

Report for OS Assignment 1:

Question 1:

- a) The given question requires us to establish a parent-child relationship between two processes. We start with creating a child process using the `fork()` command (forking a child process) and assigning its value to the pid of a child process. After creating a new child process, both processes will execute the next instruction following the `fork()` system call. We use if-else statements to control program flow. If the child's PID (consisting of the return value of `fork()`) is less than 0, then an error is raised. Initially, the else part of the program will run, i.e., the Parent process, thus printing statement A (We used `getppid()` to get pid of the parent process). Then `wait(NULL)` command is used to stop the parent process to run the child process. If `child_pid` is equal to zero, `printf()` is executed in the child process but not in the parent process, thus printing the statements C and D. After completion of the child process, the parent process resumes printing statement D.
- b) In this part of the question, we first create the `factorial()` and `Fibonacci ()` functions. This part requires us to run the `vfork()` command, which is similar to `fork()` except for program flow, as `fork()` runs both the parent and child processes simultaneously, whereas `vfork()` suspends the execution of the parent process until the completion of the child process. Thus, first `factorial(4)` is executed and printed, and then `Fibonacci(16)` is printed.
- c) (BONUS) In this part, we must run the child process after the parent process. We have used `fork()` to establish the child process and used the `wait(NULL)` command to pause the child process such that the parent process completes its execution, i.e., `Fibonacci(16)`, and then the child process is executed (`factorial(4)`).