

Operating Systems Lab Assignment - 2 Report

Name: Mayank Jha

Roll no. 2301730165

Course Code: ENCS351

Program : B.Tech CSE(Ai/MI)

Experiment: System Startup, Process Creation, and Termination

Simulation in Python

Objective:

To simulate core Operating System concepts such as system booting, process creation, execution management, and graceful shutdown using Python. The goal is to understand how an OS spawns multiple concurrent processes and logs their lifecycle using multiprocessing and logging modules.

Tools Used:

Python 3.x

Linux Environment (Ubuntu/WSL via Terminal)

multiprocessing module

logging module

time module

Tasks Performed:

- Task 1(Logging Setup): Initialized the logging configuration to capture timestamped messages and process names into a file named process_log.txt.
- Task 2 (Task Definition): Defined a function system_process() that mimics a system task by logging its start time, sleeping for a specific duration, and logging its end time.

- Task 3 (Process Creation): Created multiple child processes (e.g., Boot-Loader, Network-Service) using multiprocessing.Process and started them concurrently to simulate parallel system initialization.
- Task 4 (Synchronization & Shutdown): Implemented proper termination using the join() method to ensure the main system waits for all child processes to complete before printing "System Shutdown".

Outputs:

- Successfully simulated a system startup sequence.
- Generated a persistent log file (process_log.txt) that verified concurrent execution.
- Verified that the main process waits for child processes to finish before exiting.



System Starting...
System Shutdown.

A screenshot of a terminal window with a black background and white text. It displays two lines of text: "System Starting..." on the first line and "System Shutdown." on the second line, centered in the window.

```
☰ process_log.txt
1  2025-11-18 16:13:34,983 - Process-1 - Process-1 started
2  2025-11-18 16:13:34,984 - Process-2 - Process-2 started
3  2025-11-18 16:13:36,474 - Process-1 - Process-1 ended
4  2025-11-18 16:13:36,474 - Process-2 - Process-2 ended
5  2025-11-18 16:17:07,178 - Process-1 - Process-1 started
6  2025-11-18 16:17:07,179 - Process-2 - Process-2 started
7  2025-11-18 16:17:02,719 - Process-1 - Process-1 ended
8  2025-11-18 16:17:02,755 - Process-2 - Process-2 ended
9  2025-11-18 16:17:31,709 - Process-1 - Process-1 started
10 2025-11-18 16:17:31,710 - Process-2 - Process-2 started
11 2025-11-18 16:17:33,742 - Process-2 - Process-2 ended
12 2025-11-18 16:17:33,786 - Process-1 - Process-1 ended
13 2025-11-18 16:17:40,424 - Process-1 - Process-1 started
14 2025-11-18 16:17:40,425 - Process-2 - Process-2 started
15 2025-11-18 16:17:41,941 - Process-1 - Process-1 ended
16 2025-11-18 16:17:41,980 - Process-2 - Process-2 ended
17 2025-11-18 16:22:08,535 - Process-2 - Process-2 started
18 2025-11-18 16:22:08,535 - Process-1 - Process-1 started
19 2025-11-18 16:22:10,538 - Process-1 - Process-1 ended
20 2025-11-18 16:22:10,538 - Process-2 - Process-2 ended
21
```

Learning Outcomes:

- Concurrency: Understood how multiprocessing allows multiple tasks to run simultaneously, simulating real OS background services.
- Synchronization: Learned the importance of join() to prevent the main system from shutting down while critical processes are still running.
- System Logging: Gained experience in implementing system logs to track the state and lifecycle of processes for debugging and auditing.
- Process Lifecycle: Visualized the start, execution, and termination phases of system processes.