Operating System Lab
CS342

Tanishq Malu | Lab:4 | 1901CS63

Q1 Write a program in C, which takes n, as user input, and create n number of zombie processes. Show that the created processes are zombie processes (ADD SCREENSHOT).

Compilation:
gcc -o q1 -q1.c

Syntax:
./q1
{n: user input}

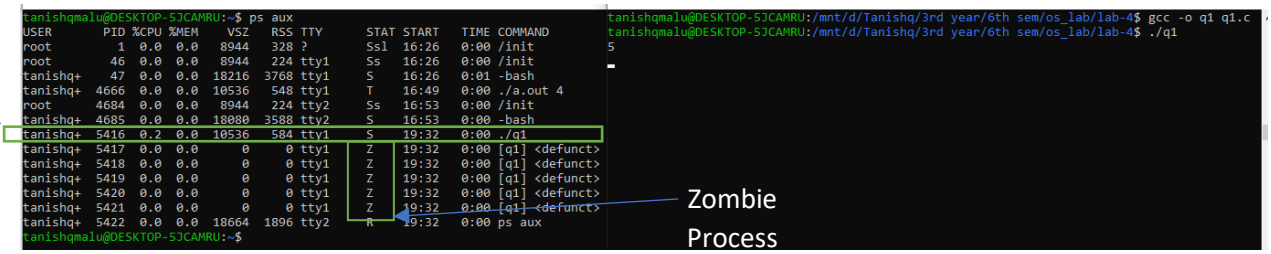
Note:
Press enter after typing ./q1, as we are not taking command line argument, rather input during execution as required in the question

Sample Input and Output:

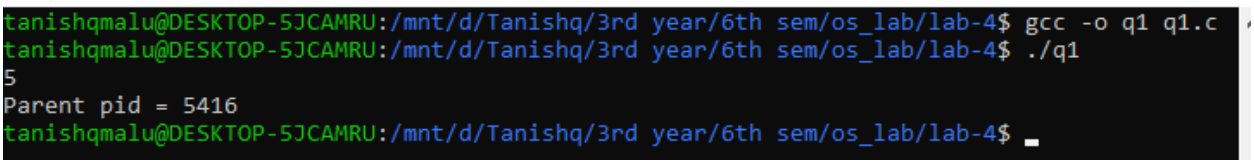
Input:
gcc -o q1 q1.c
./q1
5

Output:

Parent Process



After 20 seconds:



Q 2. Write a program in C, which takes n, as an user input, and create n number of orphan processes.

Compilation:
gcc -o q2 -q2.c

Syntax:
./q2
{n: user input}

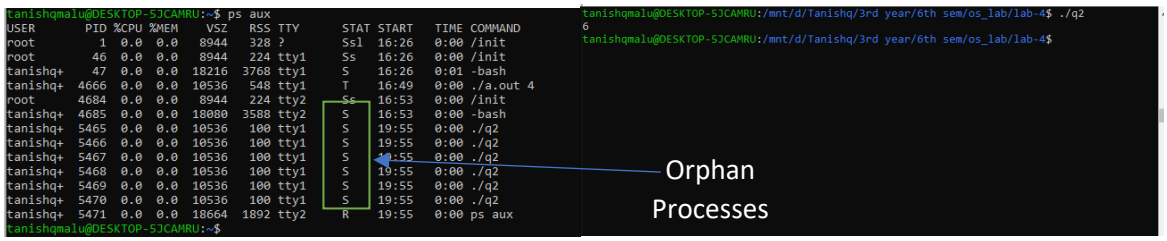
Note:
Press enter after typing ./q2, as we are not taking command line argument, rather input during execution as required in the question

Sample Input and Output:

Input:
gcc -o q2 q2.c
./q2
6

Output:

Parent process is already over and hence not present in here



After 20 seconds:

Q3. Write a program, which will

- - Take a user input N
- Make 2 child processes.
 - one child process should generate first N LUCAS sequence (https://en.wikipedia.org/wiki/Lucas_number).
- The 2nd child process should only print the LUCAS sequence (NOT GENERATE) (USE FILE OR SOME OTHER METHOD TO SHARE)

Compilation:
gcc -o q3 -q3.c

Syntax:
./q3
{n: user input}

Note:
Press enter after typing ./q3, as we are not taking command line argument, rather input during execution as required in the question

Sample Input and Output:

Case Number	1	2	3	4
Input	gcc -o q3 q3.c ./q3 8	gcc -o q3 q3.c ./q3 15	gcc -o q3 q3.c ./q3 0	gcc -o q3 q3.c ./q3 1

Output:

```
tanishqmalu@DESKTOP-5JCAMRU:/mnt/d/Tanishq/3rd year/6th sem/os_lab/lab-4$ gcc -o q3 q3.c
tanishqmalu@DESKTOP-5JCAMRU:/mnt/d/Tanishq/3rd year/6th sem/os_lab/lab-4$ ./q3
8
Generating lucas number, pid = 5484
printing lucas number, pid = 5483
2 1 3 4 7 11 18 29
tanishqmalu@DESKTOP-5JCAMRU:/mnt/d/Tanishq/3rd year/6th sem/os_lab/lab-4$ ./q3
15
Generating lucas number, pid = 5487
printing lucas number, pid = 5486
2 1 3 4 7 11 18 29 47 76 123 199 322 521 843
tanishqmalu@DESKTOP-5JCAMRU:/mnt/d/Tanishq/3rd year/6th sem/os_lab/lab-4$ ./q3
0
Generating lucas number, pid = 5490
printing lucas number, pid = 5489

tanishqmalu@DESKTOP-5JCAMRU:/mnt/d/Tanishq/3rd year/6th sem/os_lab/lab-4$ ./q3
1
Generating lucas number, pid = 5493
printing lucas number, pid = 5492
2
tanishqmalu@DESKTOP-5JCAMRU:/mnt/d/Tanishq/3rd year/6th sem/os_lab/lab-4$
```

Q4. Write a program which will
- - Make 3 threads:
- One will copy the source program to another file f2.
- second will print contents of f2.
- Third will delete the file f2.

Compilation:
gcc -o q4 -q4.c

Syntax:
./q4 file1.txt file2.txt

Sample Input and Output:

Input:
gcc -o q4 q4.c
./q4 file1.txt file2.txt

Output:

```
tanishqmalu@DESKTOP-5JCAMRU:/mnt/d/tanishq/3rd year/6th sem/OS_Lab/lab-4/1901CS63$ ./q4 file1.txt file2.txt

Task1: I will copy file f1 to another file f2. My PID: 31

Task2: I will print the new content of file f2. My PID: 32

--FILE 2--
Hello !
I am the content of file 1. But you copied me into file 2.
Now, I belong to file 2 as well
Anyway Happy coding !!


Task3: I will delete file 2. My PID: 33

Parent PID:: 30

tanishqmalu@DESKTOP-5JCAMRU:/mnt/d/tanishq/3rd year/6th sem/OS_Lab/lab-4/1901CS63$
```

---------- The End ----------