

Task 1: Batch Processing Simulation (Python)

Write a Python script to execute multiple .py files sequentially, mimicking batch processing.

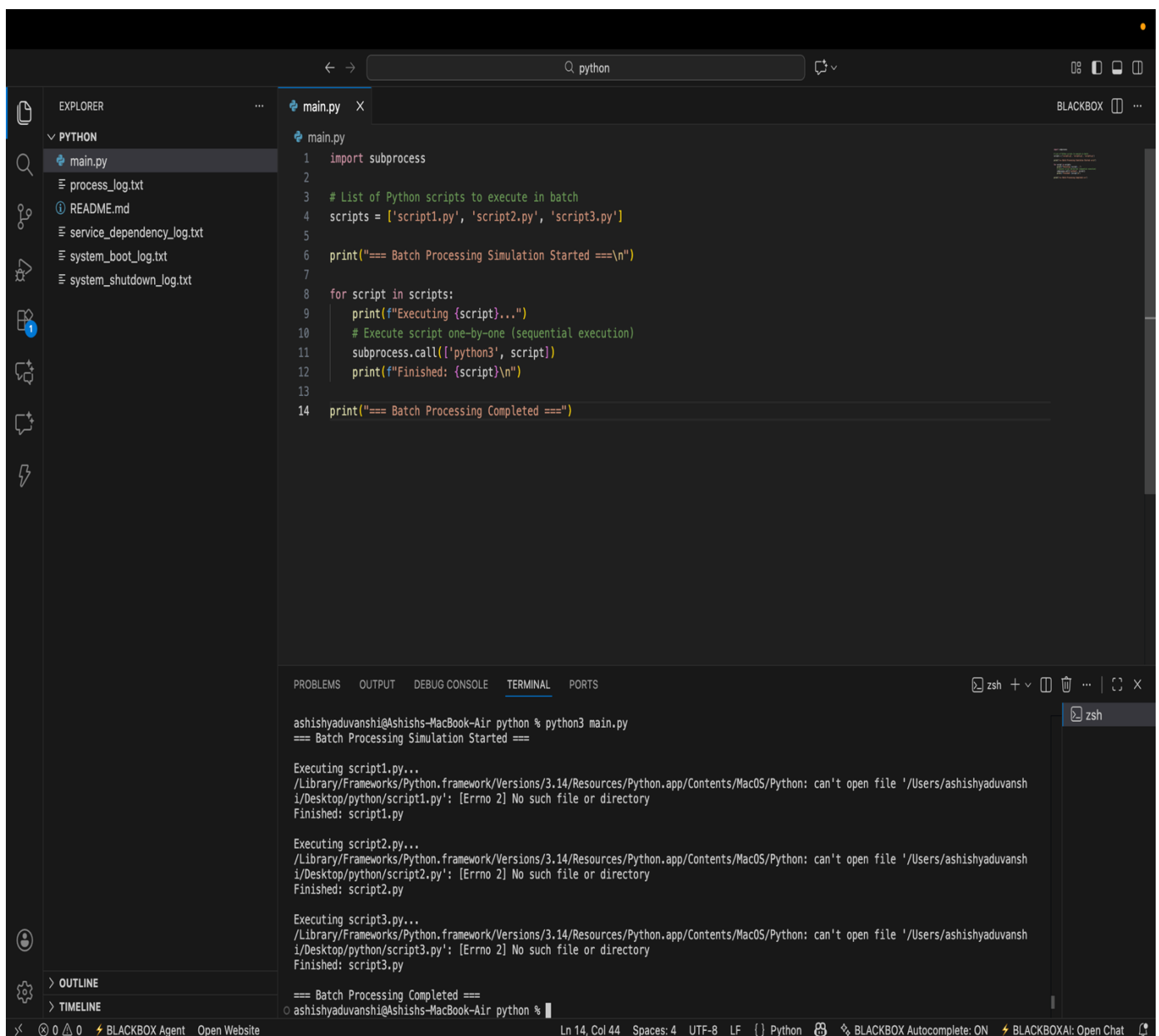
```
import subprocess
```

```
scripts = ['script1.py', 'script2.py', 'script3.py']
```

```
for script in scripts:
```

```
    print(f"Executing {script}...")
```

```
    subprocess.call(['python3', script])
```



The screenshot shows a code editor with a file explorer on the left, a main editor window, and a terminal at the bottom. The file explorer shows a project named 'PYTHON' with files: 'main.py', 'process_log.txt', 'README.md', 'service_dependency_log.txt', 'system_boot_log.txt', and 'system_shutdown_log.txt'. The main editor window shows the Python script 'main.py' with the following code:

```
1 import subprocess
2
3 # List of Python scripts to execute in batch
4 scripts = ['script1.py', 'script2.py', 'script3.py']
5
6 print("=== Batch Processing Simulation Started ===\n")
7
8 for script in scripts:
9     print(f"Executing {script}...")
10    # Execute script one-by-one (sequential execution)
11    subprocess.call(['python3', script])
12    print(f"Finished: {script}\n")
13
14 print("=== Batch Processing Completed ===")
```

The terminal window shows the output of running the script:

```
ashishyaduvanshi@Ashishs-MacBook-Air python % python3 main.py
=== Batch Processing Simulation Started ===

Executing script1.py...
/Library/Frameworks/Python.framework/Versions/3.14/Resources/Python.app/Contents/MacOS/Python: can't open file '/Users/ashishyaduvanshi/Desktop/python/script1.py': [Errno 2] No such file or directory
Finished: script1.py

Executing script2.py...
/Library/Frameworks/Python.framework/Versions/3.14/Resources/Python.app/Contents/MacOS/Python: can't open file '/Users/ashishyaduvanshi/Desktop/python/script2.py': [Errno 2] No such file or directory
Finished: script2.py

Executing script3.py...
/Library/Frameworks/Python.framework/Versions/3.14/Resources/Python.app/Contents/MacOS/Python: can't open file '/Users/ashishyaduvanshi/Desktop/python/script3.py': [Errno 2] No such file or directory
Finished: script3.py

=== Batch Processing Completed ===
ashishyaduvanshi@Ashishs-MacBook-Air python %
```

Task 2: System Startup and Logging

Simulate system startup using Python by creating multiple processes and logging their start and end into a log file.

```
import multiprocessing

import logging

import time

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

def process_task(name):
    logging.info(f'{name} started')

    time.sleep(2)

    logging.info(f'{name} terminated')

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

    p1 = multiprocessing.Process(target=process_task, args=("Process-1",))
    p2 = multiprocessing.Process(target=process_task, args=("Process-2",))

    p1.start()
    p2.start()

    p1.join()
    p2.join()

    print("System Shutdown.")
```

python

python

EXPLORER

PYTHON

main.py

process_log.txt

README.md

service_dependency_log.txt

system_boot_log.txt

system_log.txt

system_shutdown_log.txt

main.py

6 logging.basicConfig(
7 filename='system_log.txt',
8 level=logging.INFO,
9 format='%(asctime)s - %(processName)s - %(message)s'
10)
11
12 # Dummy task for child processes
13 def process_task(name):
14 logging.info(f"{name} started")
15 print(f"{name} started...")
16 time.sleep(2) # simulate system work
17 logging.info(f"{name} terminated")
18 print(f"{name} terminated.")
19
20 if __name__ == "__main__":
21 print("System Booting...\n")
22 logging.info("System Boot Initiated")
23
24 # Create child processes
25 p1 = multiprocessing.Process(target=process_task, args=("Process-1",))
26 p2 = multiprocessing.Process(target=process_task, args=("Process-2",))
27
28 # Start processes
29 p1.start()
30 p2.start()
31
32 # Wait for processes to finish
33 p1.join()
34 p2.join()
35
36 logging.info("System Shutdown Completed")
37 print("\nSystem Shutdown.")

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

ashishyadvanshi@Ashishs-MacBook-Air python % python3 main.py
System Booting...

Process-2 started...
Process-1 started...
Process-1 terminated.
Process-2 terminated.

System Shutdown.
ashishyadvanshi@Ashishs-MacBook-Air python %

zsh

OUTLINE

TIMELINE

Ln 20, Col 24

Spaces: 4

UTF-8

LF

Python

BLACKBOX Autocomplete: ON

BLACKBOXAI: Open Chat

Task 3: System Calls and IPC (Python - fork, exec, pipe)

Use system calls (fork(), exec(), wait()) and implement basic Inter-Process Communication using pipes in C or Python.

```
import os

r, w = os.pipe()

pid = os.fork()

if pid > 0:

    os.close(r)

    os.write(w, b"Hello from parent")

    os.close(w)

    os.wait()

else:

    os.close(w)

    message = os.read(r, 1024)

    print("Child received:", message.decode())

    os.close(r)
```

python

python

EXPLORER

PYTHON

main.py

process_log.txt

README.md

service_dependency_log.txt

system_boot_log.txt

system_log.txt

system_shutdown_log.txt

main.py

1 import multiprocessing

2 import os

3

4 def child_process(pipe_conn):

5 """Child reads data sent by Parent."""

6 message = pipe_conn.recv() # Read message from parent

7 print(f"[Child] PID = {os.getpid()} received: {message}")

8 pipe_conn.close()

9

10

11 def parent_child_pipe():

12 parent_conn, child_conn = multiprocessing.Pipe() # Create a pipe

13

14 print("=== PIPE IPC Simulation Started ===")

15

16 # Create child process

17 child = multiprocessing.Process(target=child_process, args=(child_conn,))

18

19 child.start()

20

21 # Parent sending message

22 msg = "Hello Child, this is Parent!"

23 print(f"[Parent] PID = {os.getpid()} sending: {msg}")

24 parent_conn.send(msg)

25

26 child.join()

27

28 print("\n=== PIPE IPC Simulation Completed ===")

29

30

31 if __name__ == "__main__":

32 parent_child_pipe()

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

ashishyaduvanshi@Ashishs-MacBook-Air python % python3 main.py

=== PIPE IPC Simulation Started ===

[Parent] PID = 34538 sending: Hello Child, this is Parent!

[Child] PID = 34540 received: Hello Child, this is Parent!

=== PIPE IPC Simulation Completed ===

ashishyaduvanshi@Ashishs-MacBook-Air python %

zsh

Ln 31, Col 24

Spaces: 4

UTF-8

LF

{ }

Python

BLACKBOX Autocomplete: ON

BLACKBOXAI: Open Chat

Task 4: VM Detection and Shell Interaction

Create a shell script to print system details and a Python script to detect if the system is running inside a virtual machine.

```
#!/bin/bash
```

```
echo "Kernel Version:"
```

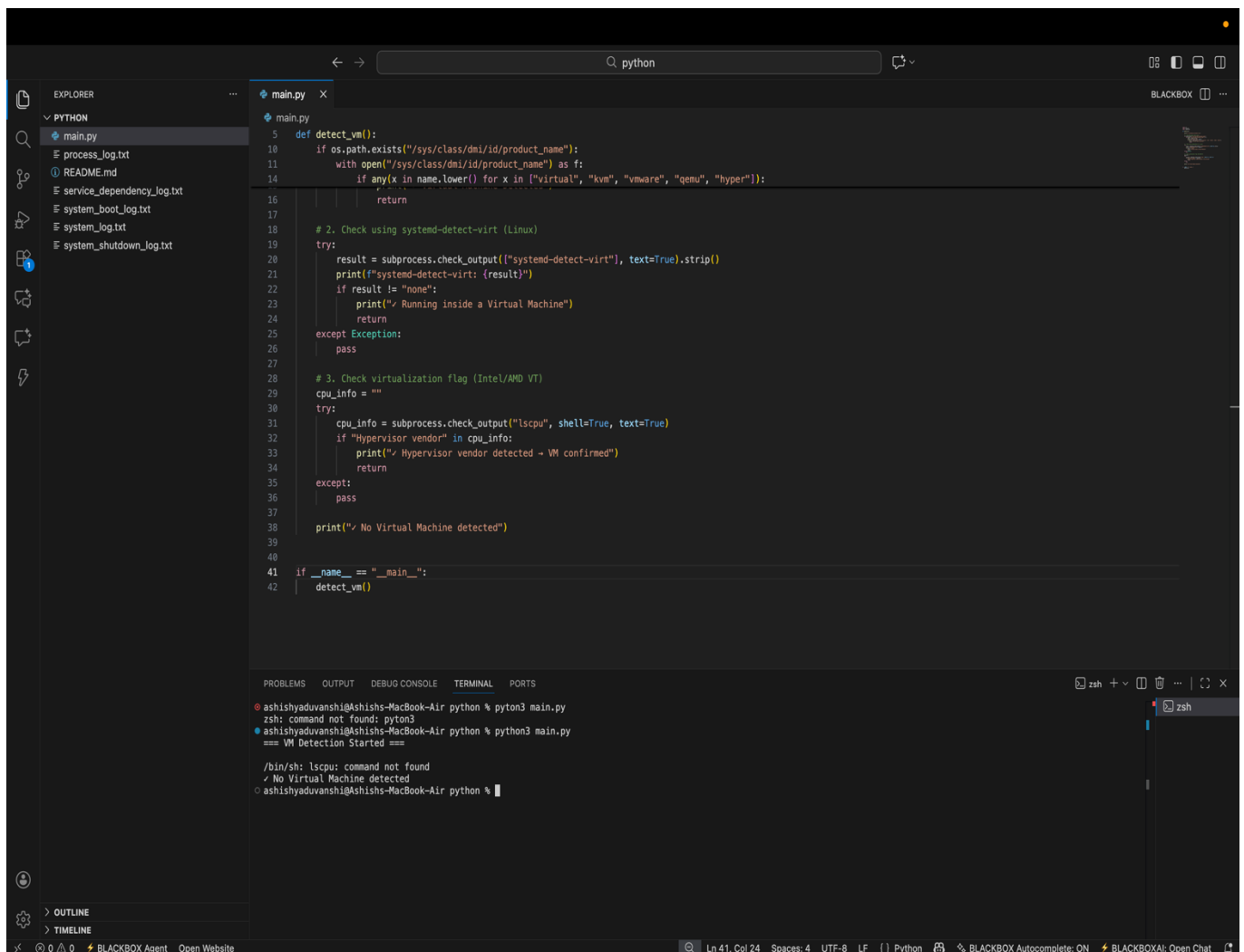
```
uname -r
```

```
echo "User:"
```

```
whoami
```

```
echo "Hardware Info:"
```

```
lscpu | grep 'Virtualization'
```



The screenshot shows a Visual Studio Code editor with a Python script named `main.py` open. The script is designed to detect if the system is running inside a virtual machine (VM) using three methods: checking the `/sys/class/dmi/id/product_name` file, using the `systemd-detect-virt` command, and checking the `lscpu` output for hypervisor vendors. The script prints the results of each check and a final message indicating whether a VM was detected.

```
5 def detect_vm():
6     if os.path.exists("/sys/class/dmi/id/product_name"):
7         with open("/sys/class/dmi/id/product_name") as f:
8             if any(x in name.lower() for x in ["virtual", "kvm", "vmware", "qemu", "hyper"]):
9                 return True
10    # 2. Check using systemd-detect-virt (Linux)
11    try:
12        result = subprocess.check_output(["systemd-detect-virt"], text=True).strip()
13        print(f"systemd-detect-virt: {result}")
14        if result != "none":
15            print(f"Running inside a Virtual Machine")
16            return True
17    except Exception:
18        pass
19    # 3. Check virtualization flag (Intel/AMD VT)
20    cpu_info = ""
21    try:
22        cpu_info = subprocess.check_output("lscpu", shell=True, text=True)
23        if "Hypervisor vendor" in cpu_info:
24            print(f"Hypervisor vendor detected - VM confirmed")
25            return True
26    except:
27        pass
28    print(f"No Virtual Machine detected")
29
30 if __name__ == "__main__":
31     detect_vm()
```

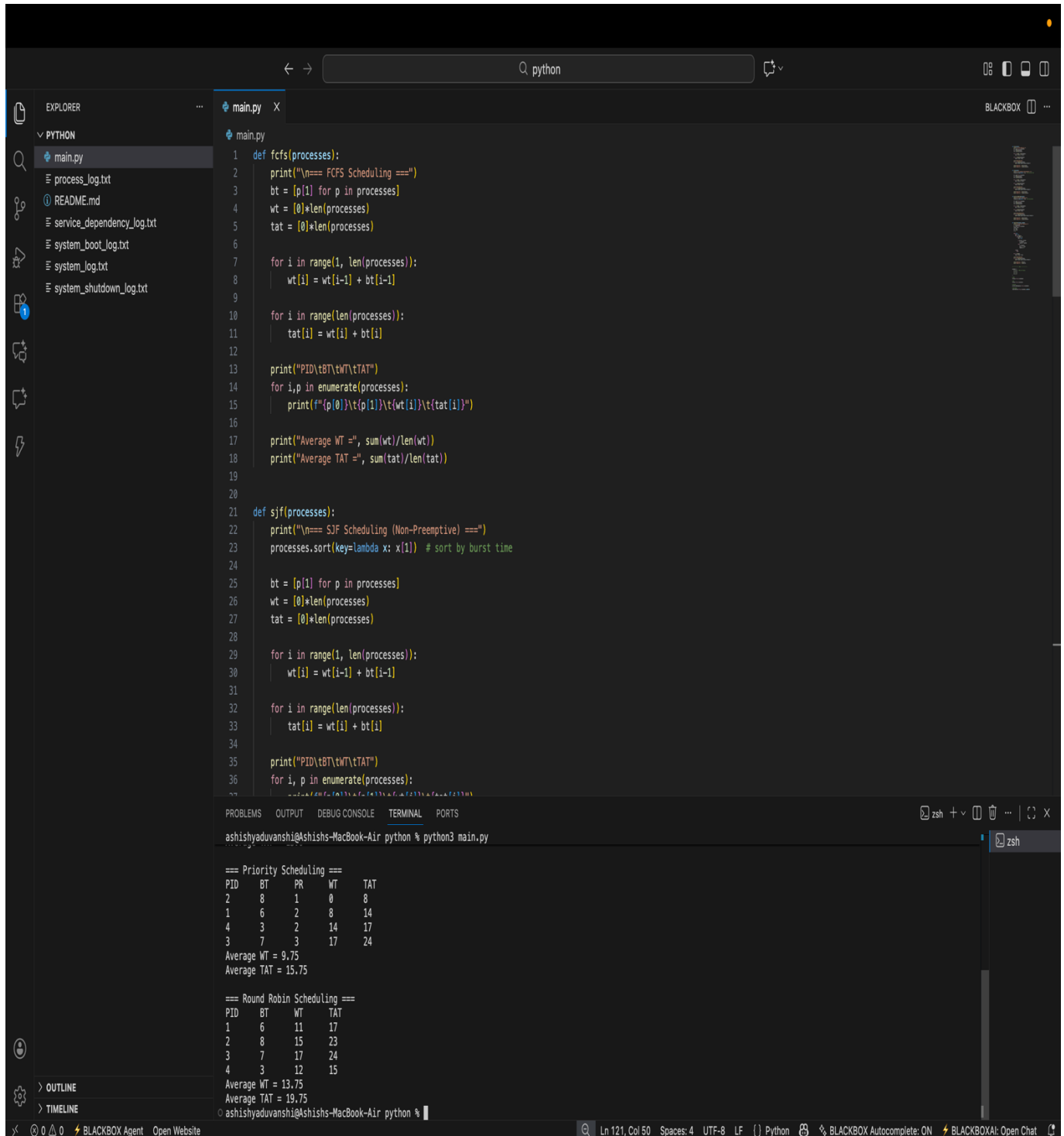
The terminal output shows the execution of the script. It starts with the command `python3 main.py`, which runs successfully. The output indicates that no virtual machine was detected, as the `lscpu` command was not found and the `systemd-detect-virt` command returned an empty result.

```
ashishyaduvanshi@Ashishs-MacBook-Air python % python3 main.py
zsh: command not found: python3
ashishyaduvanshi@Ashishs-MacBook-Air python % python3 main.py
=== VM Detection Started ===
/bin/sh: lscpu: command not found
✓ No Virtual Machine detected
ashishyaduvanshi@Ashishs-MacBook-Air python %
```

Task 5: CPU Scheduling Algorithms

Implement FCFS, SJF, Round Robin, and Priority Scheduling algorithms in Python to calculate WT and TAT.

Use existing Round Robin, FCFS, SJF, Priority scheduling Python codes from Lab 3)



```
1 def fcfs(processes):
2     print("\n=== FCFS Scheduling ===")
3     bt = [p[1] for p in processes]
4     wt = [0]*len(processes)
5     tat = [0]*len(processes)
6
7     for i in range(1, len(processes)):
8         wt[i] = wt[i-1] + bt[i-1]
9
10    for i in range(len(processes)):
11        tat[i] = wt[i] + bt[i]
12
13    print("PID\tBT\tWT\tTAT")
14    for i,p in enumerate(processes):
15        print(f"{p[0]}\t{p[1]}\t{wt[i]}\t{tat[i]}")
16
17    print("Average WT =", sum(wt)/len(wt))
18    print("Average TAT =", sum(tat)/len(tat))
19
20
21 def sjf(processes):
22     print("\n=== SJF Scheduling (Non-Preemptive) ===")
23     processes.sort(key=lambda x: x[1]) # sort by burst time
24
25     bt = [p[1] for p in processes]
26     wt = [0]*len(processes)
27     tat = [0]*len(processes)
28
29     for i in range(1, len(processes)):
30         wt[i] = wt[i-1] + bt[i-1]
31
32     for i in range(len(processes)):
33         tat[i] = wt[i] + bt[i]
34
35     print("PID\tBT\tWT\tTAT")
36     for i, p in enumerate(processes):
37         print(f"{p[0]}\t{p[1]}\t{wt[i]}\t{tat[i]}")
38
39     print("Average WT =", sum(wt)/len(wt))
40     print("Average TAT =", sum(tat)/len(tat))
41
42
43 if __name__ == '__main__':
44     processes = [
45         (1, 6), (2, 8), (3, 7), (4, 3)
46     ]
47     fcfs(processes)
48     sjf(processes)
```

ashishyaduvanshi@Ashishs-MacBook-Air python % python3 main.py

```
=== Priority Scheduling ===
PID  BT  PR  WT  TAT
2    8    1    0    8
1    6    2    8   14
4    3    2   14   17
3    7    3   17   24
Average WT = 9.75
Average TAT = 15.75

=== Round Robin Scheduling ===
PID  BT  WT  TAT
1    6   11   17
2    8   15   23
3    7   17   24
4    3   12   15
Average WT = 13.75
Average TAT = 19.75
ashishyaduvanshi@Ashishs-MacBook-Air python %
```