**Operating Systems Lab Assignment 1-**

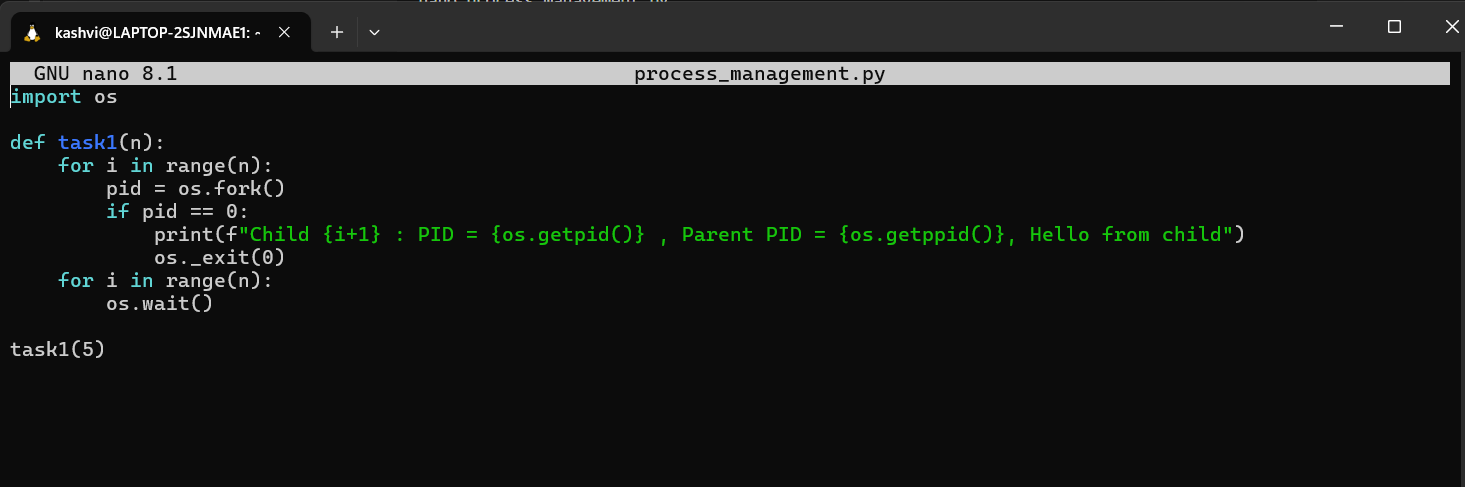
**Task1- Process Creation Utility**

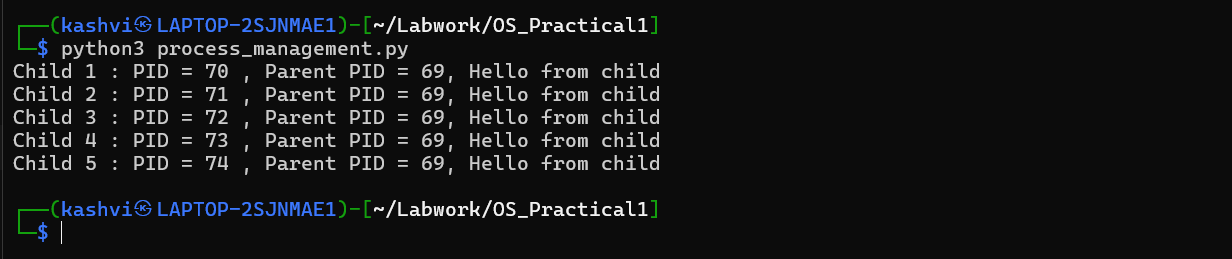
Write a Python program that creates N child processes using os.fork(). Each child prints:  
- Its PID  
- Its Parent PID  
- A custom message  
The parent should wait for all children using os.wait().

**Sol.-**

A screenshot of a computer

AI-generated content may be incorrect.

****

****

**Task 2- Command Execution Using exec()**

Modify Task 1 so that each child process executes a Linux command (ls, date, ps, etc.) using os.execvp() or subprocess.run().

**Sol.-**

**A computer screen with green text

AI-generated content may be incorrect.**

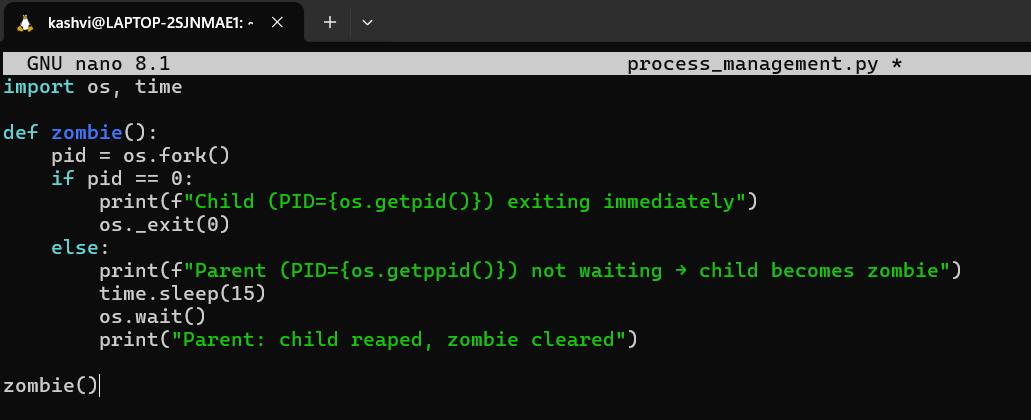
**A screenshot of a computer screen

AI-generated content may be incorrect.**

**Task 3 - Zombie & Orphan Processes**

Zombie: Fork a child and skip wait() in the parent.  
Orphan: Parent exits before the child finishes.  
Use ps -el | grep defunct to identify zombies.

**Sol.-**



**A screen shot of a computer code

AI-generated content may be incorrect.**

**A computer screen with green text

AI-generated content may be incorrect.**

**A screenshot of a computer code

AI-generated content may be incorrect.**

**Task 4 - Inspecting Process Info from /proc**

Take a PID as input. Read and print:  
- Process name, state, memory usage from /proc/[pid]/status  
- Executable path from /proc/[pid]/exe  
- Open file descriptors from /proc/[pid]/fd

**Sol.- -A computer screen with green and white text

AI-generated content may be incorrect.**

**A computer screen shot of a computer code

AI-generated content may be incorrect.**

**Task 5 - Process Prioritization**

Create multiple CPU-intensive child processes. Assign different nice() values. Observe and log execution order to show scheduler impact.

**Sol.-**

**A computer screen shot of a program

AI-generated content may be incorrect.**

**A computer screen with white text

AI-generated content may be incorrect.**