

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:

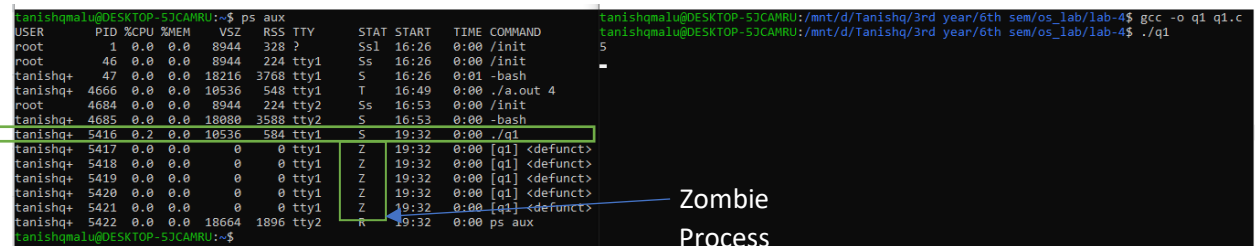
Input:

`gcc -o q1 q1.c`

`./q1`

5

Output:



```
tanishqmalu@DESKTOP-5JCAMRU:~$ ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0  8944   328 ?        Ss1  16:26   0:00 /init
root        46  0.0  0.0  8944   224 tty1     Ss   16:26   0:00 /init
tanishq+  47  0.0  0.0  18216  3768 tty1     S    16:26   0:01 -bash
tanishq+  466  0.0  0.0  10536   548 tty1     T    16:49   0:00 ./a.out 4
root       4684  0.0  0.0  8944   224 tty2     Ss   16:53   0:00 /init
tanishq+  4685  0.0  0.0  18080  3588 tty2     S    16:53   0:00 -bash
tanishq+  5416  0.2  0.0  10536   584 tty1     S    19:32   0:00 ./q1
tanishq+  5417  0.0  0.0      0      0 tty1     Z    19:32   0:00 [q1] <defunct>
tanishq+  5418  0.0  0.0      0      0 tty1     Z    19:32   0:00 [q1] <defunct>
tanishq+  5419  0.0  0.0      0      0 tty1     Z    19:32   0:00 [q1] <defunct>
tanishq+  5420  0.0  0.0      0      0 tty1     Z    19:32   0:00 [q1] <defunct>
tanishq+  5421  0.0  0.0      0      0 tty1     Z    19:32   0:00 [q1] <defunct>
tanishq+  5422  0.0  0.0  18664  1896 tty2     R    19:32   0:00 ps aux
tanishqmalu@DESKTOP-5JCAMRU:~$
```

After 20 seconds:

```
tanishqmalu@DESKTOP-5JCAMRU:/mnt/d/Tanishq/3rd year/6th sem/os_lab/lab-4$ gcc -o q1 q1.c
tanishqmalu@DESKTOP-5JCAMRU:/mnt/d/Tanishq/3rd year/6th sem/os_lab/lab-4$ ./q1
5
Parent pid = 5416
tanishqmalu@DESKTOP-5JCAMRU:/mnt/d/Tanishq/3rd year/6th sem/os_lab/lab-4$ _
```

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:

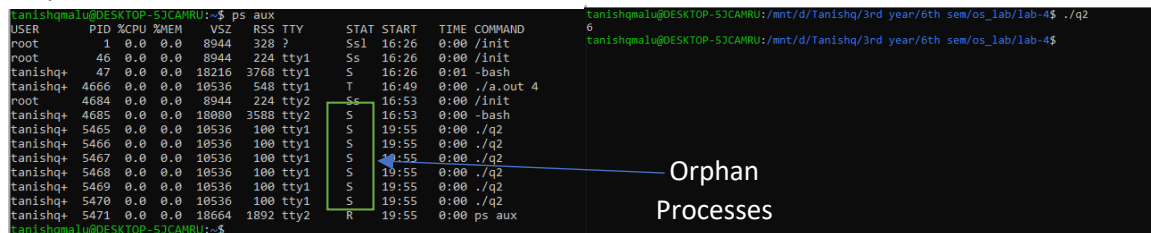
Input:

`gcc -o q2 q2.c`

`./q2`

6

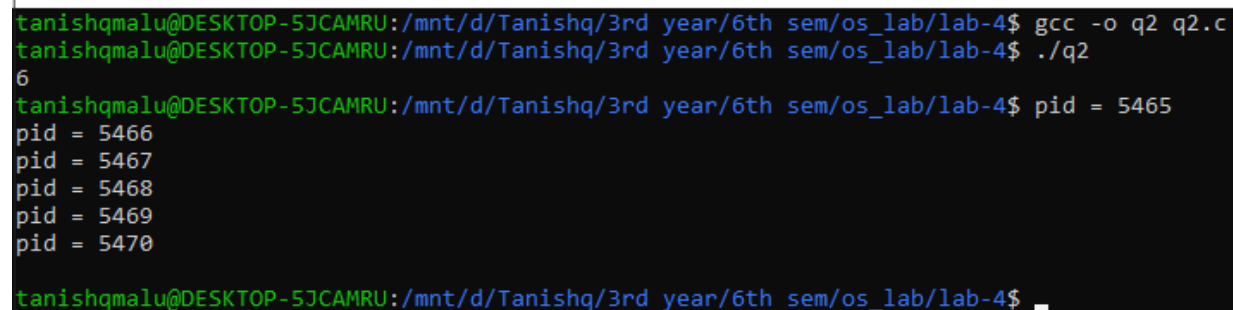
Output:



The screenshot shows a terminal window with the command `ps aux` executed. The output lists several processes, including the parent process (PID 4666) and six child processes (PIDs 5466-5471) that are orphaned. A green box highlights the child processes, and an arrow points from the text "Orphan Processes" to this box. Another arrow points from a text box on the left to the parent process.

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.0	0.0	8944	328	?	Ss1	16:26	0:00	/init
root	46	0.0	0.0	8944	224	tty1	Ss	16:26	0:00	/init
tanishq+	47	0.0	0.0	18216	3768	tty1	S	16:26	0:01	-bash
tanishq+	4666	0.0	0.0	10536	548	tty1	T	16:49	0:00	./a.out 4
root	4684	0.0	0.0	8944	224	tty2	Ss	16:53	0:00	/init
tanishq+	4685	0.0	0.0	18000	3508	tty2	S	16:53	0:00	-bash
tanishq+	5465	0.0	0.0	10536	100	tty1	S	19:55	0:00	./q2
tanishq+	5466	0.0	0.0	10536	100	tty1	S	19:55	0:00	./q2
tanishq+	5467	0.0	0.0	10536	100	tty1	S	19:55	0:00	./q2
tanishq+	5468	0.0	0.0	10536	100	tty1	S	19:55	0:00	./q2
tanishq+	5469	0.0	0.0	10536	100	tty1	S	19:55	0:00	./q2
tanishq+	5470	0.0	0.0	10536	100	tty1	S	19:55	0:00	./q2
tanishq+	5471	0.0	0.0	18664	1892	tty2	R	19:55	0:00	ps aux

After 20 seconds:



The screenshot shows the terminal output of the program execution. The user enters `gcc -o q2 q2.c`, `./q2`, and `6`. The program then prints the PIDs of the six orphan processes: 5466, 5467, 5468, 5469, and 5470.

```
tanishqmalu@DESKTOP-5JCAMRU:/mnt/d/Tanishq/3rd year/6th sem/os_lab/lab-4$ gcc -o q2 q2.c
tanishqmalu@DESKTOP-5JCAMRU:/mnt/d/Tanishq/3rd year/6th sem/os_lab/lab-4$ ./q2
6
tanishqmalu@DESKTOP-5JCAMRU:/mnt/d/Tanishq/3rd year/6th sem/os_lab/lab-4$ pid = 5465
pid = 5466
pid = 5467
pid = 5468
pid = 5469
pid = 5470
tanishqmalu@DESKTOP-5JCAMRU:/mnt/d/Tanishq/3rd year/6th sem/os_lab/lab-4$
```

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$ gcc -o q4 q4.c
tanishqmalu@DESKTOP-5JCAMRU:/mnt/d/Tanishq/3rd year/6th sem/os_lab/lab-4$ ./q4 file1.txt file2.txt

Task1: I will copy file f1 to another file f2. My PID: 5501

--FILE 2 Contents before copying--
I will disappear from here !!
Task2: I will print the new content of file f2. My PID: 5502

--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: 5503

Parent PID:: 5500

tanishqmalu@DESKTOP-5JCAMRU:/mnt/d/Tanishq/3rd year/6th sem/os_lab/lab-4$
```

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