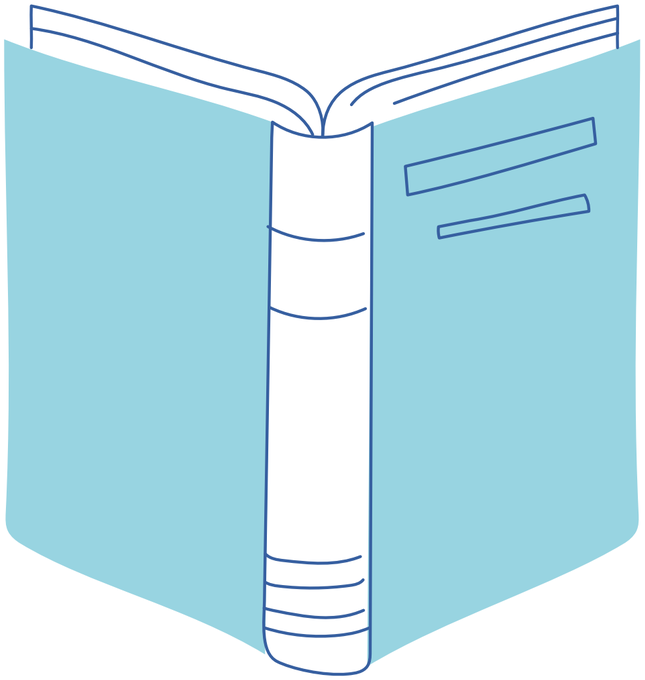
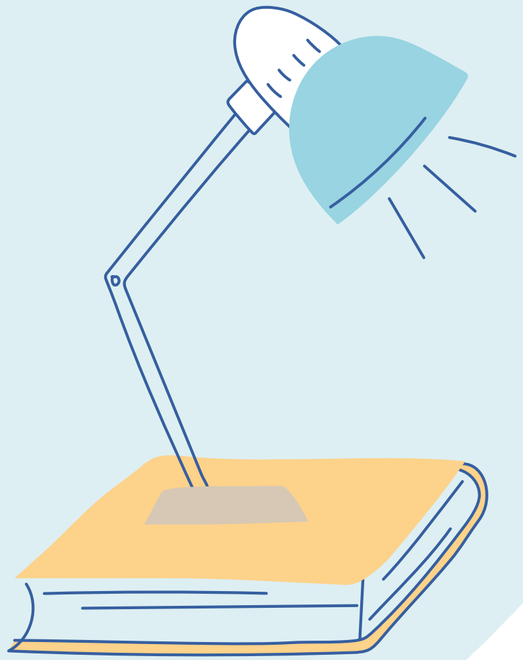


# Assignment

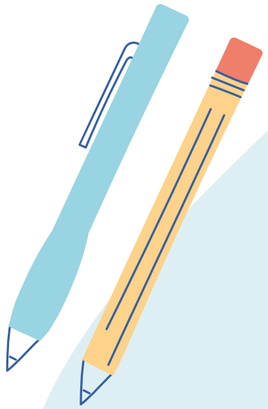
# Lab06-OS

By Prak Pychey



# Overview

- *Task 1: Running and Monitoring Processes*
- *Task 2: Managing Process Priority*
- *Task 3: Process Termination*
- *Task 4: Resource Management*



# Task 1

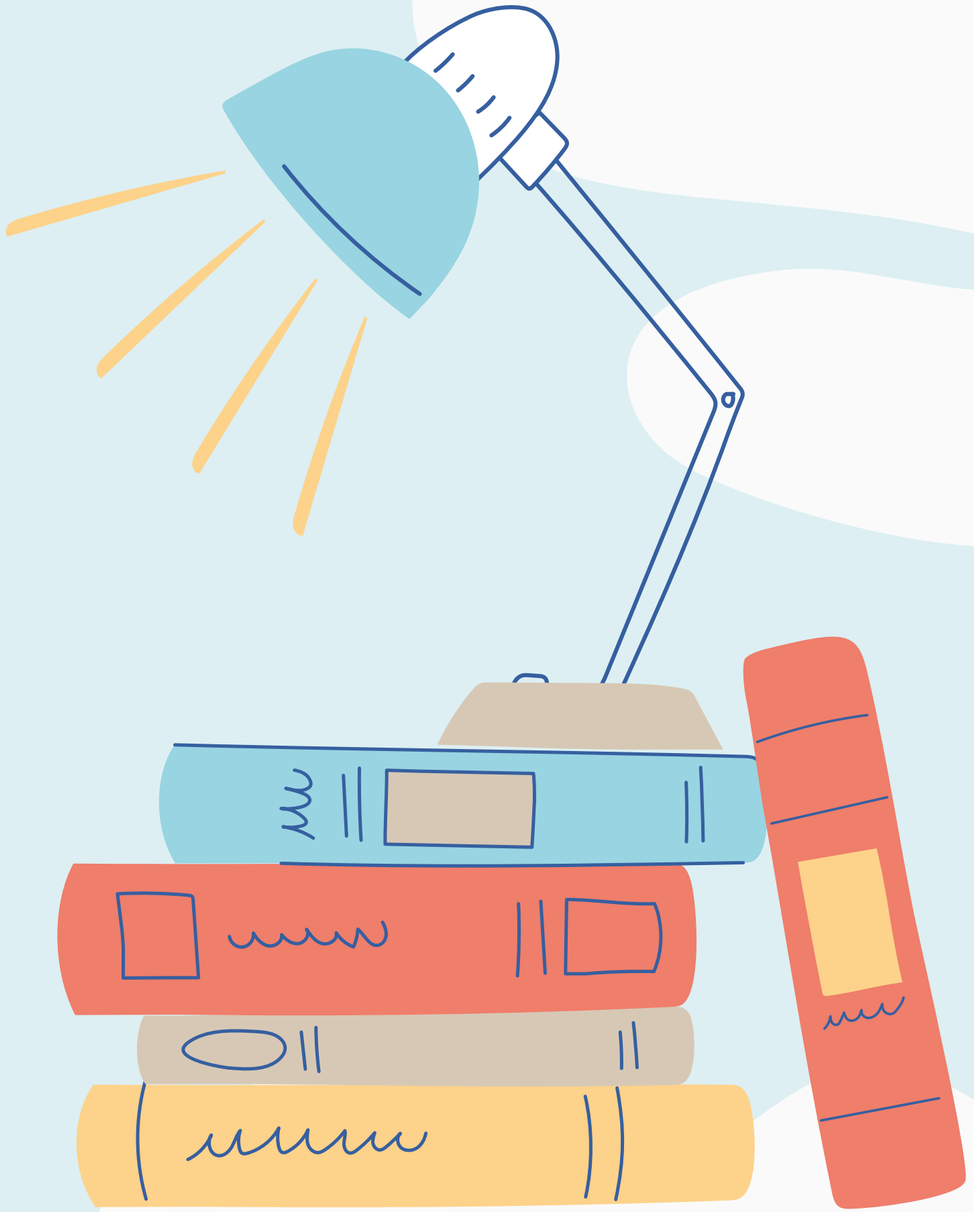
**Task 1: Running and Monitoring Processes**

1. Setup:
    a. Write three Python scripts: cpu_intensive.py, memory_intensive.py, and long_running.py.
    b. Run each script: python3 [script_name].py &
    c. Explain what does the command above do?
2. Monitoring:
    a. Use *ps aux* to list the processes.
    b. Observe CPU and memory usage with top or htop.
    *c.* Explain each attribute of the process associated with *ps aux*

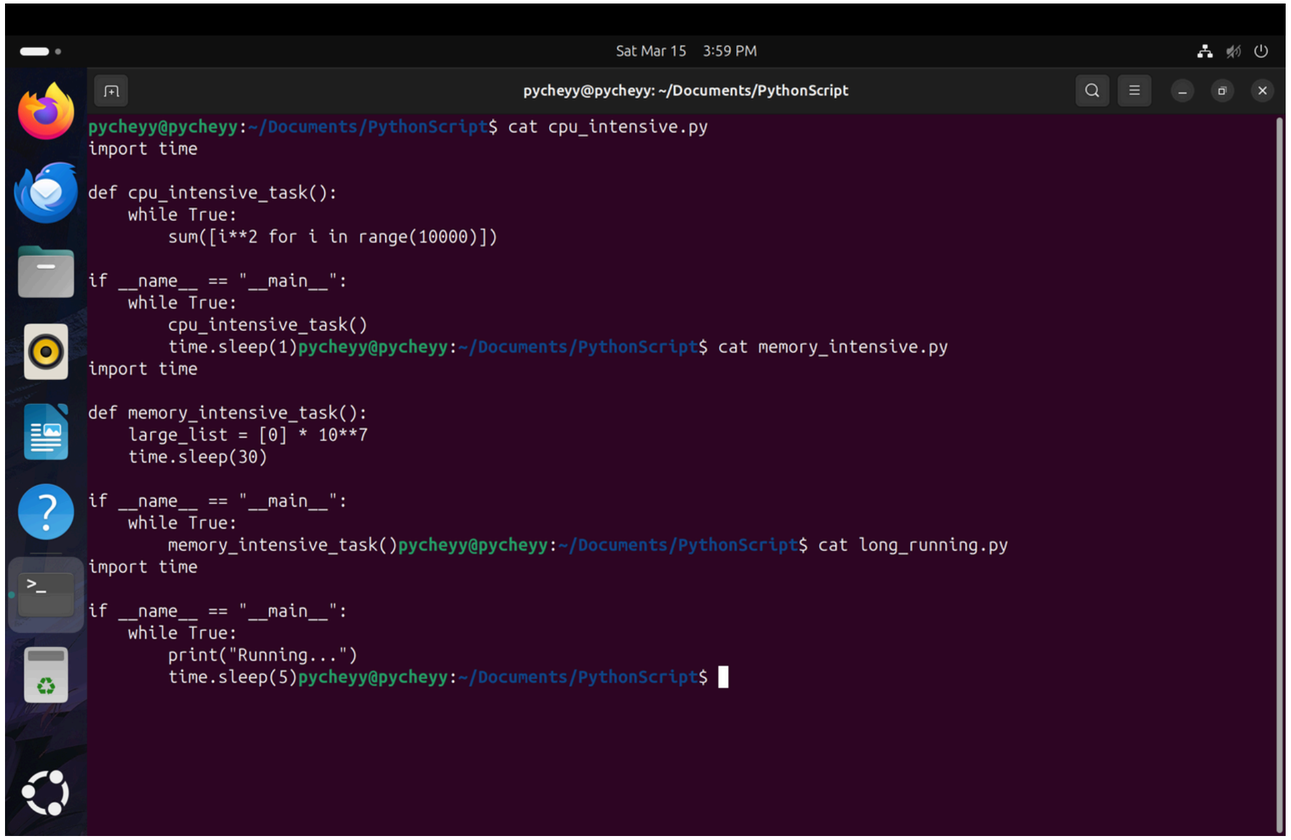
# Answer

1. c. The command python3 [script].py & allow the script to run in the background.
2. c. Each attributes of ps aux:

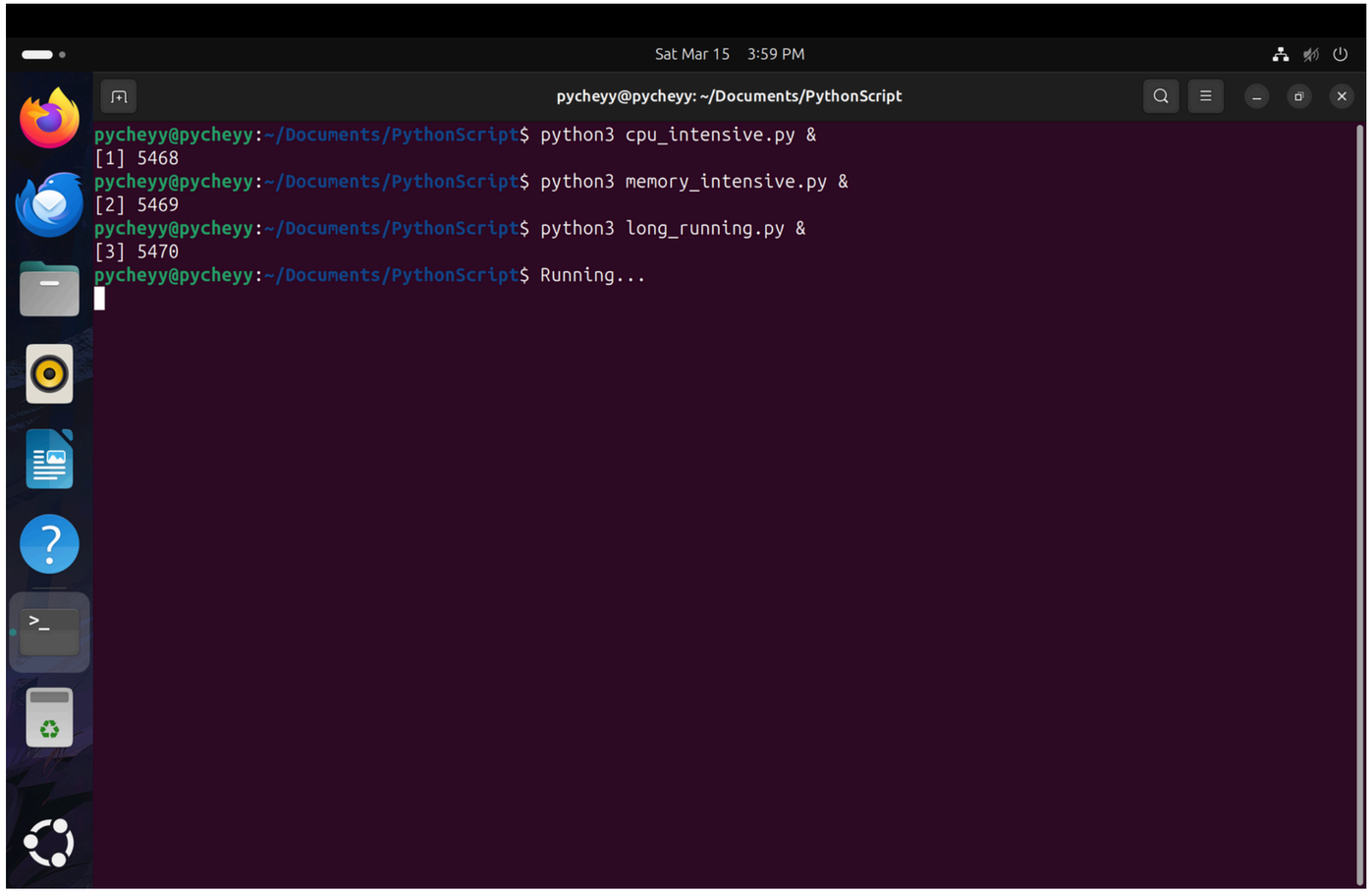
- USER: which user own the process
- PID: process ID
- %CPU: how much cpu is the process using in %
- %MEM: how much ram is the process using in %
- VSZ: show all virtual memory allocated in KB
- RSS: show all ram memory actually being used in KB
- TTY: which terminal process associate with
- STAT: process state such R for running and others
- START: process start time
- TIME: total time of CPU being used for process
- COMMAND: show command that start the process



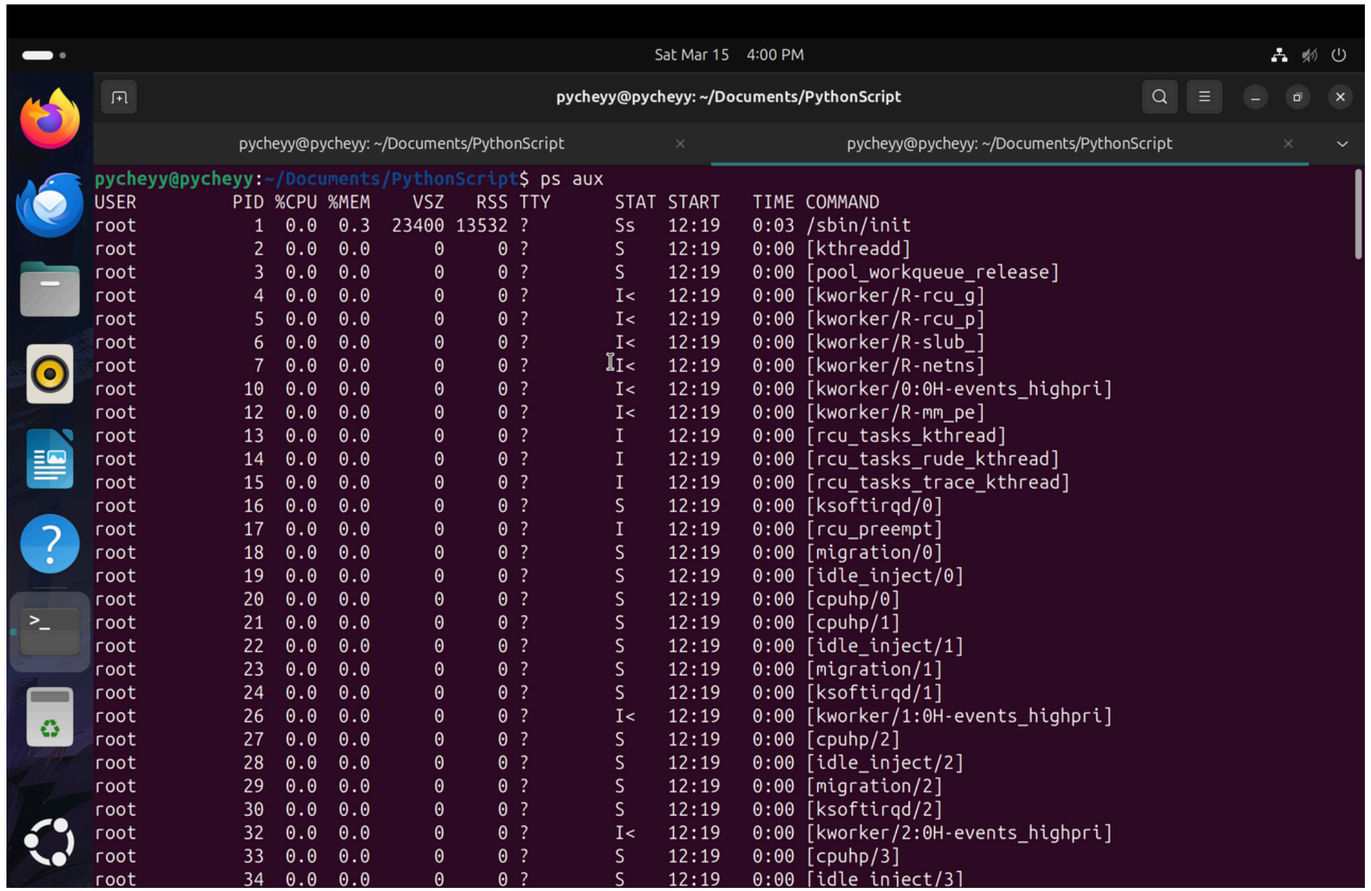
# 1.a


Sat Mar 15 3:59 PM
pycheyy@pycheyy: ~/Documents/PythonScript
pycheyy@pycheyy:~/Documents/PythonScript$ cat cpu_intensive.py
import time
def cpu_intensive_task():
while True:
sum([i**2 for i in range(10000)])
if __name__ == "__main__":
while True:
cpu_intensive_task()
time.sleep(1)pycheyy@pycheyy:~/Documents/PythonScript$ cat memory_intensive.py
import time
def memory_intensive_task():
large_list = [0] * 10**7
time.sleep(30)
if __name__ == "__main__":
while True:
memory_intensive_task()pycheyy@pycheyy:~/Documents/PythonScript$ cat long_running.py
import time
if __name__ == "__main__":
while True:
print("Running...")
time.sleep(5)pycheyy@pycheyy:~/Documents/PythonScript$

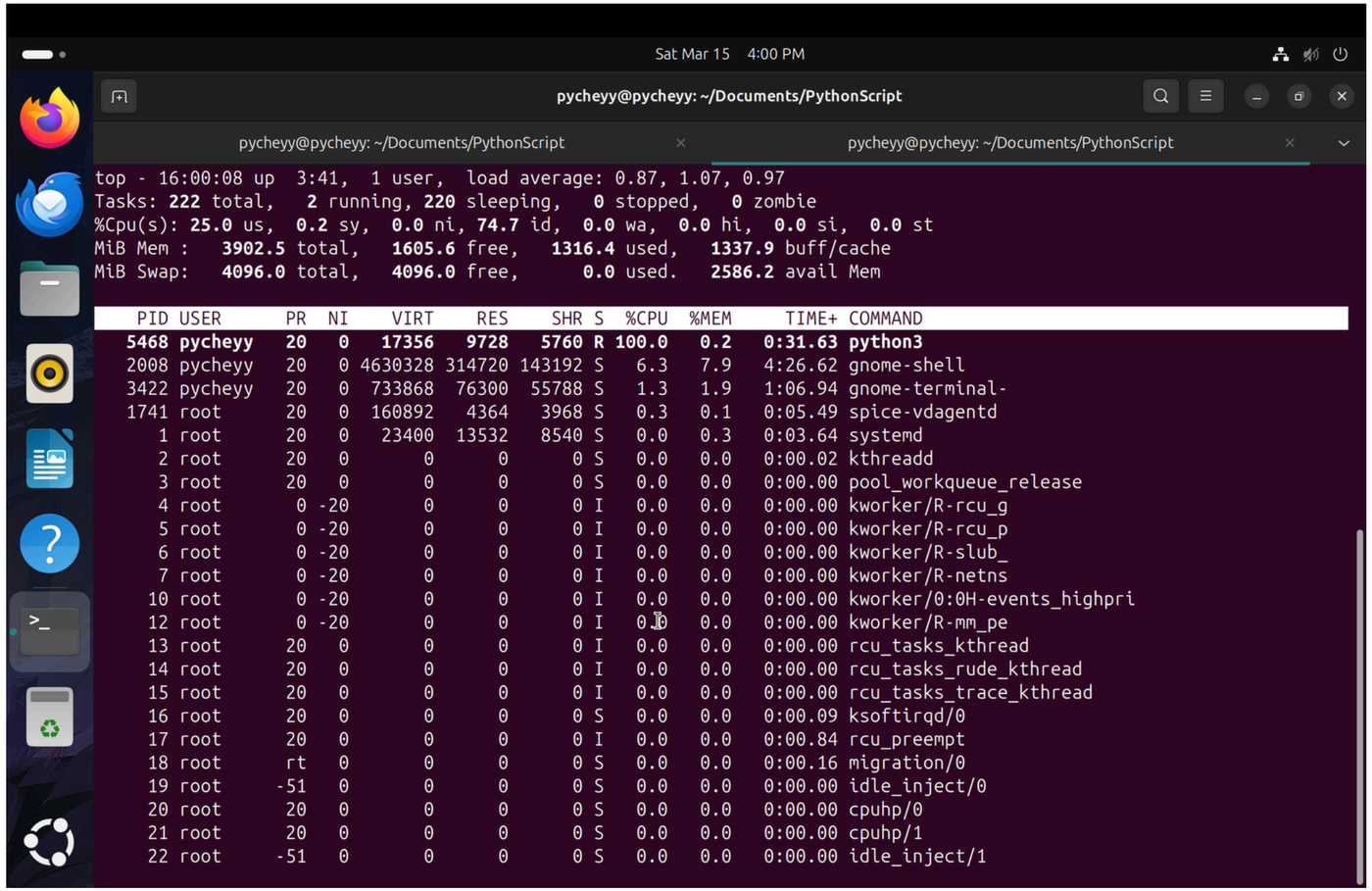
# 1.b


Sat Mar 15 3:59 PM
pycheyy@pycheyy: ~/Documents/PythonScript
pycheyy@pycheyy:~/Documents/PythonScript$ python3 cpu_intensive.py &
[1] 5468
pycheyy@pycheyy:~/Documents/PythonScript$ python3 memory_intensive.py &
[2] 5469
pycheyy@pycheyy:~/Documents/PythonScript$ python3 long_running.py &
[3] 5470
pycheyy@pycheyy:~/Documents/PythonScript$ Running...

# 2.a


Sat Mar 15 4:00 PM
pycheyy@pycheyy: ~/Documents/PythonScript
pycheyy@pycheyy: ~/Documents/PythonScript
pycheyy@pycheyy: ~/Documents/PythonScript
pycheyy@pycheyy:~/Documents/PythonScript$ ps aux
USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND
root 1 0.0 0.3 23400 13532 ? Ss 12:19 0:03 /sbin/init
root 2 0.0 0.0 0 0 ? S 12:19 0:00 [kthreadd]
root 3 0.0 0.0 0 0 ? S 12:19 0:00 [pool_workqueue_release]
root 4 0.0 0.0 0 0 ? I< 12:19 0:00 [kworker/R-rcu_g]
root 5 0.0 0.0 0 0 ? I< 12:19 0:00 [kworker/R-rcu_p]
root 6 0.0 0.0 0 0 ? I< 12:19 0:00 [kworker/R-slub_]
root 7 0.0 0.0 0 0 ? I< 12:19 0:00 [kworker/R-netns]
root 10 0.0 0.0 0 0 ? I< 12:19 0:00 [kworker/0:0H-events_highpri]
root 12 0.0 0.0 0 0 ? I< 12:19 0:00 [kworker/R-mm_pe]
root 13 0.0 0.0 0 0 ? I 12:19 0:00 [rcu_tasks_kthread]
root 14 0.0 0.0 0 0 ? I 12:19 0:00 [rcu_tasks_rude_kthread]
root 15 0.0 0.0 0 0 ? I 12:19 0:00 [rcu_tasks_trace_kthread]
root 16 0.0 0.0 0 0 ? S 12:19 0:00 [ksoftirqd/0]
root 17 0.0 0.0 0 0 ? I 12:19 0:00 [rcu_preempt]
root 18 0.0 0.0 0 0 ? S 12:19 0:00 [migration/0]
root 19 0.0 0.0 0 0 ? S 12:19 0:00 [idle_inject/0]
root 20 0.0 0.0 0 0 ? S 12:19 0:00 [cpuhp/0]
root 21 0.0 0.0 0 0 ? S 12:19 0:00 [cpuhp/1]
root 22 0.0 0.0 0 0 ? S 12:19 0:00 [idle_inject/1]
root 23 0.0 0.0 0 0 ? S 12:19 0:00 [migration/1]
root 24 0.0 0.0 0 0 ? S 12:19 0:00 [ksoftirqd/1]
root 26 0.0 0.0 0 0 ? I< 12:19 0:00 [kworker/1:0H-events_highpri]
root 27 0.0 0.0 0 0 ? S 12:19 0:00 [cpuhp/2]
root 28 0.0 0.0 0 0 ? S 12:19 0:00 [idle_inject/2]
root 29 0.0 0.0 0 0 ? S 12:19 0:00 [migration/2]
root 30 0.0 0.0 0 0 ? S 12:19 0:00 [ksoftirqd/2]
root 32 0.0 0.0 0 0 ? I< 12:19 0:00 [kworker/2:0H-events_highpri]
root 33 0.0 0.0 0 0 ? S 12:19 0:00 [cpuhp/3]
root 34 0.0 0.0 0 0 ? S 12:19 0:00 [idle_inject/3]

# 2.b


Sat Mar 15 4:00 PM
pycheyy@pycheyy: ~/Documents/PythonScript
pycheyy@pycheyy: ~/Documents/PythonScript
pycheyy@pycheyy: ~/Documents/PythonScript
top - 16:00:08 up 3:41, 1 user, load average: 0.87, 1.07, 0.97
Tasks: 222 total, 2 running, 220 sleeping, 0 stopped, 0 zombie
%Cpu(s): 25.0 us, 0.2 sy, 0.0 ni, 74.7 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 3902.5 total, 1605.6 free, 1316.4 used, 1337.9 buff/cache
MiB Swap: 4096.0 total, 4096.0 free, 0.0 used. 2586.2 avail Mem
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
5468 pycheyy 20 0 17356 9728 5760 R 100.0 0.2 0:31.63 python3
2008 pycheyy 20 0 4630328 314720 143192 S 6.3 7.9 4:26.62 gnome-shell
3422 pycheyy 20 0 733868 76300 55788 S 1.3 1.9 1:06.94 gnome-terminal-
1741 root 20 0 160892 4364 3968 S 0.3 0.1 0:05.49 spice-vdagentd
1 root 20 0 23400 13532 8540 S 0.0 0.3 0:03.64 systemd
2 root 20 0 0 0 0 S 0.0 0.0 0:00.02 kthreadd
3 root 20 0 0 0 0 S 0.0 0.0 0:00.00 pool_workqueue_release
4 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/R-rcu_g
5 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/R-rcu_p
6 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/R-slub_
7 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/R-netns
10 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/0:0H-events_highpri
12 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker/R-mm_pe
13 root 20 0 0 0 0 I 0.0 0.0 0:00.00 rcu_tasks_kthread
14 root 20 0 0 0 0 I 0.0 0.0 0:00.00 rcu_tasks_rude_kthread
15 root 20 0 0 0 0 I 0.0 0.0 0:00.00 rcu_tasks_trace_kthread
16 root 20 0 0 0 0 S 0.0 0.0 0:00.09 ksoftirqd/0
17 root 20 0 0 0 0 I 0.0 0.0 0:00.84 rcu_preempt
18 root rt 0 0 0 0 S 0.0 0.0 0:00.16 migration/0
19 root -51 0 0 0 0 S 0.0 0.0 0:00.00 idle_inject/0
20 root 20 0 0 0 0 S 0.0 0.0 0:00.00 cpuhp/0
21 root 20 0 0 0 0 S 0.0 0.0 0:00.00 cpuhp/1
22 root -51 0 0 0 0 S 0.0 0.0 0:00.00 idle_inject/1

# Task 2

**Task 2: Managing Process Priority**

3. Prioritization:
    a. Identify the PID of *cpu_intensive.py*.
    b. Lower its priority using renice.
    Priority range: -20 ( highest prio) : 19 ( lowest prio )

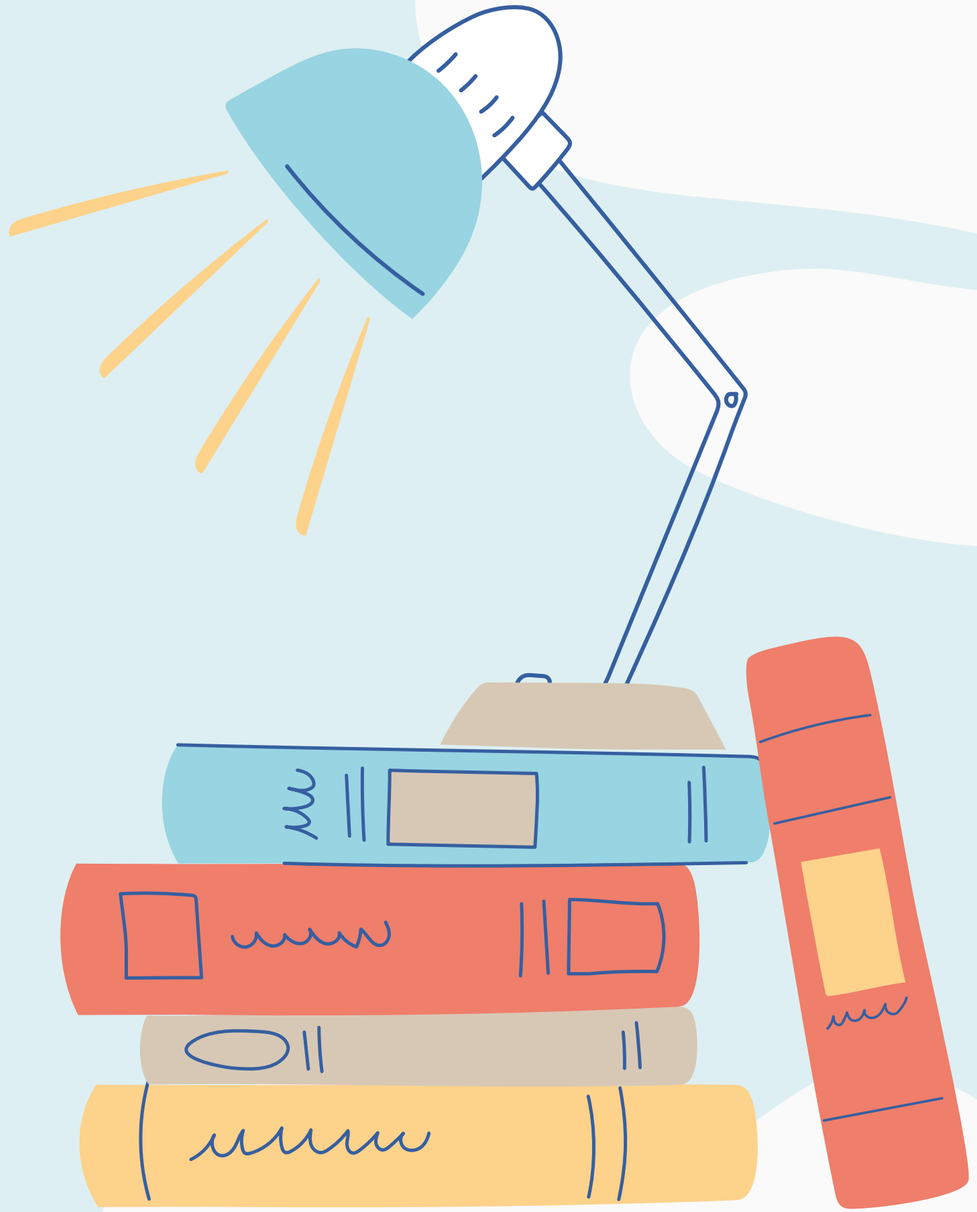
    nice -n 10 python3 cpu_intensive_1.py &
    nice -n -1 python3 cpu_intensive_2.py &

    renice -n -10 -p  PID of cpu_intensive_1.py &
    renice -n 5 -p PID cpu_intensive_2.py &

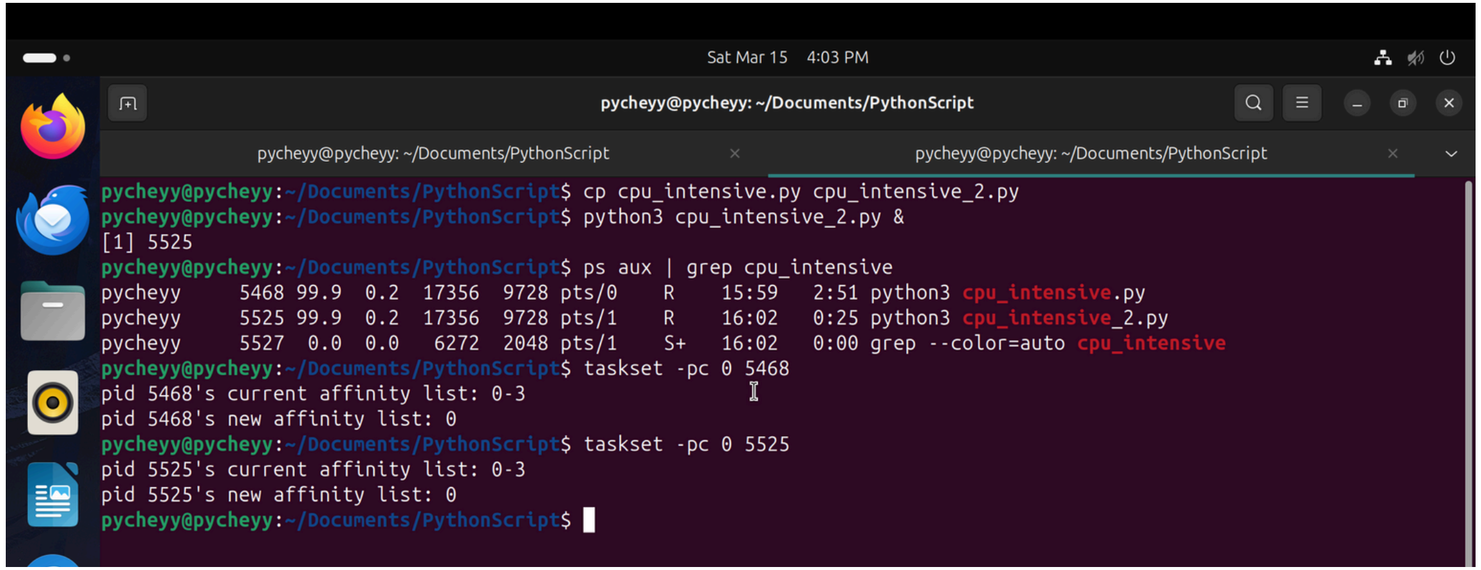
# Answer

3. The following steps:

- I first make another copy of the cpu_intensive and name it cpu_intensive_2.py
- run the cpu2.py
- identify both cpu.py PID
- use taskset -pc to make sure it run the same cpu core
- use top -p cpu1-PID, cpu2-PID to look
- then i press c to see the process name detail
- then i set its PI but since both file already running, i used renice for both case.

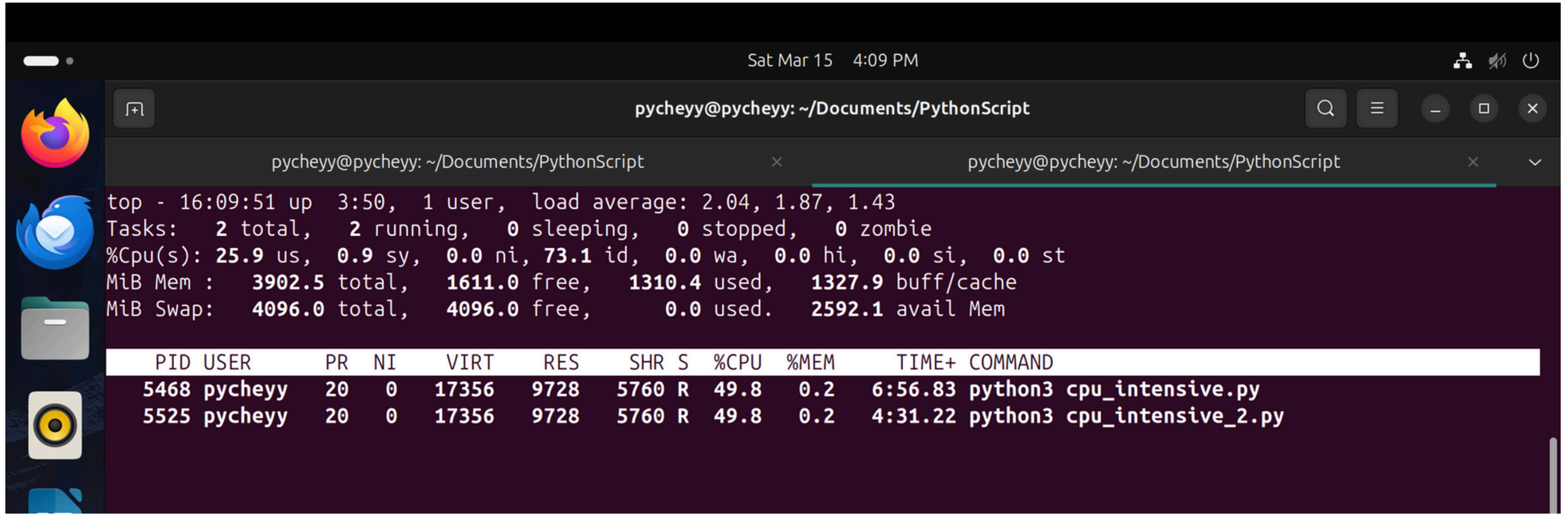


# Answer



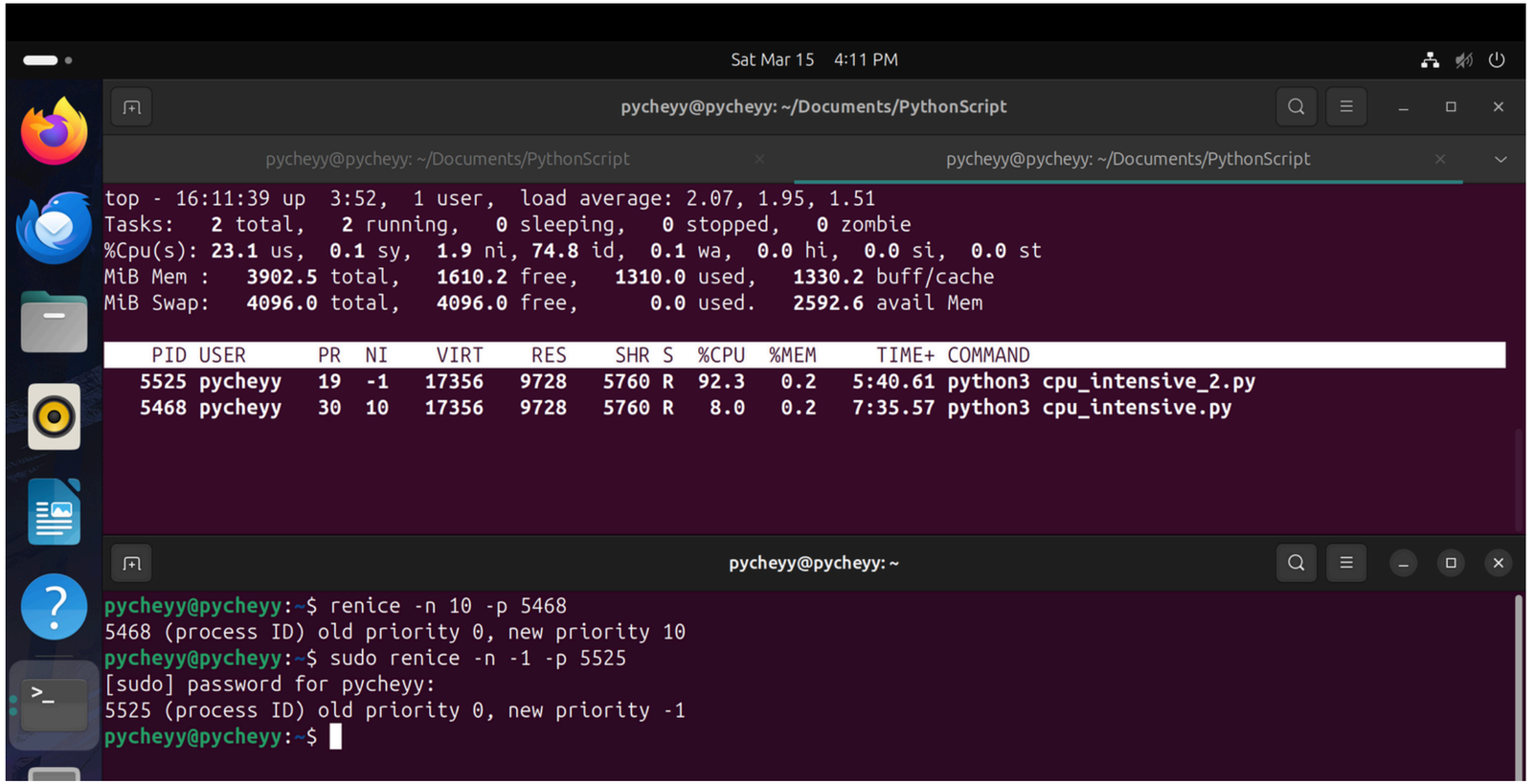

- I first make another copy of the cpu_intensive and name it cpu_intensive_2.py
- run the cpu2.py
- identify both cpu.py PID
- use taskset -pc to make sure it run the same cpu core

# Answer



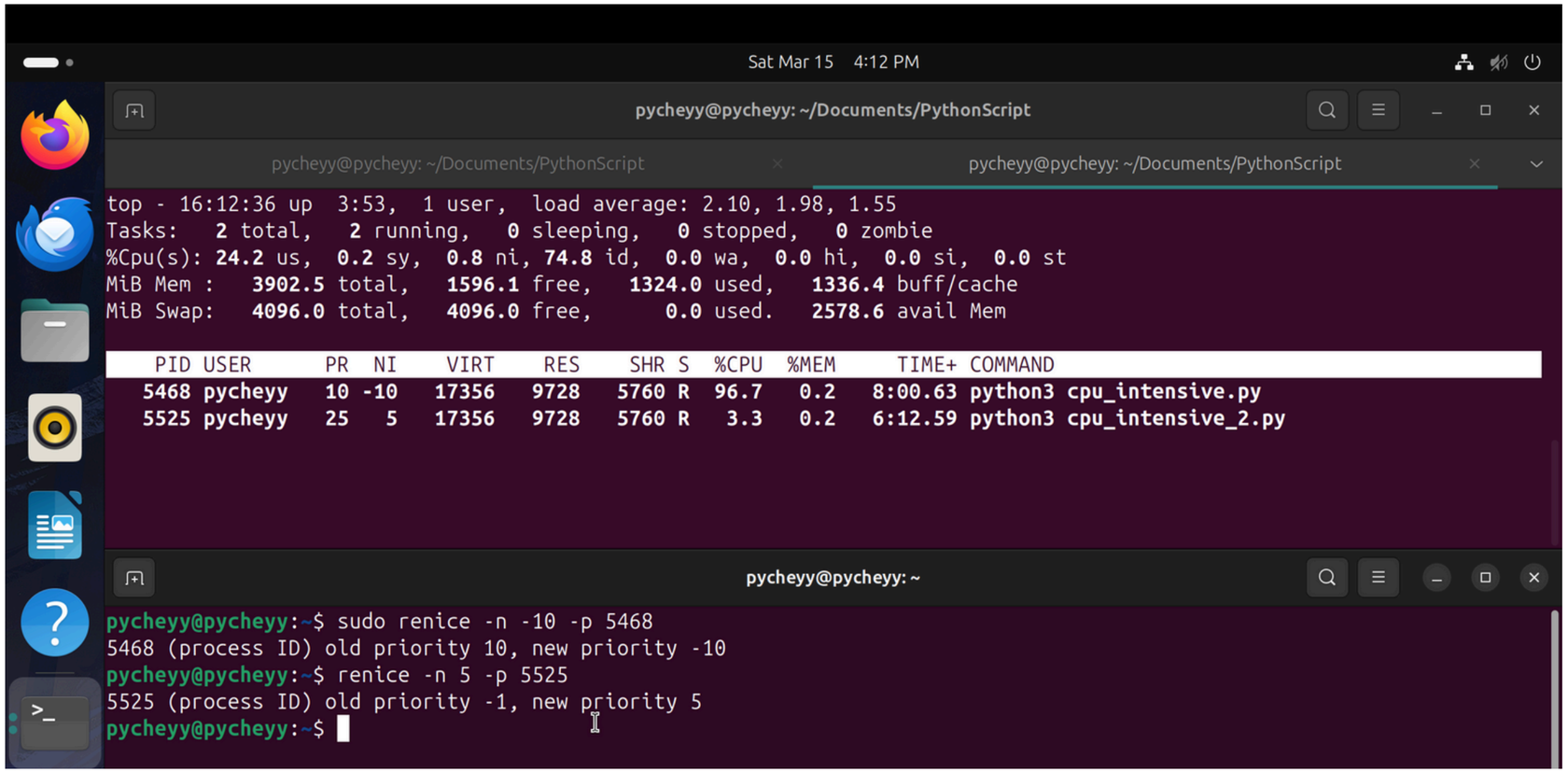

- use top -p cpu1-PID, cpu2-PID to look
- then i press c to see the process name detail

# Answer




- since both file already running, i used renice for both case instead of nice to start the process.
- nice -n -10 for cpu1 and sudo nice -n -1 for cpu2

# Answer




- sudo renice -n -10 for cpu1 and renice -n 5 for cpu2

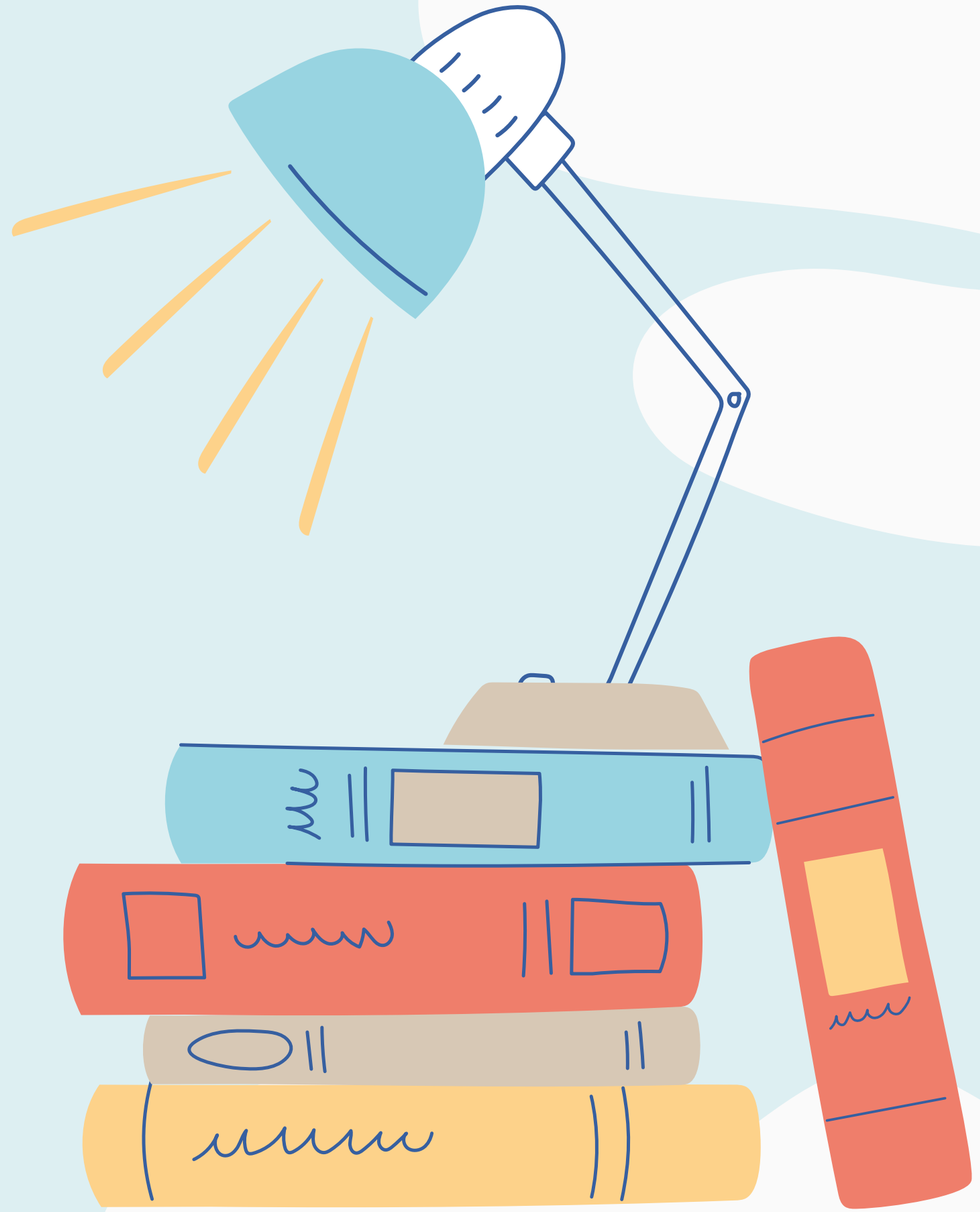
# Task 3

**Task 3: Process Termination**

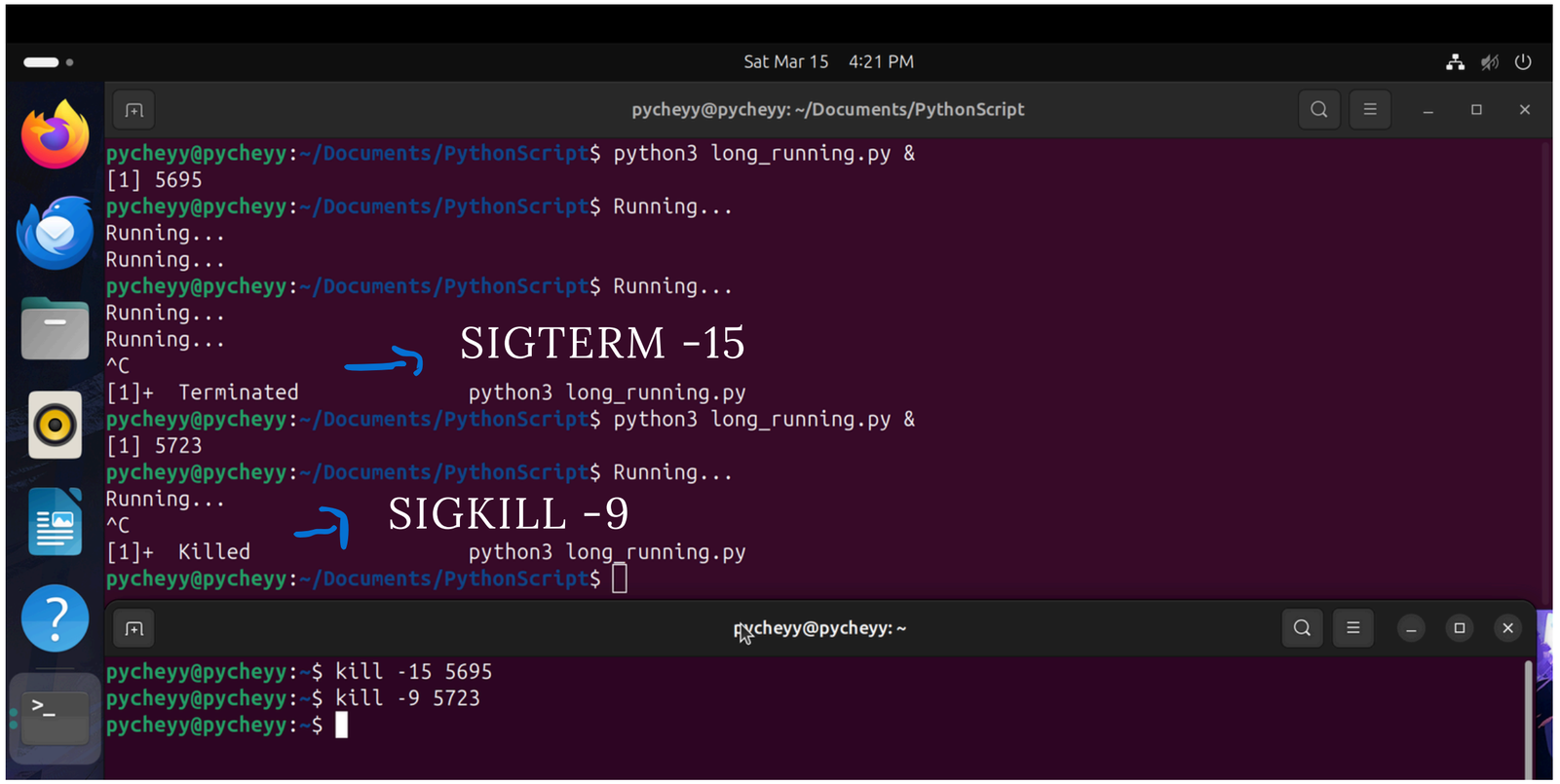
4. Termination:
    a. Use kill to terminate long_running.py with SIGTERM and SIGKILL.
    b. Explain the differences between SIGTERM and SIGKILL.

# Answer

4. The different between sigterm and sigkill is that:
   - SIGTERM: politely ask the process to terminate. allowing cleanup.
   - SIGKILL: forcefully kill the process  immediately without allowing cleanup.



# Answer

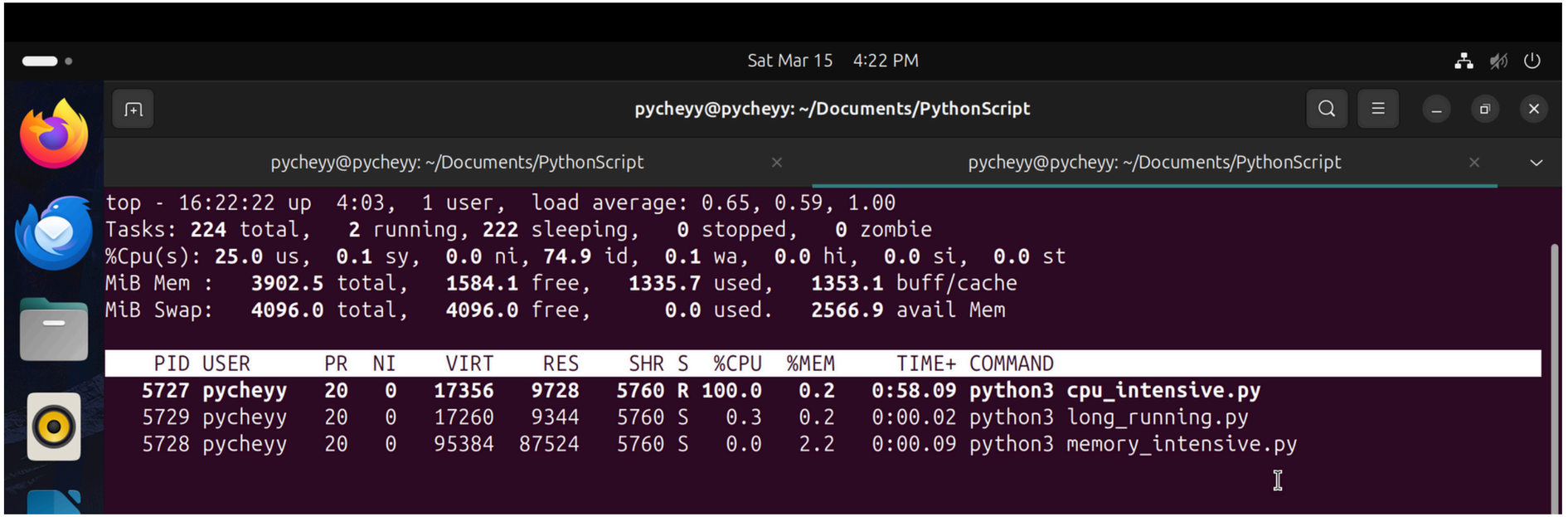

Sat Mar 15 4:21 PM
pycheyy@pycheyy: ~/Documents/PythonScript
pycheyy@pycheyy:~/Documents/PythonScript$ python3 long_running.py &
[1] 5695
pycheyy@pycheyy:~/Documents/PythonScript$ Running...
Running...
Running...
pycheyy@pycheyy:~/Documents/PythonScript$ Running...
Running...
Running...
^C
[1]+ Terminated python3 long_running.py
pycheyy@pycheyy:~/Documents/PythonScript$ python3 long_running.py &
[1] 5723
pycheyy@pycheyy:~/Documents/PythonScript$ Running...
Running...
^C
[1]+ Killed python3 long_running.py
pycheyy@pycheyy:~/Documents/PythonScript$
SIGTERM -15
SIGKILL -9
pycheyy@pycheyy: ~
pycheyy@pycheyy:~$ kill -15 5695
pycheyy@pycheyy:~$ kill -9 5723
pycheyy@pycheyy:~$

# Task 4

**Task 4: Resource Management**

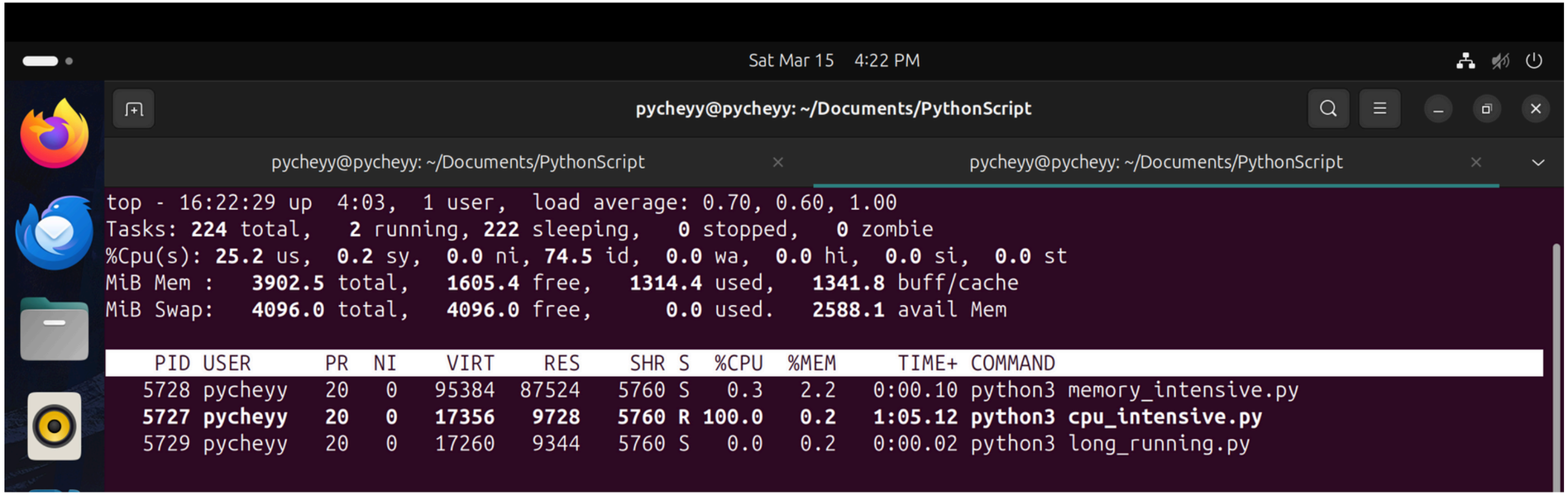
5. Resource Observation:
    a. Determine which script is most resource-intensive using top or htop.

# Order by CPU%




cpu_intensive use the most cpu percentage which is close to 100%

# Order by MEM%




memory_intensive use the most memory percentage of RAM which is 2.2%

# Thank you

*Prak Pychey*

*prakpycheyy@gmail.com*

# Resources