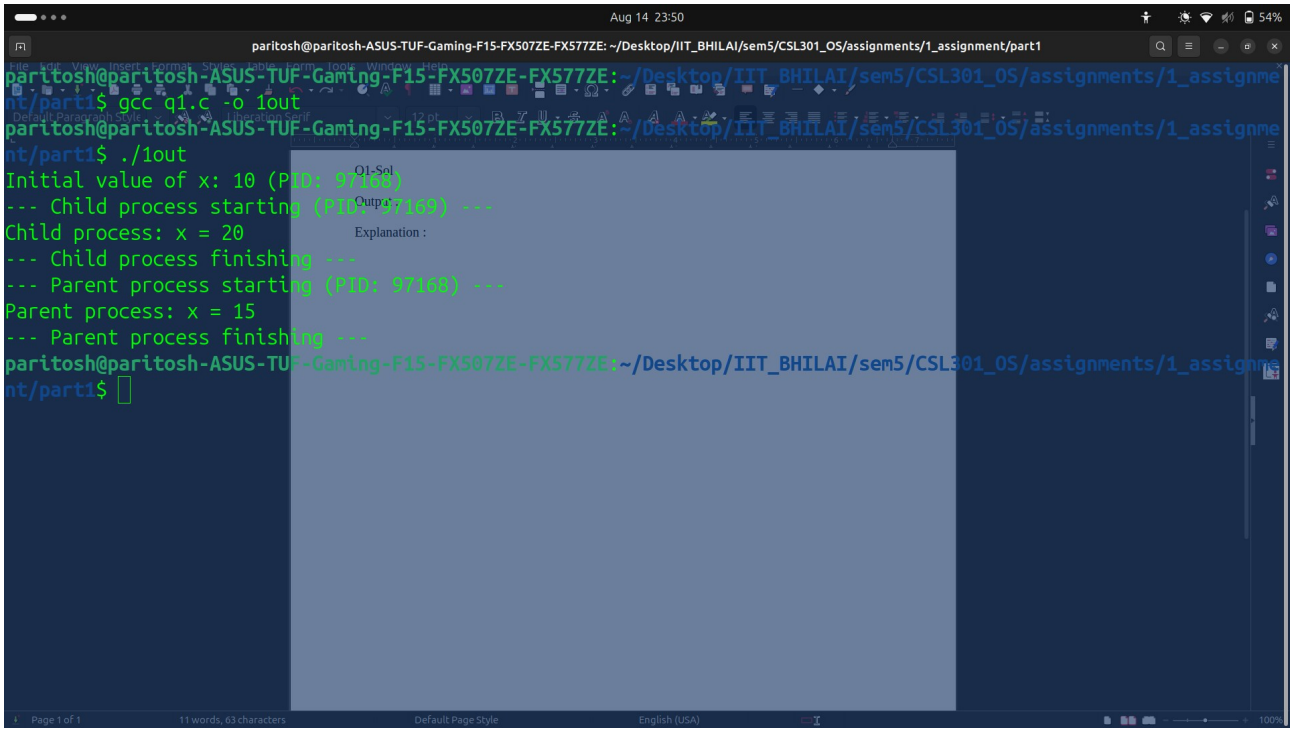
HOME ASSIGNMENT 1
PART 1
ID:12341550

Q1-SOLUTION

OUTPUT:



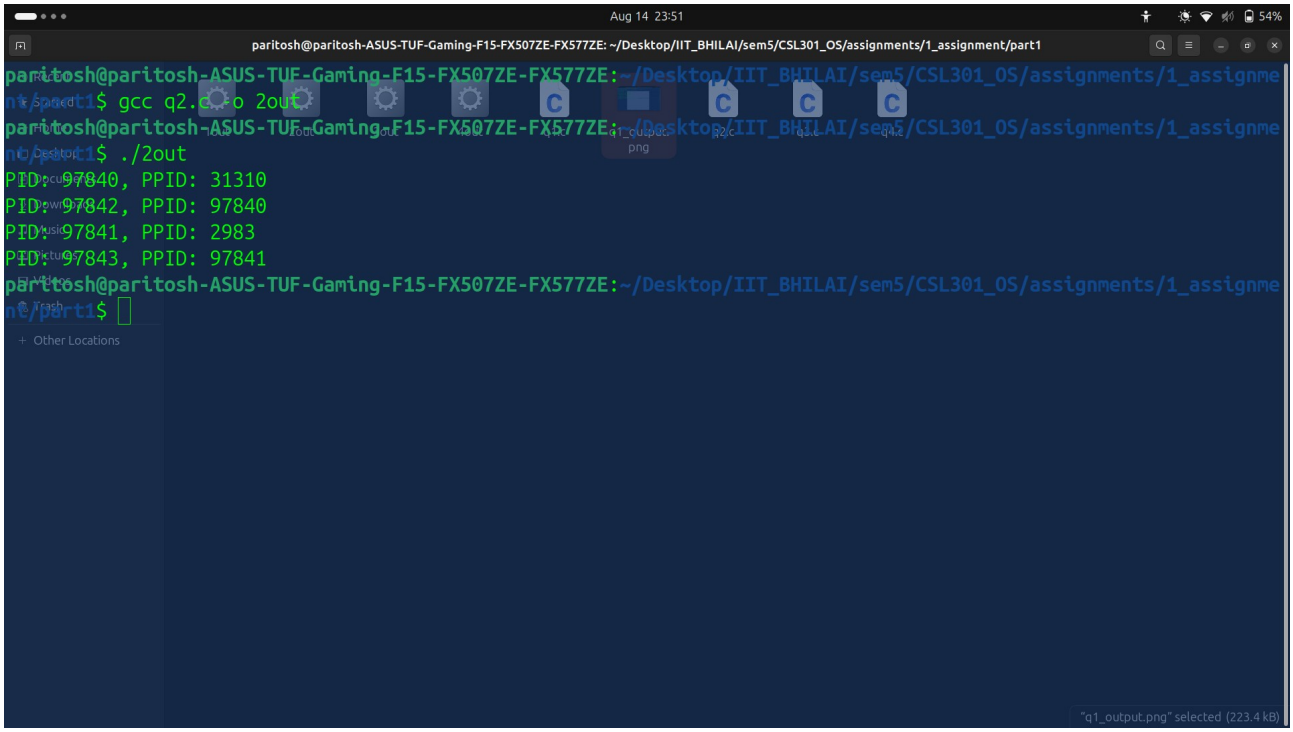
EXPLANATION :

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 :

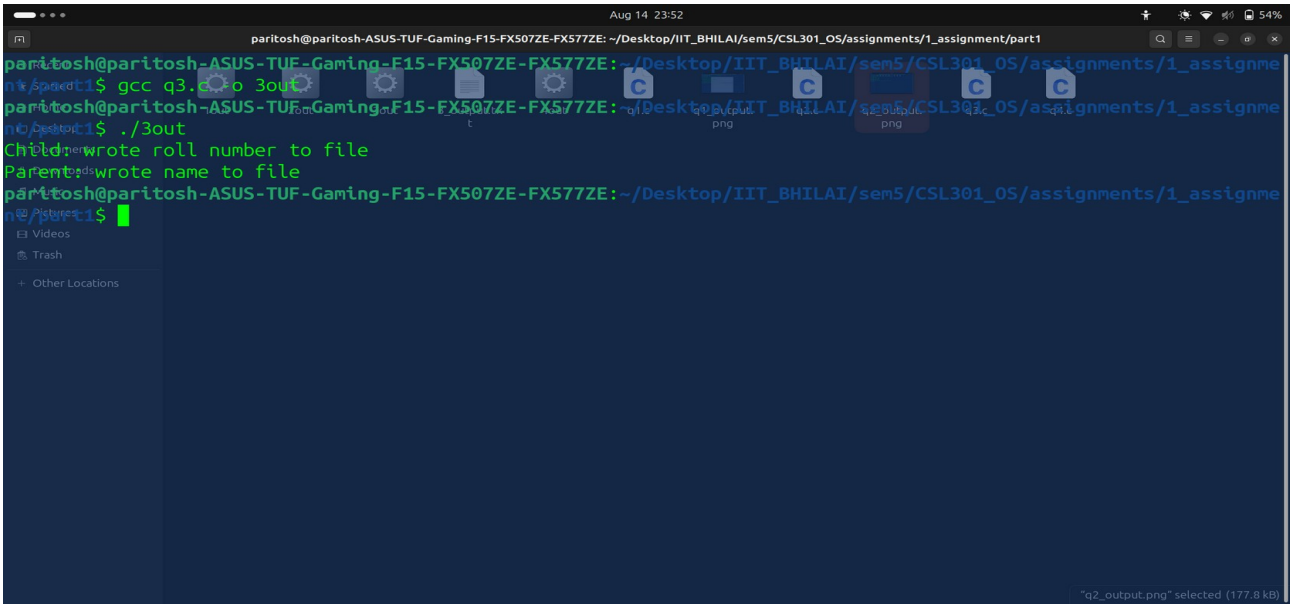


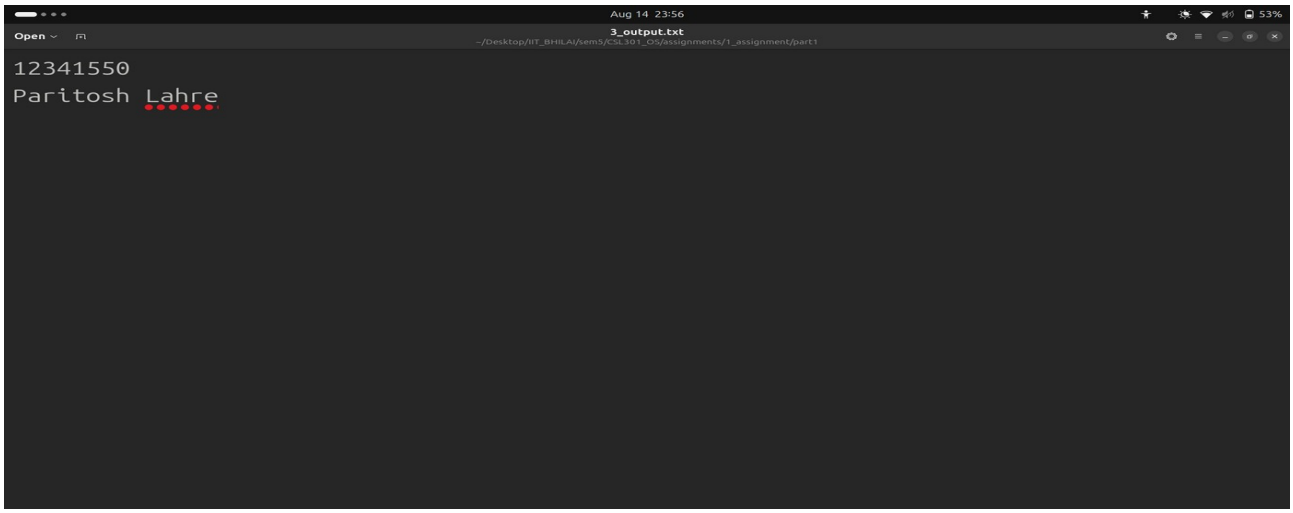
EXPLANATION :

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.

Q3-SOLUTION

OUTPUT:



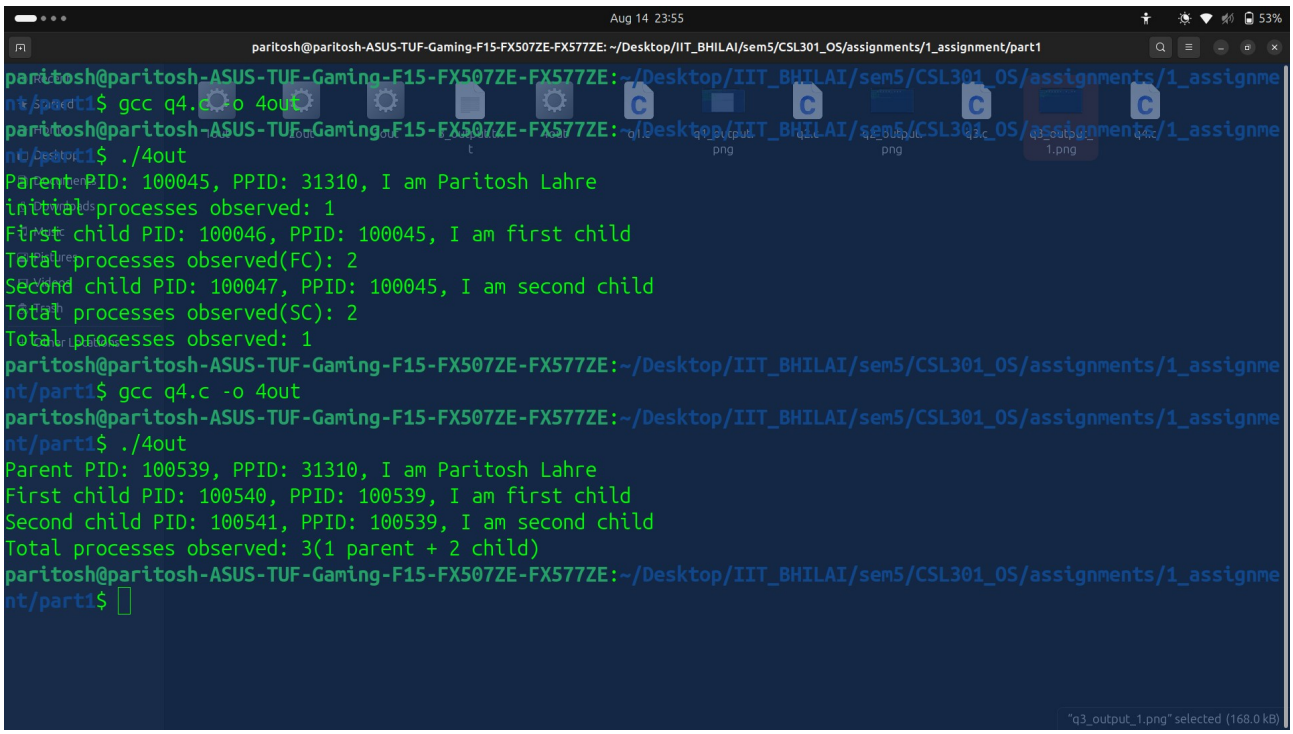


EXPLANATION :

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 fle 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 overwritting 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:



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 is role. Now to observe number of processes I first created a counter which we can saw in above image when I first ran the script , counter increment by 1 when parent start then in first fork() created copy of it and increment it to 2 but since its a copy of parent counter , after finishing counter again become 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 .