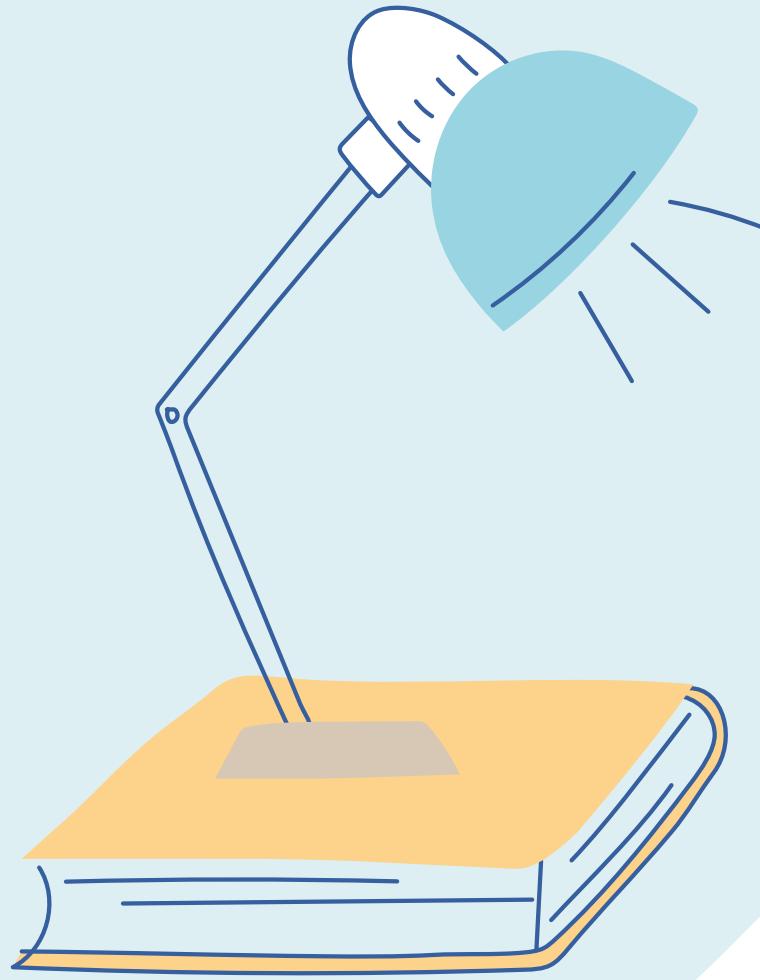


Assignment

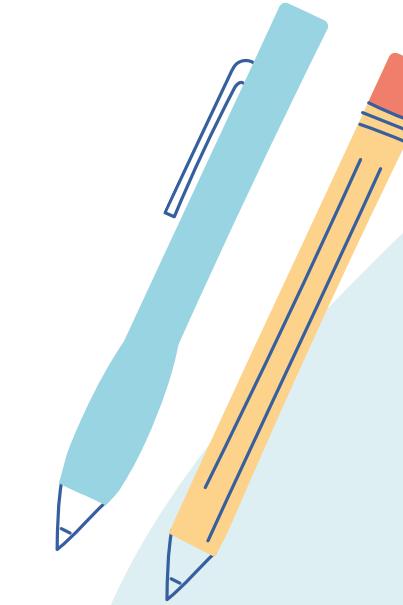
Lab06-OS

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`

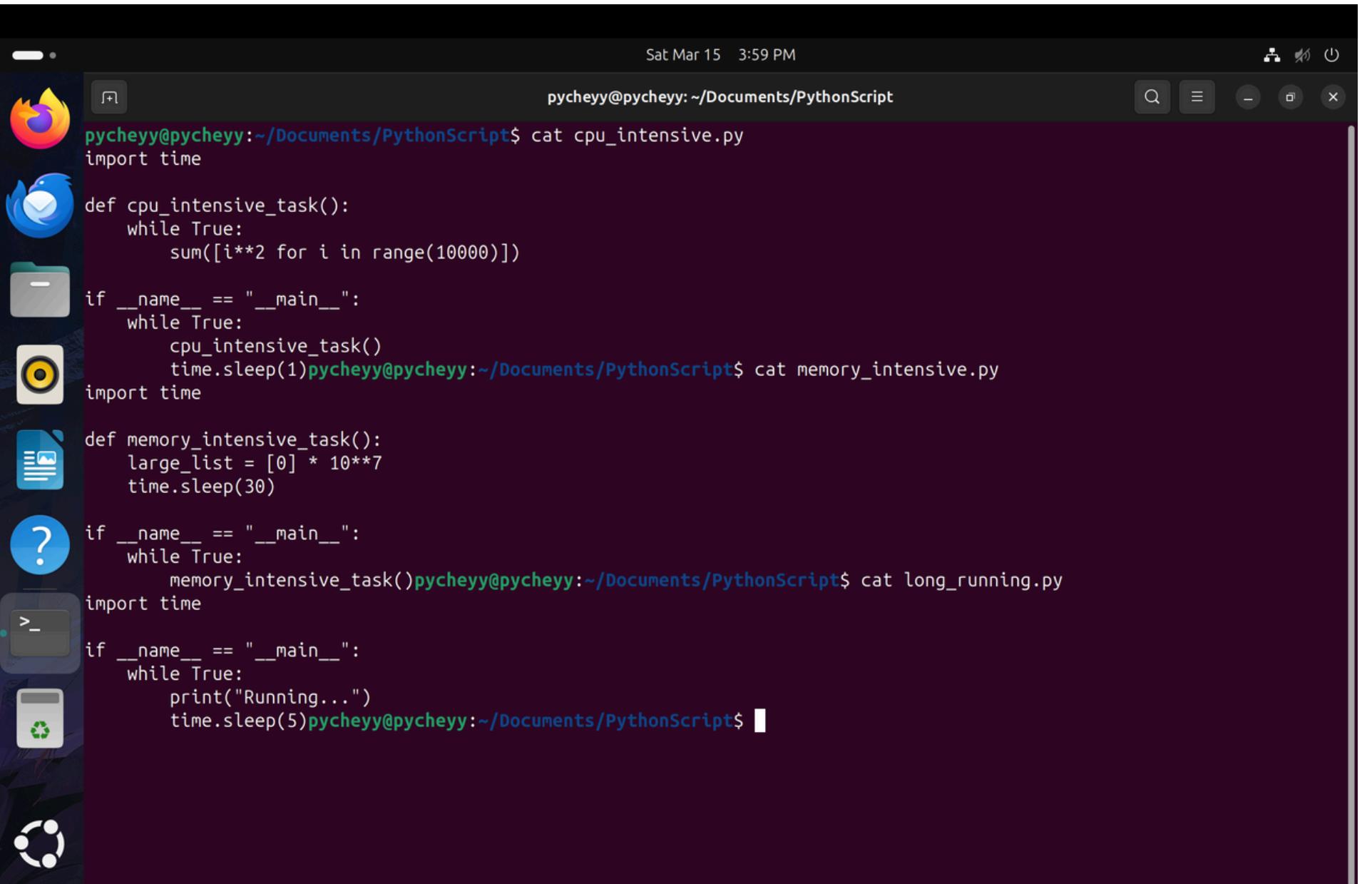


Answer

- 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

A screenshot of a Linux desktop environment, likely Ubuntu, featuring a dark theme. A terminal window is open in the center, displaying Python code. The terminal title bar shows "pycheyy@pycheyy: ~/Documents/PythonScript". The code consists of three files: "cpu_intensive.py", "memory_intensive.py", and "long_running.py".

```
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

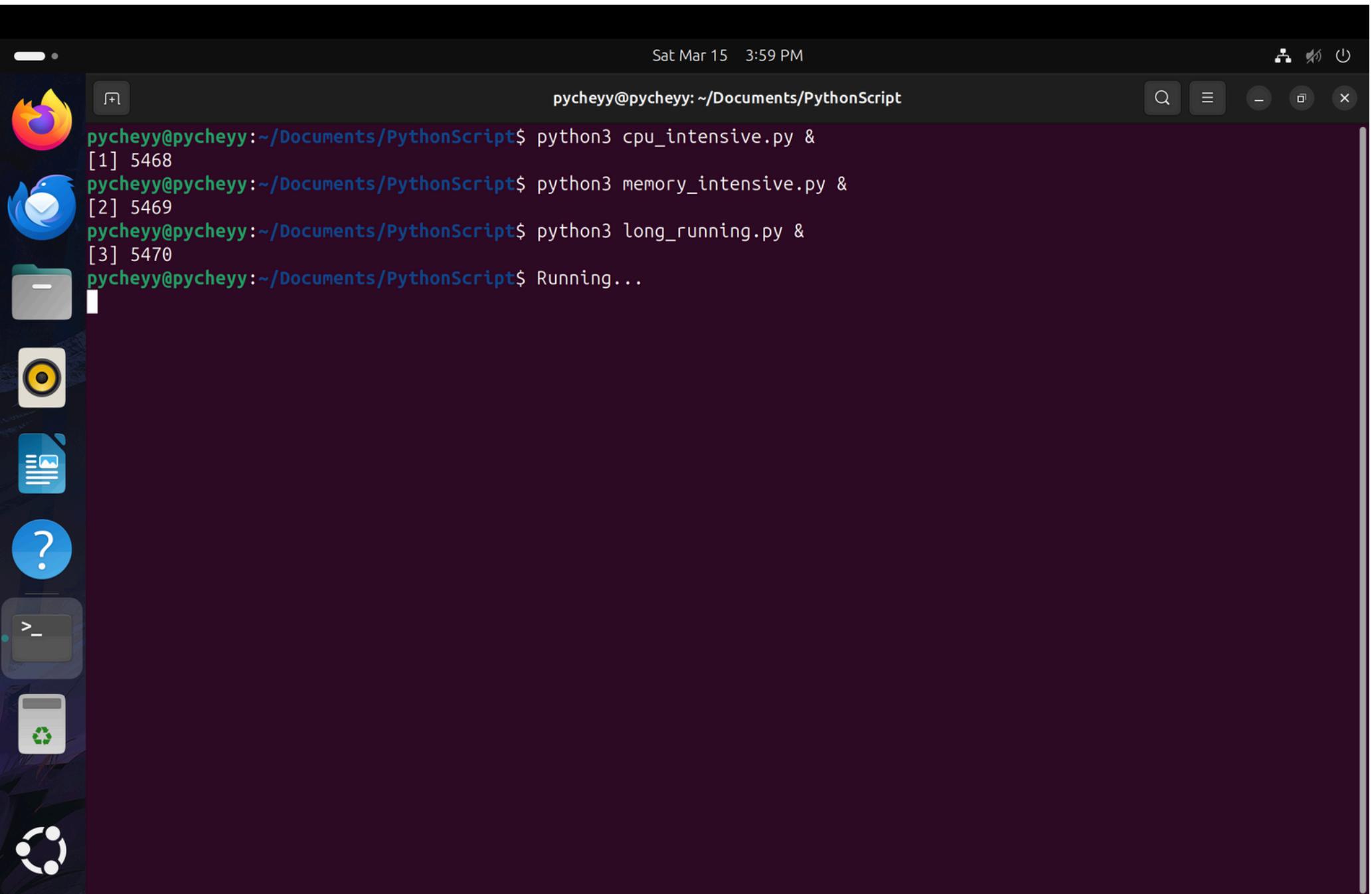
def memory_intensive_task():
    large_list = [0] * 10**7
    time.sleep(30)

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$
```

The desktop background features a light blue and white abstract wave pattern. The taskbar on the left lists icons for various applications, including a browser, file manager, terminal, and system settings.

1.b

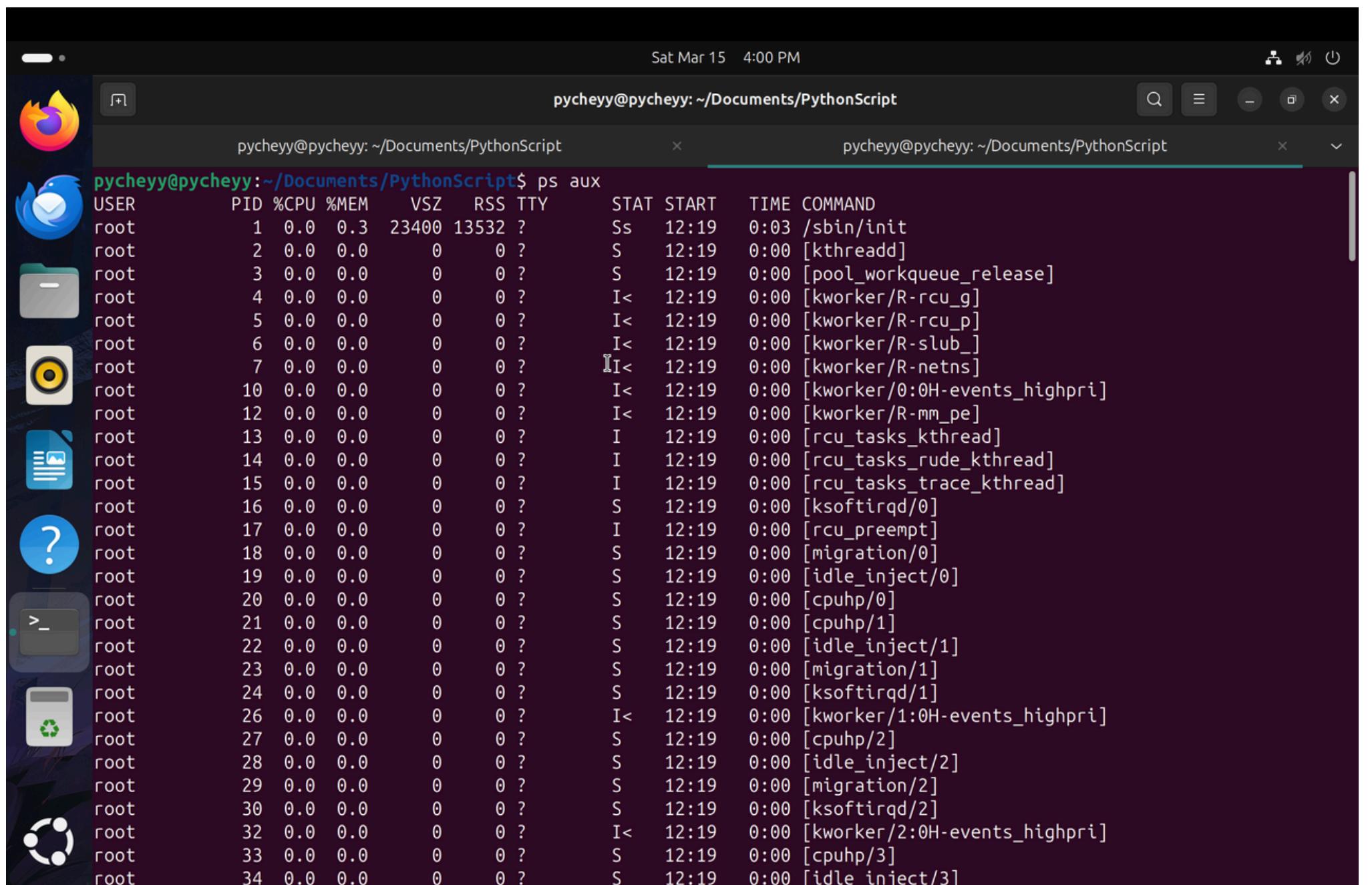


A screenshot of a terminal window on a Linux desktop environment. The terminal has a dark purple background and a black header bar. The header bar displays the date and time as "Sat Mar 15 3:59 PM" and the current working directory as "pycheyy@pycheyy: ~/Documents/PythonScript". The terminal window title is also "pycheyy@pycheyy: ~/Documents/PythonScript". The command line shows the user executing three parallel Python scripts:

```
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...
```

The terminal window has a standard window control bar at the top with icons for minimize, maximize, and close. To the left of the terminal window, there is a vertical dock containing icons for various applications: a browser (Firefox), a file manager (Nautilus), a terminal (Konsole), a help icon (question mark), a file icon, a system tray icon, a recycle bin, and the Unity logo.

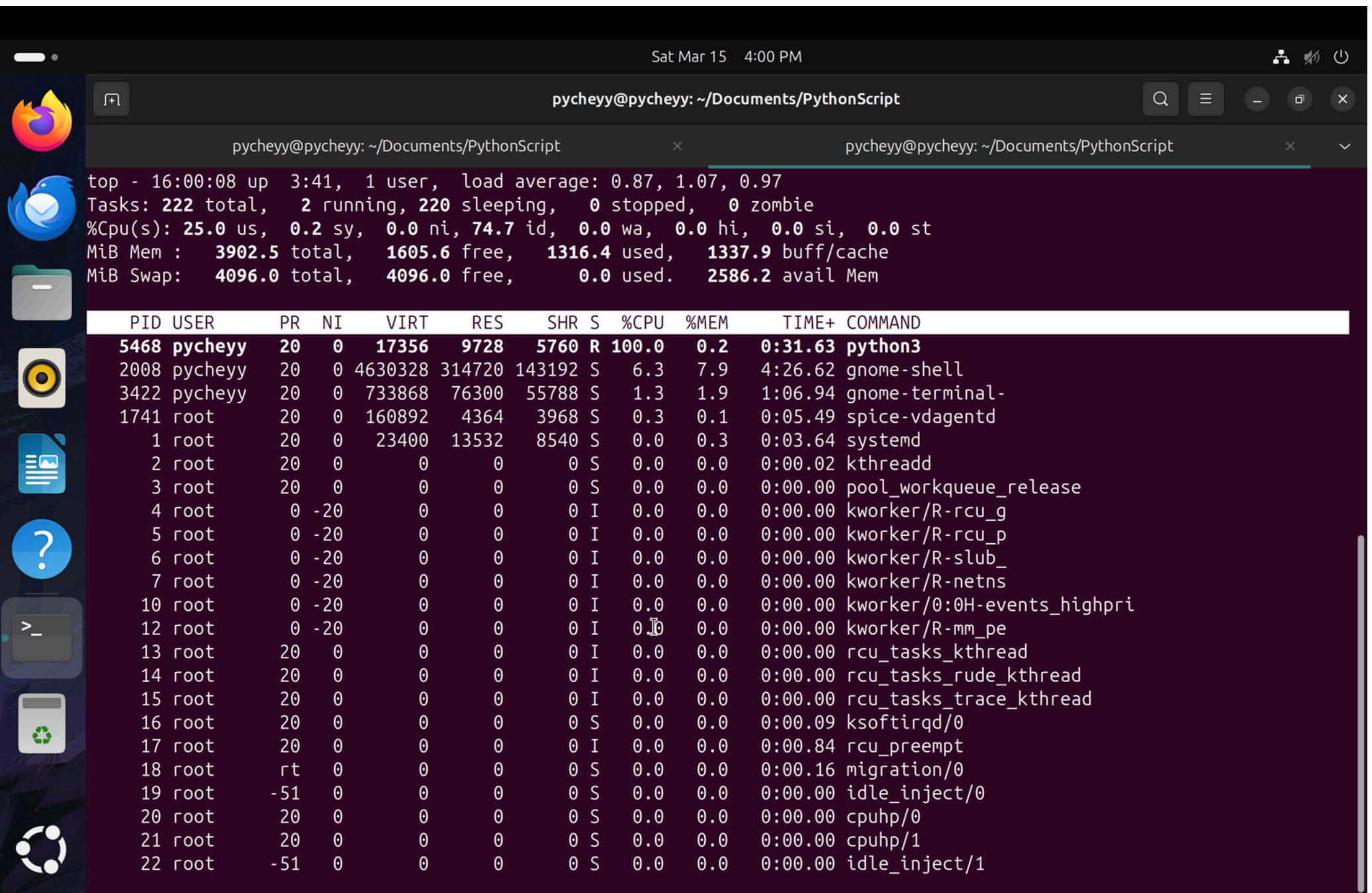
2.a



A screenshot of a Linux terminal window titled "pycheyy@pycheyy: ~/Documents/PythonScript". The window shows the output of the "ps aux" command. The terminal has a dark background and light-colored text. The output lists numerous processes running as root, primarily kernel-related tasks like "kworker" and "rcu_tasks_kthread". The terminal window is part of a desktop environment, with icons for various applications visible on the left side of the screen.

```
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 ?        II< 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
```

```
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 &
```



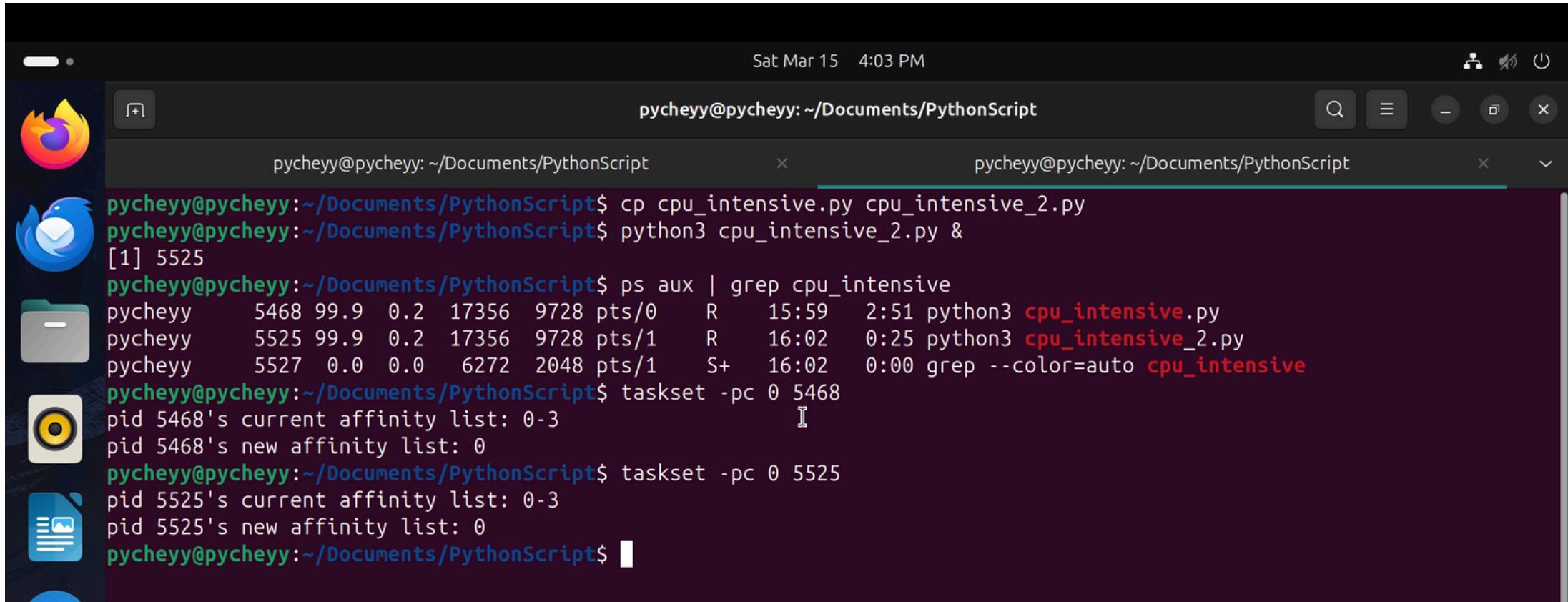
Answer

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.



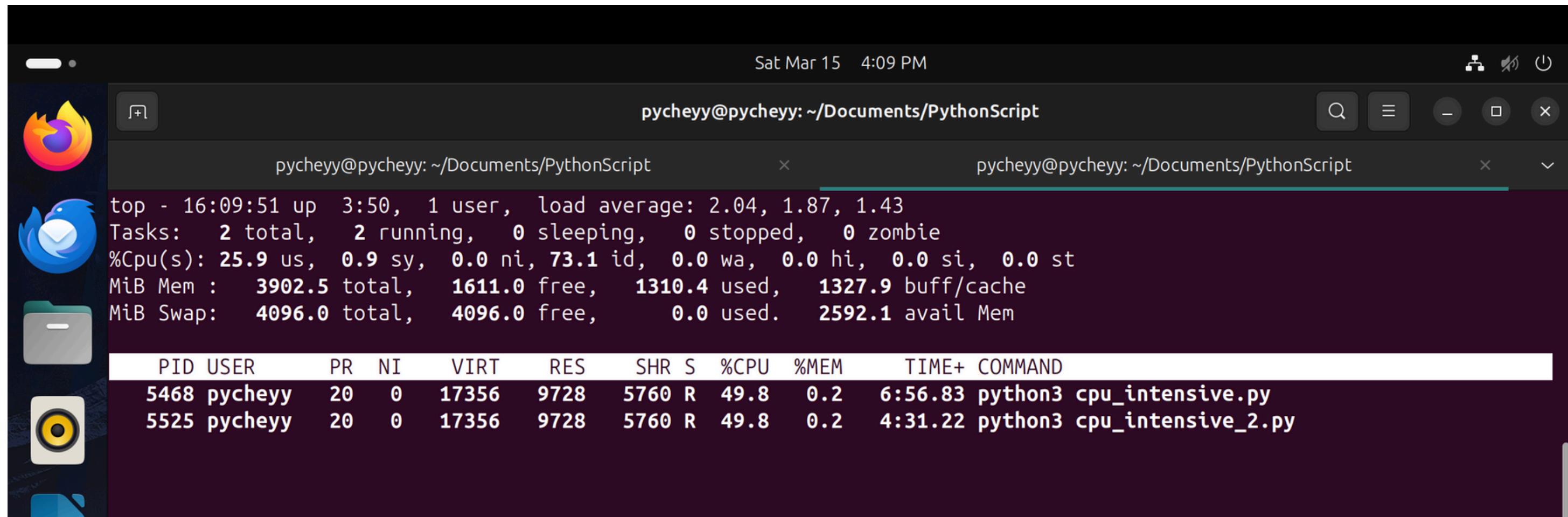
Answer



```
Sat Mar 15 4:03 PM
pycheyy@pycheyy: ~/Documents/PythonScript
pycheyy@pycheyy: ~/Documents/PythonScript
pycheyy@pycheyy:~/Documents/PythonScript$ cp cpu_intensive.py cpu_intensive_2.py
pycheyy@pycheyy:~/Documents/PythonScript$ python3 cpu_intensive_2.py &
[1] 5525
pycheyy@pycheyy:~/Documents/PythonScript$ ps aux | grep cpu_intensive
pycheyy      5468 99.9  0.2 17356  9728 pts/0    R    15:59   2:51 python3 cpu_intensive.py
pycheyy      5525 99.9  0.2 17356  9728 pts/1    R    16:02   0:25 python3 cpu_intensive_2.py
pycheyy      5527  0.0  0.0   6272  2048 pts/1    S+   16:02   0:00 grep --color=auto cpu_intensive
pycheyy@pycheyy:~/Documents/PythonScript$ taskset -pc 0 5468
pid 5468's current affinity list: 0-3
pid 5468's new affinity list: 0
pycheyy@pycheyy:~/Documents/PythonScript$ taskset -pc 0 5525
pid 5525's current affinity list: 0-3
pid 5525's new affinity list: 0
pycheyy@pycheyy:~/Documents/PythonScript$
```

- 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 runs the same CPU core

Answer



Sat Mar 15 4:09 PM

pycheyy@pycheyy: ~/Documents/PythonScript

pycheyy@pycheyy: ~/Documents/PythonScript

pycheyy@pycheyy: ~/Documents/PythonScript

```
top - 16:09:51 up 3:50, 1 user, load average: 2.04, 1.87, 1.43
Tasks: 2 total, 2 running, 0 sleeping, 0 stopped, 0 zombie
%Cpu(s): 25.9 us, 0.9 sy, 0.0 ni, 73.1 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 3902.5 total, 1611.0 free, 1310.4 used, 1327.9 buff/cache
MiB Swap: 4096.0 total, 4096.0 free, 0.0 used. 2592.1 avail Mem

      PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
  5468 pycheyy    20   0  17356  9728  5760 R  49.8  0.2  6:56.83 python3 cpu_intensive.py
  5525 pycheyy    20   0  17356  9728  5760 R  49.8  0.2  4:31.22 python3 cpu_intensive_2.py
```

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

Answer

The screenshot shows a Linux desktop environment with a dark theme. On the left, there is a dock with icons for a browser (Firefox), file manager (Nautilus), system monitor (Dolphin), terminal (Konsole), and help (KHelpCenter). Two terminal windows are open in the background:

- The top terminal window is titled "pycheyy@pycheyy: ~/Documents/PythonScript". It displays the output of the "top" command:

```
top - 16:11:39 up 3:52, 1 user, load average: 2.07, 1.95, 1.51
Tasks: 2 total, 2 running, 0 sleeping, 0 stopped, 0 zombie
%Cpu(s): 23.1 us, 0.1 sy, 1.9 ni, 74.8 id, 0.1 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 3902.5 total, 1610.2 free, 1310.0 used, 1330.2 buff/cache
MiB Swap: 4096.0 total, 4096.0 free, 0.0 used. 2592.6 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
5525	pycheyy	19	-1	17356	9728	5760	R	92.3	0.2	5:40.61	python3 cpu_intensive_2.py
5468	pycheyy	30	10	17356	9728	5760	R	8.0	0.2	7:35.57	python3 cpu_intensive.py
- The bottom terminal window is titled "pycheyy@pycheyy: ~". It shows the user changing the priority of two processes using the "renice" command:

```
pycheyy@pycheyy:~$ renice -n 10 -p 5468
5468 (process ID) old priority 0, new priority 10
pycheyy@pycheyy:~$ sudo renice -n -1 -p 5525
[sudo] password for pycheyy:
5525 (process ID) old priority 0, new priority -1
pycheyy@pycheyy:~$
```

- 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

Sat Mar 15 4:12 PM

pycheyy@pycheyy: ~/Documents/PythonScript

pycheyy@pycheyy: ~/Documents/PythonScript

pycheyy@pycheyy: ~/Documents/PythonScript

```
top - 16:12:36 up 3:53, 1 user, load average: 2.10, 1.98, 1.55
Tasks: 2 total, 2 running, 0 sleeping, 0 stopped, 0 zombie
%Cpu(s): 24.2 us, 0.2 sy, 0.8 ni, 74.8 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 3902.5 total, 1596.1 free, 1324.0 used, 1336.4 buff/cache
MiB Swap: 4096.0 total, 4096.0 free, 0.0 used. 2578.6 avail Mem

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
5468 pycheyy 10 -10 17356 9728 5760 R 96.7 0.2 8:00.63 python3 cpu_intensive.py
5525 pycheyy 25 5 17356 9728 5760 R 3.3 0.2 6:12.59 python3 cpu_intensive_2.py
```

pycheyy@pycheyy:~\$ sudo renice -n -10 -p 5468
5468 (process ID) old priority 10, new priority -10
pycheyy@pycheyy:~\$ renice -n 5 -p 5525
5525 (process ID) old priority -1, new priority 5
pycheyy@pycheyy:~\$ █

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

Task 3

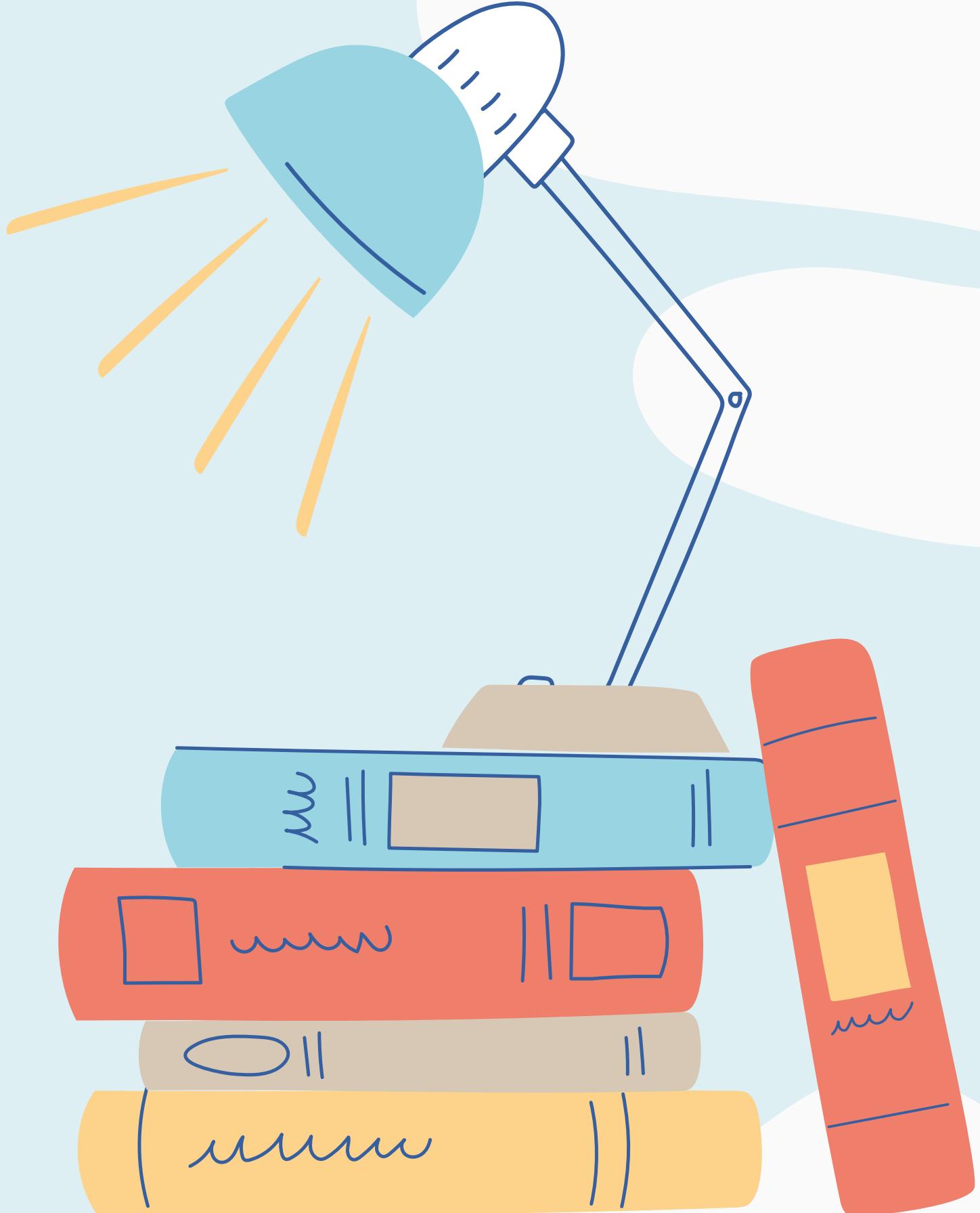
Task 3: Process Termination

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 difference 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

The screenshot shows a Linux desktop environment with two terminal windows. The top terminal window has a dark background and displays the following session:

```
Sat Mar 15 4:21 PM
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$
```

A red arrow points from the '^C' interrupt to the text 'SIGTERM -15'. Another red arrow points from the '^C' interrupt to the text 'SIGKILL -9'.

The bottom terminal window also has a dark background and displays:

```
pycheyy@pycheyy:~$ kill -15 5695
pycheyy@pycheyy:~$ kill -9 5723
pycheyy@pycheyy:~$
```

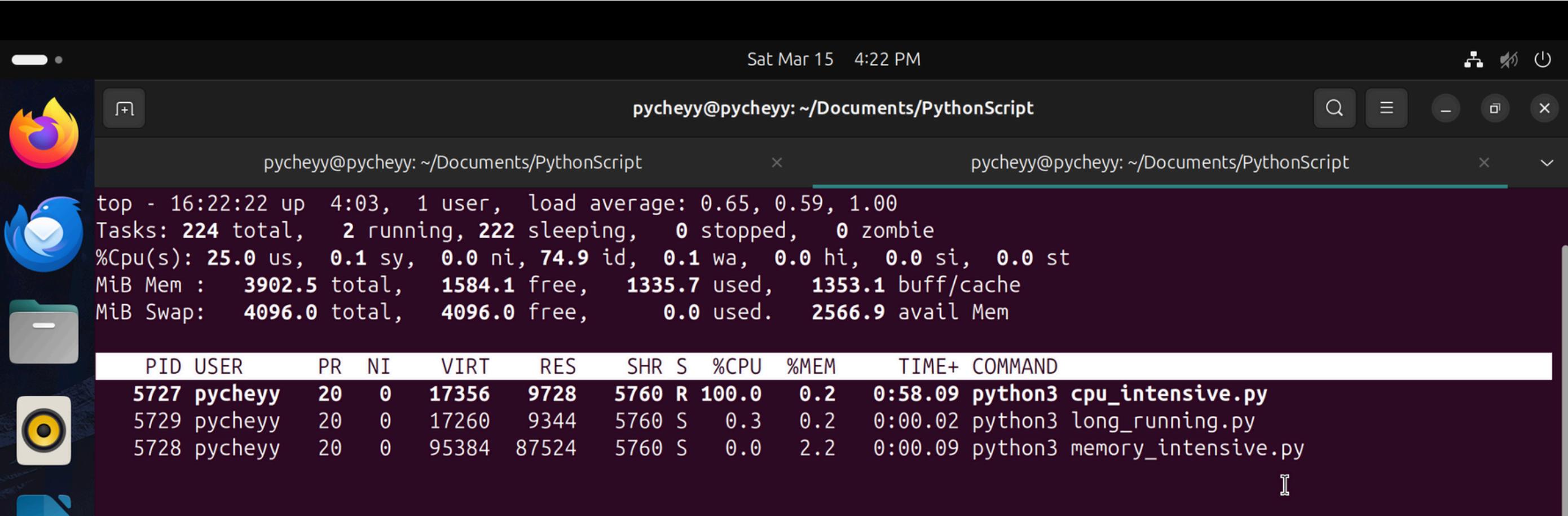
Task 4

Task 4: Resource Management

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



Order by CPU%



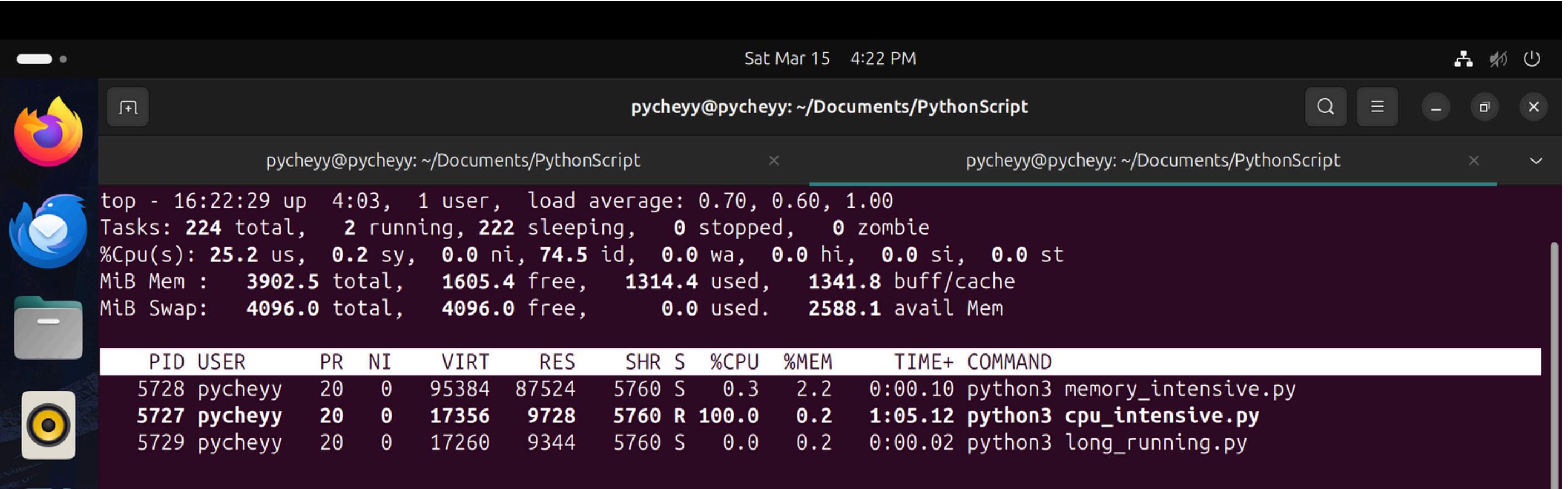
```
Sat Mar 15 4:22 PM
pycheyy@pycheyy: ~/Documents/PythonScript
pycheyy@pycheyy: ~/Documents/PythonScript
pycheyy@pycheyy: ~/Documents/PythonScript

top - 16:22:22 up 4:03, 1 user, load average: 0.65, 0.59, 1.00
Tasks: 224 total, 2 running, 222 sleeping, 0 stopped, 0 zombie
%Cpu(s): 25.0 us, 0.1 sy, 0.0 ni, 74.9 id, 0.1 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 3902.5 total, 1584.1 free, 1335.7 used, 1353.1 buff/cache
MiB Swap: 4096.0 total, 4096.0 free, 0.0 used. 2566.9 avail Mem

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
5727 pycheyy 20 0 17356 9728 5760 R 100.0 0.2 0:58.09 python3 cpu_intensive.py
5729 pycheyy 20 0 17260 9344 5760 S 0.3 0.2 0:00.02 python3 long_running.py
5728 pycheyy 20 0 95384 87524 5760 S 0.0 2.2 0:00.09 python3 memory_intensive.py
```

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

Order by MEM%

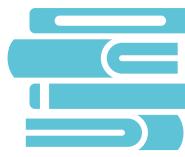


```
Sat Mar 15 4:22 PM
pycheyy@pycheyy: ~/Documents/PythonScript
pycheyy@pycheyy: ~/Documents/PythonScript
pycheyy@pycheyy: ~/Documents/PythonScript

top - 16:22:29 up 4:03, 1 user, load average: 0.70, 0.60, 1.00
Tasks: 224 total, 2 running, 222 sleeping, 0 stopped, 0 zombie
%Cpu(s): 25.2 us, 0.2 sy, 0.0 ni, 74.5 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 3902.5 total, 1605.4 free, 1314.4 used, 1341.8 buff/cache
MiB Swap: 4096.0 total, 4096.0 free, 0.0 used. 2588.1 avail Mem

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
5728 pycheyy 20 0 95384 87524 5760 S 0.3 2.2 0:00.10 python3 memory_intensive.py
5727 pycheyy 20 0 17356 9728 5760 R 100.0 0.2 1:05.12 python3 cpu_intensive.py
5729 pycheyy 20 0 17260 9344 5760 S 0.0 0.2 0:00.02 python3 long_running.py
```

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



Thank you



Prak Pychey



prakpycheyy@gmail.com

Resources

