Tushar Mohan
2019393

Documentation

The question 1 is divided into 2 codes, one which includes forking to do the required task and one which includes making 2 threads to do the same task.

When we fork a process, a child process is created, using fork(). The parent process and the child process, both run concurrently on the processor in this case. The parent process is made to wait while the child process is in action, so as to prevent creation of orphan process (from the child process with no parent at the time of its termination). When a process is forked, the child process is given a different memory space with all the necessary resources allocated to it. This way, both the child as well as the parent processes have their separate memory spaces hence any alteration of a variable by either of the 2 doesn't have any effect in the variable the other.

When threads are formed, they too run concurrently but in the same memory space. Hence, during the output, there is a difference from the forked code output, as we can see that the value of "a" is majorly different by plus or minus 1 depending on which thread is being executed at that very point with a few exceptions arising due to the random scheduling policy of the threads executed in the system.