Assignment - 04
CS341: Operating System Lab

General Instruction

- Assignments should be completed and evaluated in the Lab session.
- Markings will be based on the correctness and soundness of the outputs. Marks will be deducted in case of plagiarism.
- Proper indentation & appropriate comments (if necessary) are mandatory in the code.

1. Design a C program named complexProcessComm.c that showcases advanced process management and inter-process communication (IPC) using both pipes and shared memory. In this program, the parent process will fork three child processes to perform sequential tasks. The first child process generates a list of 10 random integers and sends this list through a pipe to the second child process. The second child receives the list, calculates its average, and sends the average value through another pipe to the third child process. The third child reads the average, updates a global counter in shared memory, and prints both the average value and the updated counter. The parent process should print its own process ID (PID) and the PIDs of the three child processes, then terminate, allowing the child processes to complete their tasks. Ensure that each child process performs its task in the correct sequence, demonstrating effective use of both pipes for communication and shared memory for synchronization and data sharing. The final output of the third child process should show the calculated average and the incremented global counter value.
2. Write a C program named complexZombie.c that demonstrates process management, inter-process communication (IPC), and data structures. The program should fork two child processes: the first child generates a list of 10 random integers, sends this list through a pipe to the second child, which sorts the list using an efficient algorithm (like quicksort), and measures the sorting time. The parent process prints its PID and the PIDs of the two child processes, then sleeps for 1 minute, allowing the second child to become a zombie. After sleeping, the parent should use the ps command to check and display the zombie state of the second child process. Finally, the parent should wait for the second child to exit and clean up the zombie. Concepts involved include process creation (fork()), IPC with pipes, sorting algorithms, zombie processes, and process state checking using ps.