OS Lab Assignment 2 – Process Management using Python Multiprocessing

1. Aim of the Experiment

To simulate process creation, execution, and synchronization using Python's multiprocessing module, and to log system activities including process lifecycle events, system startup, and shutdown.

2. Objectives

- Understand how an OS handles multiple processes.

- Simulate process behavior using Python.

- Learn process creation, start, join, and termination.

- Use logging to track process execution.

3. Tools & Technologies

- Python 3

- multiprocessing

- logging

- time module

4. Program Description

4.1 Logging Initialization

A log file (process_log.txt) is created to record events.

4.2 System Boot Simulation

simulate_system_startup() logs system start.

4.3 Process Creation

Three processes:

- Init-Process

- User-Process-1

- Background-Process

Each logs start and end, and sleeps for some duration.

4.4 Joining Processes

OS waits for termination of all processes using p.join()

4.5 Shutdown

Logs system shutdown sequence.

5. Output Example

Terminal:

```
System Starting...

System Shutdown.
```

Sample Log:

Process started, ended, system shutdown messages.

6. Conclusion

The experiment demonstrates OS-level concepts such as process creation, management, and logging using Python multiprocessing.

7. References

Python documentation, OS Concepts by Silberschatz.