

HOME ASSIGNMENT 1

PART 1

ID:12341550

Q1-SOLUTION

OUTPUT:

```
paritosh@paritosh-ASUS-TUF-Gaming-F15-FX507ZE-FX577ZE: ~/Desktop/IIT_BHILAI/sem5/CSL301_OS/assignments/1_assignment/part1$ gcc q1.c -o iout
paritosh@paritosh-ASUS-TUF-Gaming-F15-FX507ZE-FX577ZE: ~/Desktop/IIT_BHILAI/sem5/CSL301_OS/assignments/1_assignment/part1$ ./iout
Initial value of x: 10 (PID: 97168)
--- Child process starting (PID: 97169) ---
Child process: x = 20
--- Child process finishing ---
--- Parent process starting (PID: 97168) ---
Parent process: x = 15
--- Parent process finishing ---
paritosh@paritosh-ASUS-TUF-Gaming-F15-FX507ZE-FX577ZE: ~/Desktop/IIT_BHILAI/sem5/CSL301_OS/assignments/1_assignment/part1$
```

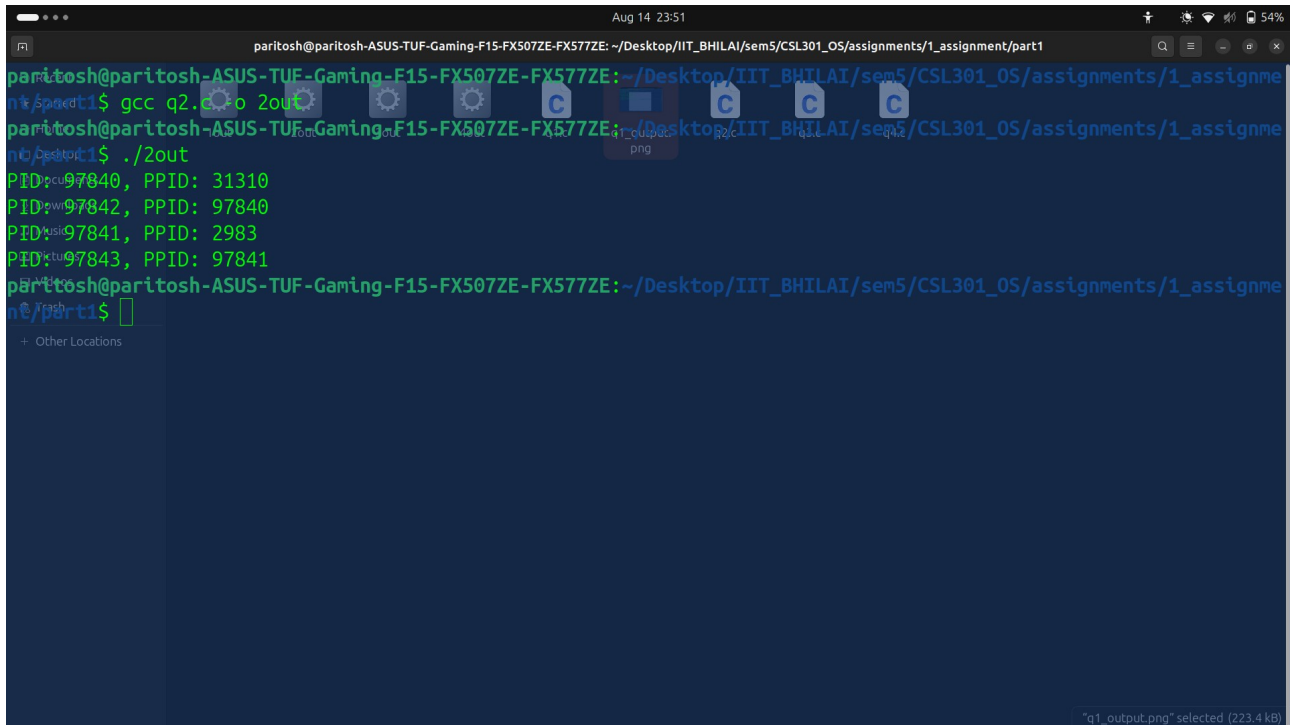
EXPLANATION :

When a process calls `fork()`, the operating system creates a new child process that is a copy of the parent's entire memory space, including global variables, stack, and heap. After the fork, the parent and child run independently. They have separate copies of all variables changing a variable in one process changes only that process's memory, not the other's.

That's why as shown above in child process global value copied value got increment of 10 which didn't affect the original global variable because in parent process there is increment of 5 only.

Q2-SOLUTION

OUTPUT :



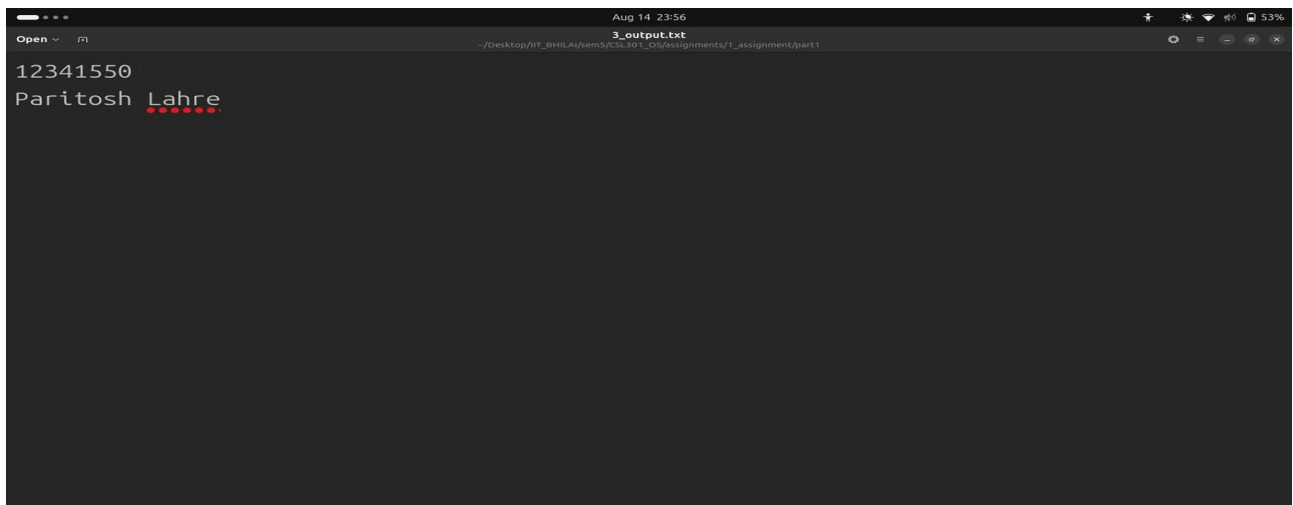
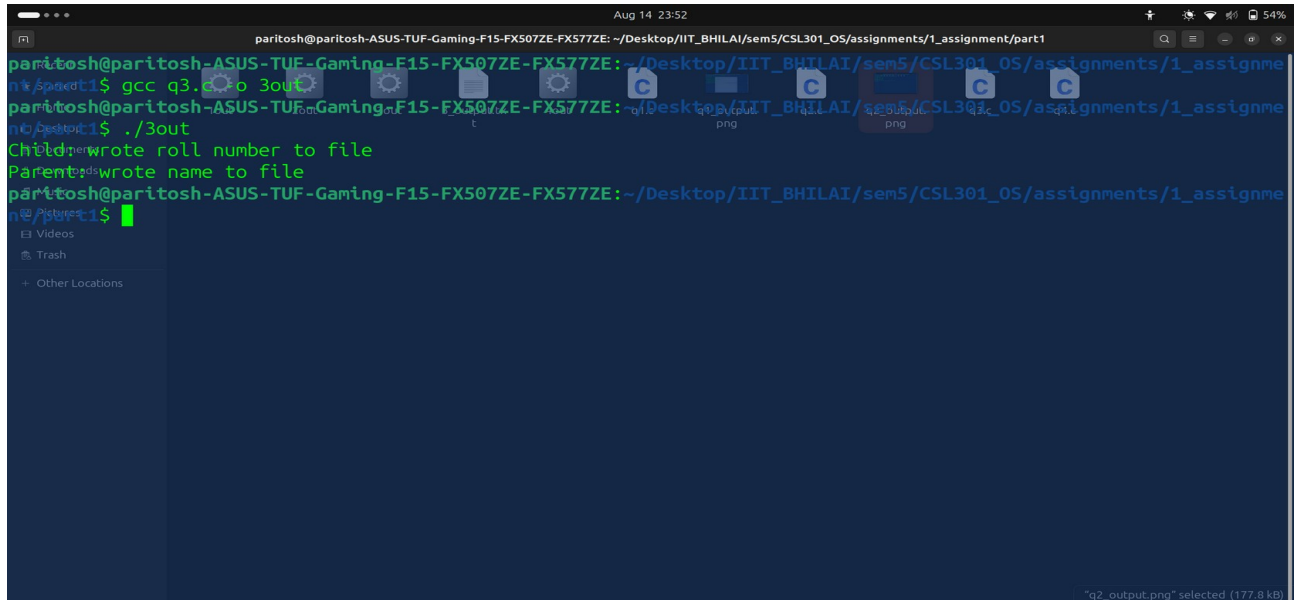
```
paritosh@paritosh-ASUS-TUF-Gaming-F15-FX507ZE-FX577ZE: ~/Desktop/IIT_BHILAI/sem5/CSL301_OS/assignments/1_assignment/part1
paritosh@paritosh-ASUS-TUF-Gaming-F15-FX507ZE-FX577ZE:~/Desktop/IIT_BHILAI/sem5/CSL301_OS/assignments/1_assignment/part1$ gcc q2.c -o 2out
paritosh@paritosh-ASUS-TUF-Gaming-F15-FX507ZE-FX577ZE:~/Desktop/IIT_BHILAI/sem5/CSL301_OS/assignments/1_assignment/part1$ ./2out
PID: 97840, PPID: 31310
PID: 97842, PPID: 97840
PID: 97841, PPID: 2983
PID: 97843, PPID: 97841
paritosh@paritosh-ASUS-TUF-Gaming-F15-FX507ZE-FX577ZE:~/Desktop/IIT_BHILAI/sem5/CSL301_OS/assignments/1_assignment/part1$
```

EXPLANATION :

In my observation there were total 4 processes created. First fork() splits into 2 processes : one parent , one child. Second fork() splits 2 more processes by those previous two processes.

Process tree which I could visualise is parent called 1st child in first fork() call (two processes) and then 2nd child by parent and 3rd child or grand child by 1st child in second fork() call.

OUTPUT:

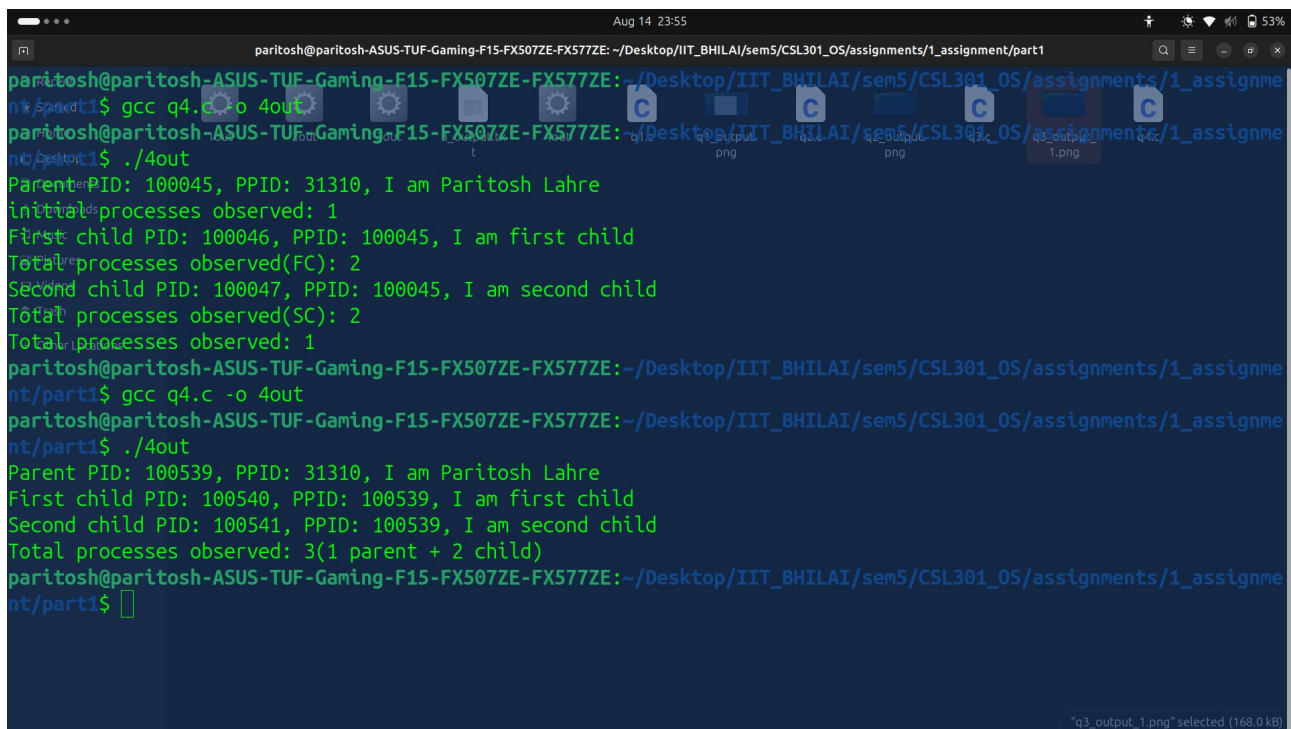


EXPLANATION :

When `fork()` is called child inherits copies of parent's file descriptor(let FD). These FD in the parent and child refers to the same open file description means they share the same file offset and status flags(like append).As in my program `append` is called before `fork()` both process append their messages without overwriting which ensures the right order of writing. Also the order depends on which process is called first as I used `wait` so child is called first so my roll number is written first in output text file then my name from parent process.

Q4-SOLUTION

OUTPUT:



```
paritosh@paritosh-ASUS-TUF-Gaming-F15-FX507ZE-FX577ZE: ~/Desktop/IIT_BHILAI/sem5/CSL301_OS/assignments/1_assignment/part1
paritosh@paritosh-ASUS-TUF-Gaming-F15-FX507ZE-FX577ZE: ~/Desktop/IIT_BHILAI/sem5/CSL301_OS/assignments/1_assignment/part1$ gcc q4.c -o 4out
paritosh@paritosh-ASUS-TUF-Gaming-F15-FX507ZE-FX577ZE: ~/Desktop/IIT_BHILAI/sem5/CSL301_OS/assignments/1_assignment/part1$ ./4out
Parent PID: 100045, PPID: 31310, I am Paritosh Lahre
initial processes observed: 1
First child PID: 100046, PPID: 100045, I am first child
Total processes observed(FC): 2
Second child PID: 100047, PPID: 100045, I am second child
Total processes observed(SC): 2
Total processes observed: 1
paritosh@paritosh-ASUS-TUF-Gaming-F15-FX507ZE-FX577ZE: ~/Desktop/IIT_BHILAI/sem5/CSL301_OS/assignments/1_assignment/part1$ gcc q4.c -o 4out
paritosh@paritosh-ASUS-TUF-Gaming-F15-FX507ZE-FX577ZE: ~/Desktop/IIT_BHILAI/sem5/CSL301_OS/assignments/1_assignment/part1$ ./4out
Parent PID: 100539, PPID: 31310, I am Paritosh Lahre
First child PID: 100540, PPID: 100539, I am first child
Second child PID: 100541, PPID: 100539, I am second child
Total processes observed: 3(1 parent + 2 child)
paritosh@paritosh-ASUS-TUF-Gaming-F15-FX507ZE-FX577ZE: ~/Desktop/IIT_BHILAI/sem5/CSL301_OS/assignments/1_assignment/part1$
```

EXPLANATION:

So total processes observed is 3 (1 parent and 2 child). Parent starts first then created the first child by first fork() waited for completion and created second child by second fork and executed its role. Now to observe number of processes I first created a counter which we can see in above image when I first ran the script, counter increment by 1 when parent starts then in first fork() created copy of it and increment it to 2 but since it's a copy of parent counter, after finishing counter again becomes one as same as parent. Then when second fork() called same thing happened as of first fork() increment of 2 then when process is completed parent's counter is printed as 1 which indicates all its child process is finished.

Both child are siblings as each child created after when its role is finished.