C0-4967-A98F-E4F8C3B1B95D,
--enable-offline-auto-reload --enable-offline-auto-reload-visible-only --num-raster-threads=4 --enable-main-frame-before-activation --service-request-channel-token=6928899370061756375
--renderer-client-id=279 --no-v8-untrusted-code-mitigations --shared-files=v8_context_snapshot_data:100,v8_natives_data:101
this is parent28600
[Finished in 5.1s]
```

it is showing Z+ for child process pid 30979 which tells it zombie process

Orphan:

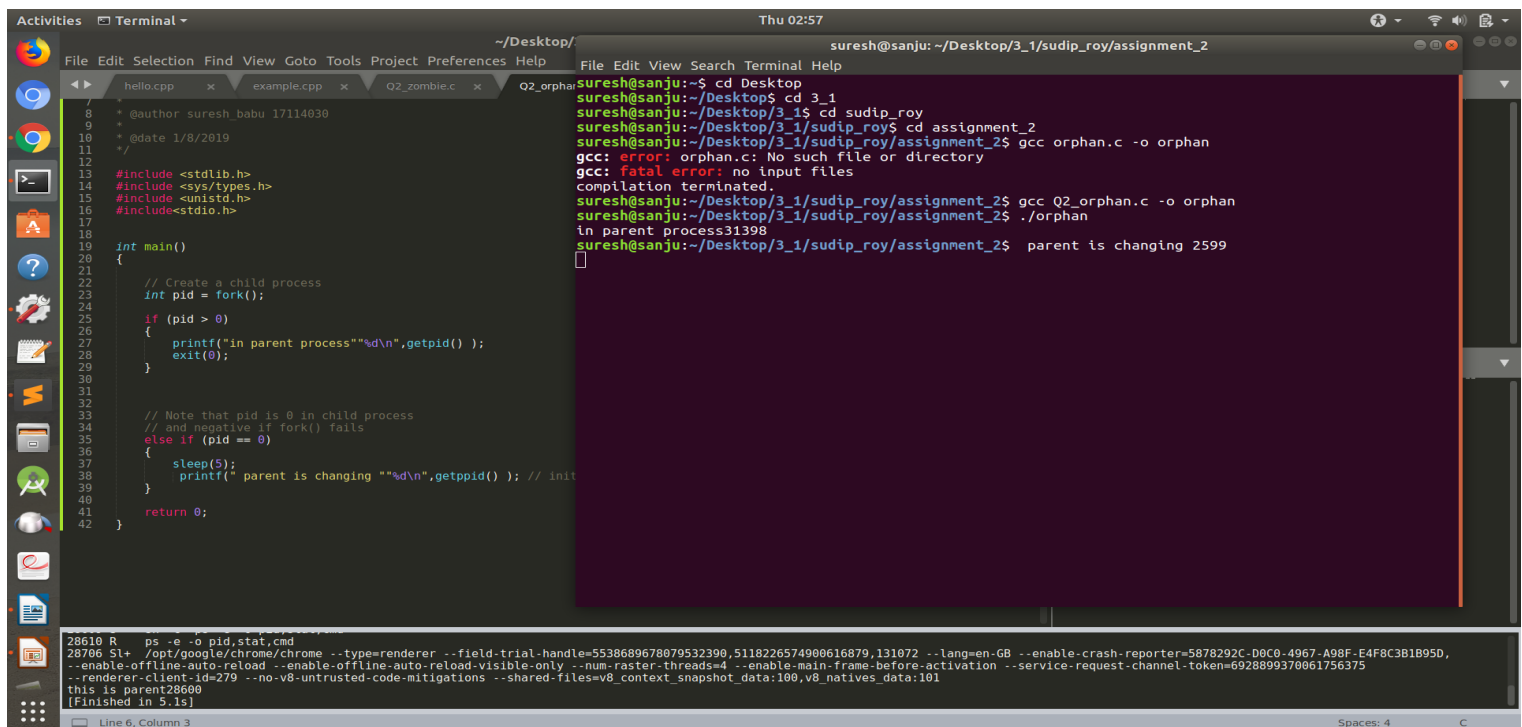
Data Structures and functions

- 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 its process and exit without taking exit status from child,
child doesn't have parent after that init will adopt the child .

We can check this printing the ppid of child ,it will not be same original parent



```
File Edit Selection Find View Goto Tools Project Preferences Help File Edit View Search Terminal Help
~/Desktop/
suresh@sanju: ~/Desktop/3_1/sudip_roy/assignment_2
suresh@sanju:~$ cd Desktop
suresh@sanju:~/Desktop$ cd 3_1
suresh@sanju:~/Desktop/3_1$ cd sudip_roy
suresh@sanju:~/Desktop/3_1/sudip_roy$ cd assignment_2
suresh@sanju:~/Desktop/3_1/sudip_roy/assignment_2$ gcc orphan.c -o orphan
gcc: error: orphan.c: No such file or directory
gcc: fatal error: no input files
compilation terminated.
suresh@sanju:~/Desktop/3_1/sudip_roy/assignment_2$ gcc Q2_orphan.c -o orphan
suresh@sanju:~/Desktop/3_1/sudip_roy/assignment_2$ ./orphan
in parent process31398
suresh@sanju:~/Desktop/3_1/sudip_roy/assignment_2$ parent is changing 2599
[]

7
8 * @author suresh_babu 17114030
9
10 * @date 1/8/2019
11 */
12
13 #include <stdlib.h>
14 #include <sys/types.h>
15 #include <unistd.h>
16 #include <stdio.h>
17
18 int main()
19 {
20
21
22 // Create a child process
23 int pid = fork();
24
25 if (pid > 0)
26 {
27     printf("in parent process""&dn",getpid() );
28     exit(0);
29 }
30
31
32 // Note that pid is 0 in child process
33 // and negative if fork() fails
34 else if (pid == 0)
35 {
36     sleep(5);
37     printf(" parent is changing ""&dn",getppid() ); // init
38 }
39
40 return 0;
41
42 }
```

```
28610 R ps -e -o pid,stat,cmd
28706 Sl+ /opt/google/chrome/chrome --type=renderer --field-trial-handle=5538689678679532390,5118226574900616879,131072 --lang=en-GB --enable-crash-reporter=5878292C-D0C0-4967-A98F-E4F8C3B1B95D,
--enable-offline-auto-reload --enable-offline-auto-reload-visible-only --num-raster-threads=4 --enable-main-frame-before-activation --service-request-channel-token=6928899370061756375
--renderer-client-id=279 --no-v8-untrusted-code-mitigations --shared-files=v8_context_snapshot_data:100,v8_natives_data:101
this is parent20600
[Finished in 5.1s]
```

Line 6, Column 3 Spaces: 4 C

screen shots of doxygen:

