ASSIGNMENT -2

Name: Monika Raghav

Roll. No.: 2301420010

Course: B.Tech(CSE) Data Science

Problem Statement:

Modern operating systems are responsible for initializing system components, creating processes, managing execution, and gracefully shutting down. This lab aims to simulate these core concepts using Python, helping students visualize how processes are handled at the OS level. The focus is on creating a simplified startup mechanism that spawns multiple processes and logs their lifecycle using the multiprocessing and logging modules. This hands-on simulation enhances conceptual clarity and promotes coding proficiency in scripting real-world OS behavior.

Task 1: Write a Python script to simulate a basic system startup sequence.

Code:

Output:

Task 2: Use the multiprocessing module to create at least two child processes that perform dummy tasks.

Code:

Output:

Task 3: Implement proper logging to track process start and end times.

Code:

Output:

```
┌──(monika㉿Monika)-[~]
└─$ python3 system_startup.py

System Starting ...
System Shutdown.
```

```
┌──(monika㉿Monika)-[~]
└─$ cat process_log.txt

2025-09-29 14:03:34,306 - Process-1 - Process-1 started
2025-09-29 14:03:34,308 - Process-2 - Process-2 started
2025-09-29 14:03:36,309 - Process-1 - Process-1 ended
2025-09-29 14:03:36,309 - Process-2 - Process-2 ended
```

Task 4: Generate a log file (process_log.txt) to reflect system-like behavior.

Code:

```
File  Actions  Edit  View  Help
 GNU nano 8.4                                    task4_logfile.py
# Task 4: Ensure proper termination and display log contents

import multiprocessing
import time
import logging

# Configure logging
logging.basicConfig(
    filename='process_log.txt',
    level=logging.INFO,
    format='%(asctime)s - %(processName)s - %(message)s'
)

def system_process(task_name):
    logging.info(f"{task_name} started")
    time.sleep(2)
    logging.info(f"{task_name} ended")

if __name__ == '__main__':
    print("System Starting ... ")

    # Create processes
    p1 = multiprocessing.Process(target=system_process, args=('Process-1',))
    p2 = multiprocessing.Process(target=system_process, args=('Process-2',))

    # Start processes
    p1.start()
    p2.start()

    # Wait for processes to complete (proper termination)
    p1.join()
    p2.join()
```

```
    print("System Shutdown.")

    # Show the log file content to verify
    with open('process_log.txt', 'r') as f:
        print("\nLog File Content:\n")
        print(f.read())

    filename='process_log.txt',
    level=logging.INFO,
    format='%(asctime)s - %(processName)s - %(message)s'
)

def system_process(task_name):
    logging.info(f"{task_name} started")
    time.sleep(2)
    logging.info(f"{task_name} ended")

if __name__ == '__main__':
    print("System Starting ... ")

    # Create processes
    p1 = multiprocessing.Process(target=system_process, args=('Process-1',))
    p2 = multiprocessing.Process(target=system_process, args=('Process-2',))

    # Start processes

    # Show the log file content to verify
    with open('process_log.txt', 'r') as f:
        print("\nLog File Content:\n")
        print(f.read())
```

Output:

```
┌──(monika㉿Monika)-[~]
└─$ python3 task4_logfile.py

System Starting ...
System Shutdown.

Log File Content:

2025-09-29 14:03:34,306 - Process-1 - Process-1 started
2025-09-29 14:03:34,308 - Process-2 - Process-2 started
2025-09-29 14:03:36,309 - Process-1 - Process-1 ended
2025-09-29 14:03:36,309 - Process-2 - Process-2 ended
2025-09-29 14:07:55,256 - Process-1 - Process-1 started
2025-09-29 14:07:55,258 - Process-2 - Process-2 started
2025-09-29 14:07:57,258 - Process-1 - Process-1 ended
2025-09-29 14:07:57,259 - Process-2 - Process-2 ended
2025-09-29 15:03:00,556 - Process-1 - Process-1 started
2025-09-29 15:03:00,561 - Process-2 - Process-2 started
2025-09-29 15:03:02,561 - Process-1 - Process-1 ended
2025-09-29 15:03:02,562 - Process-2 - Process-2 ended
```

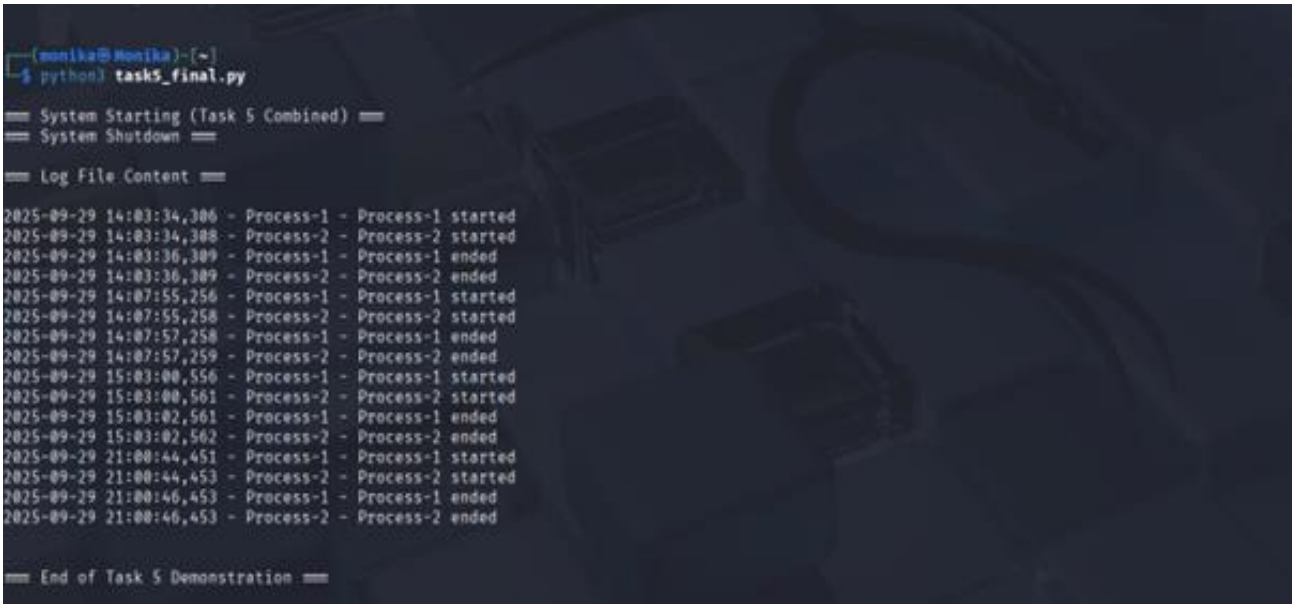
Task 5: Submit the Python script and log file along with a short report explaining your implementation.

Code:

```
File  Actions  Edit  View  Help
 GNU nano 8.4                                    task5_final.py *
# Task 5: Combined demonstration of all tasks

import multiprocessing
import time
import logging

# Configure logging
logging.basicConfig(
    filename='process_log.txt',
    level=logging.INFO,
    format='%(asctime)s - %(processName)s - %(message)s'
)

def system_process(task_name):
    """Simulate a dummy system process."""
    logging.info(f"{task_name} started")
    time.sleep(2)
    logging.info(f"{task_name} ended")

if __name__ == '__main__':
    print("=== System Starting (Task 5 Combined) ===")

    # Create and start processes
    p1 = multiprocessing.Process(target=system_process, args=('Process-1',))
    p2 = multiprocessing.Process(target=system_process, args=('Process-2',))
    p1.start()
    p2.start()

    # Ensure proper termination
    p1.join()
    p2.join()

    print("=== System Shutdown ===")
```

```
    # Display log contents
    with open('process_log.txt', 'r') as f:
        print("\n=== Log File Content ===\n")
        print(f.read())

    print("\n=== End of Task 5 Demonstration ===")
```

Output:


python3 task5_final.py
== System Starting (Task 5 Combined) ==
== System Shutdown ==
== Log File Content ==
2025-09-29 14:03:34,306 - Process-1 - Process-1 started
2025-09-29 14:03:34,308 - Process-2 - Process-2 started
2025-09-29 14:03:36,309 - Process-1 - Process-1 ended
2025-09-29 14:03:36,309 - Process-2 - Process-2 ended
2025-09-29 14:07:55,256 - Process-1 - Process-1 started
2025-09-29 14:07:55,258 - Process-2 - Process-2 started
2025-09-29 14:07:57,258 - Process-1 - Process-1 ended
2025-09-29 14:07:57,259 - Process-2 - Process-2 ended
2025-09-29 15:03:00,556 - Process-1 - Process-1 started
2025-09-29 15:03:00,561 - Process-2 - Process-2 started
2025-09-29 15:03:02,561 - Process-1 - Process-1 ended
2025-09-29 15:03:02,562 - Process-2 - Process-2 ended
2025-09-29 21:00:44,451 - Process-1 - Process-1 started
2025-09-29 21:00:44,453 - Process-2 - Process-2 started
2025-09-29 21:00:46,453 - Process-1 - Process-1 ended
2025-09-29 21:00:46,453 - Process-2 - Process-2 ended
== End of Task 5 Demonstration ==