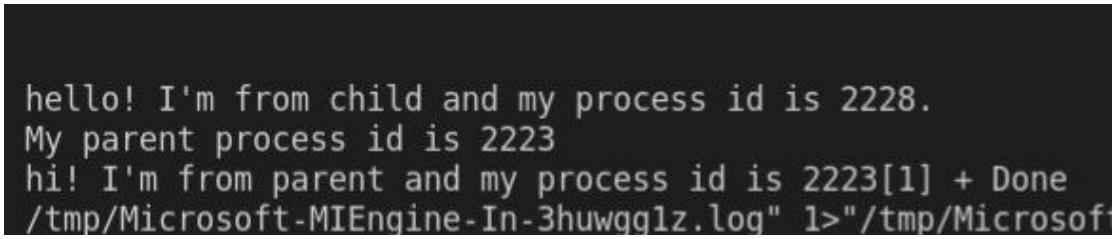
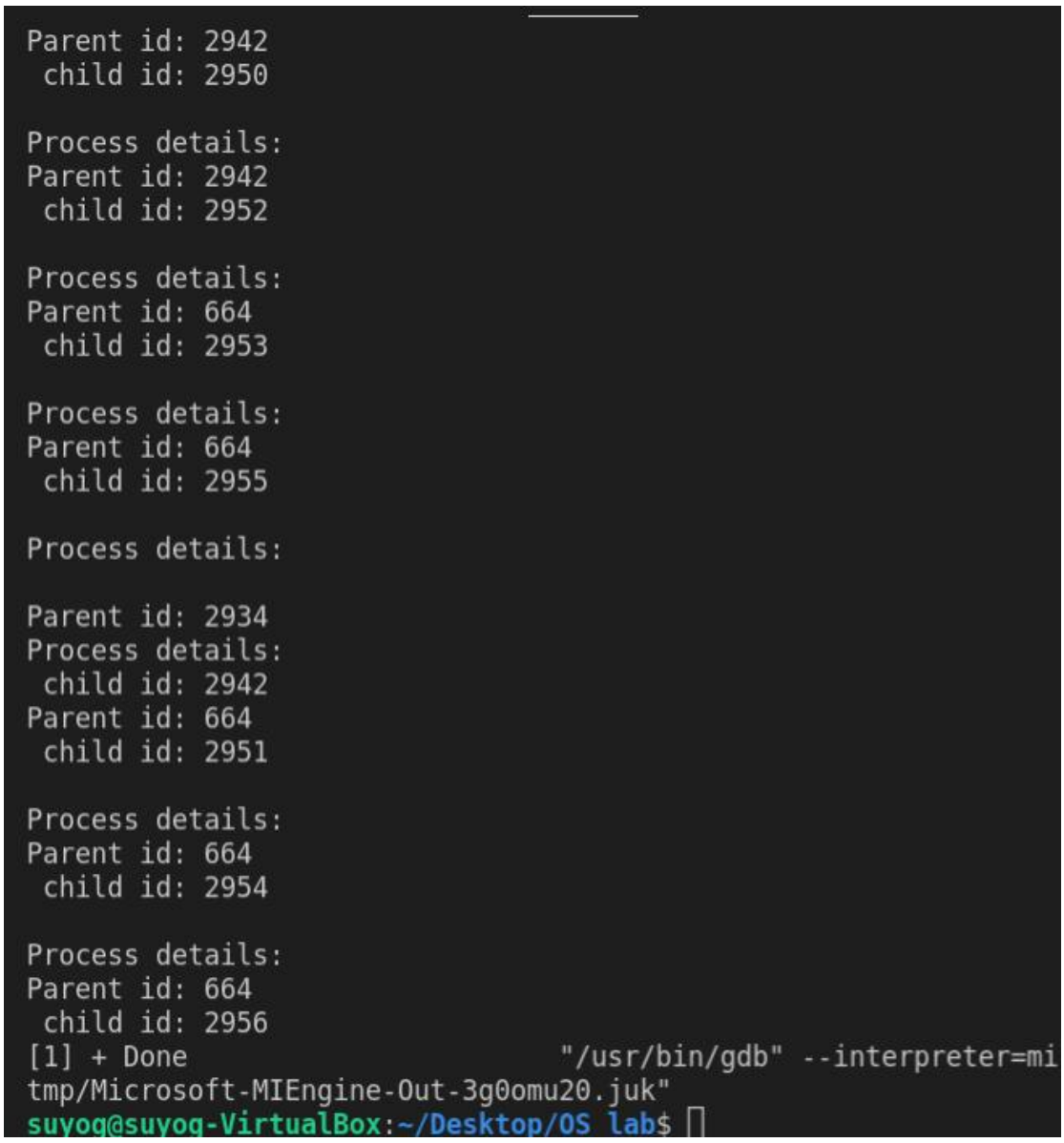
Results:

1. Task1



Task 1 a



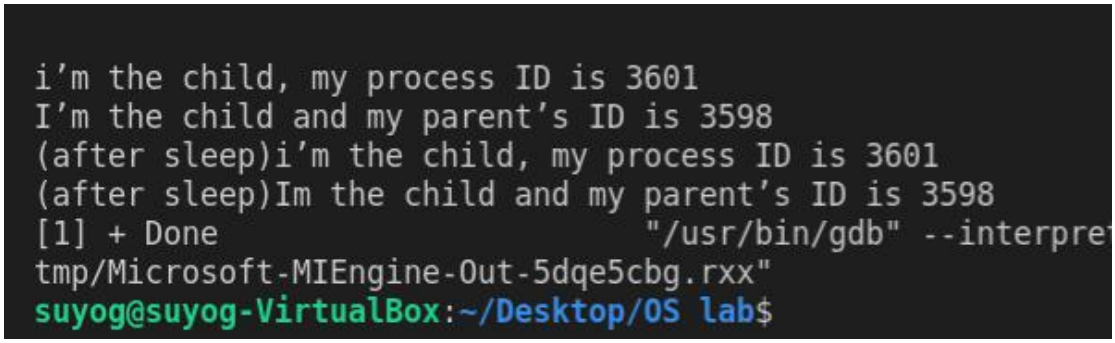
There are eight process in total out of which seven are child. Their process id along with their parent id is shown as above.

2.

In this program, the parent executes its program and the DEL2 holds the screen and again child executes and holds screen from DEL1.

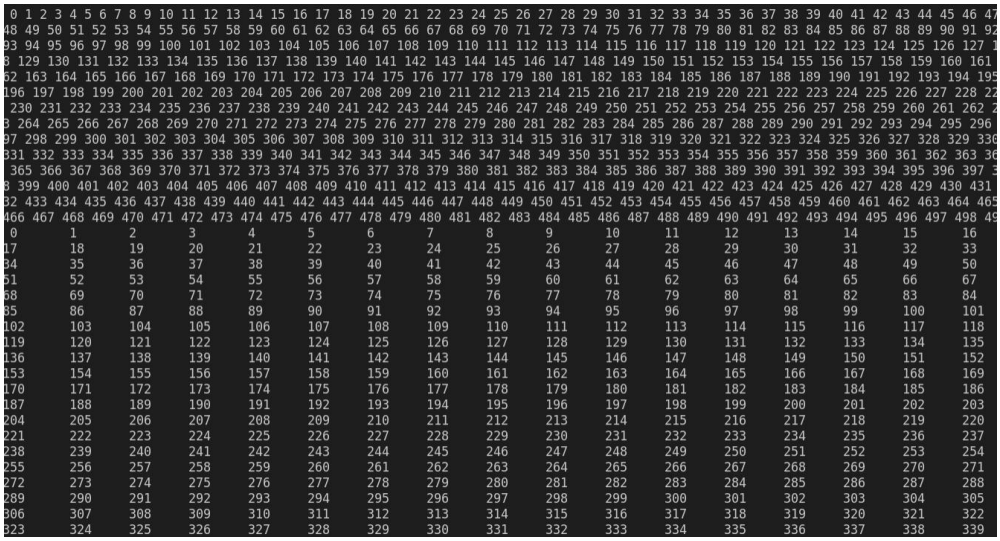
When we increase the delays DEL1 and DEL2 then we saw that the output of two process overlapped. This is due to non-uniform delays and both process are running at the same time.

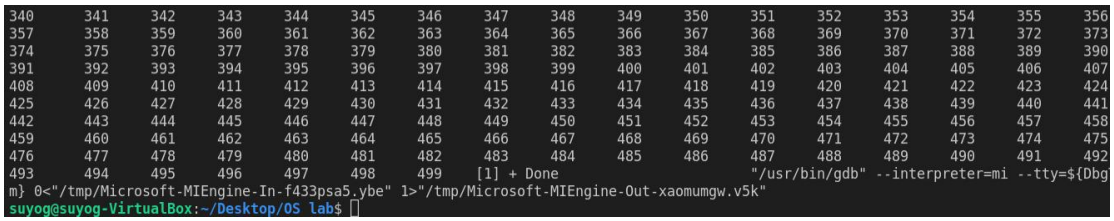
3.



Here, the execution of fork() was successful so the value of pid will be 0 and hence only the statements under if with condition pid=0 is executed.
Adding the sleep statement delayed the output by some time . The child process is an orphan process and a new parent is given to it by the system.

4.





In this program, first the parent executed and then the execution of the child took place.

After edit

When the wait(0) statement is added, then the parent waits for its child to complete its execution and the executes itself. Only the parent can wait for the child and not the other way around.

Discussion:

In this lab session, we developed the concept of parent and child process. We used the fork statement to observe the behavior of parent and child process under different circumstances. We also observed the process id's using getpid() and getppid() statement. Also, the behavior of the processes under various delay conditions were also observed. Along with these, we also used the wait() statement to make the parent process wait till the completion of child process.

Conclusion:

In concluding, we became familiar with various process concepts and manipulation metrics for different processes. After this lab session we are able to understand the concept of parent and child process as well as the importance of proper delays for these processes.