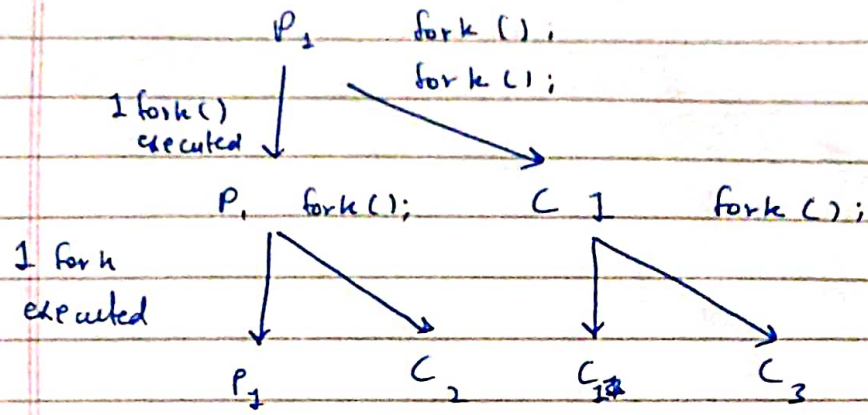
Shubham Singh
2022300118
SE Comps B

Exp 3 report : System calls and processes

Illustration of multiple consecutive fork calls:



n consecutive fork calls result in 2^n copies of the process (including the parent)

Learnings and experiences: In this experiment we learnt about the fork() and wait() system calls. The fork() system call is used to create a copy of the process that would run concurrently with the original using the same registers and program counter i.e. it would not run on another core, but ~~was~~ the processes would ~~switch~~ exchange control of the processor while one was waiting. The wait() call makes the parent process wait till all of its children have terminated. This can be used ~~to~~ with other calls to make the parent act upon the exit messages of the children. It can also be used to ~~prevent~~ send orphan processes (when init calls it). We also learnt about zombie and orphan processes. There are types of processes based on the ~~order~~ state of the parent process. Orphan processes are those whose parent terminates first, and they are allocated another parent (the init process), while zombie functions are those which end before their parent but their PIDs are still kept so that the parent could later read their exit status.