

Question No # 4

This piece of code creates $2^n - 1$ processes apart from its own (Main) process, while n is the number of times the `fork()` function is called, so we can say 7 processes are created by one primary process hence 8 processes in total.

A process is an independent running state of program having its own PCB, memory space and a copy of code. A variable is defined in the parent process will be inherited by every other process created by it, but changes by the child process will not be up-inherited by the parent. Here every child will increment the value of variable `a` independently but by copy method.

A simple hierarchy of the process creation has been visually presented in Image 1.

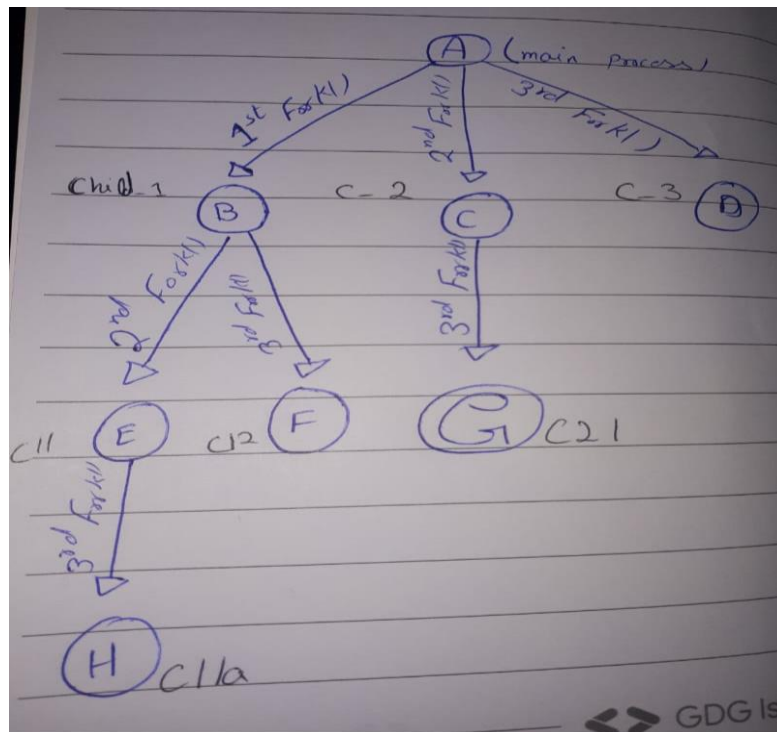


Image 1

Initially variable `a` possesses value 5, which is then inherited by its 1st child named `B`, now parent will run increment while process `B` will also increment the value, Now both of the processes possess value of `a = 6`, after that 2nd fork is called which will further create 2 children, one for `B` named `E` and other for `A` (Main process) named `C`. Its time for another increment which will run in all 4 processes (`A, B, C, E`) after that 3rd fork will be called and it will create 4 more processes (`F, G, H, D`), followed by another increment and now `a` will have value = 7 in all of its processes,

At the end every process, child and parent will have value = 7, but none value from the child process is worthy for parent as there is no `exit` & `wait` statement used.

Any process from these processes will become orphan process if its parent ends execution before the child has ended its execution and taken by `Init` process, and the short time of span during which it is parent less, it will be called `Zombie process`.