COMP2812 Systems Software

**Task 01 Report**

# Knowledge Base Entry

## Title:

How to Identify and Resolve “High CPU Usage by a Background Service in Linux”

## Introduction:

This article explains how to find and stop a background process that is using a lot of CPU power in a Linux system. This is important because high CPU usage can make the computer very slow.

## Steps to Identify the Problem:

1. First, we ran the command `**top**` to see which process was using a lot of CPU.

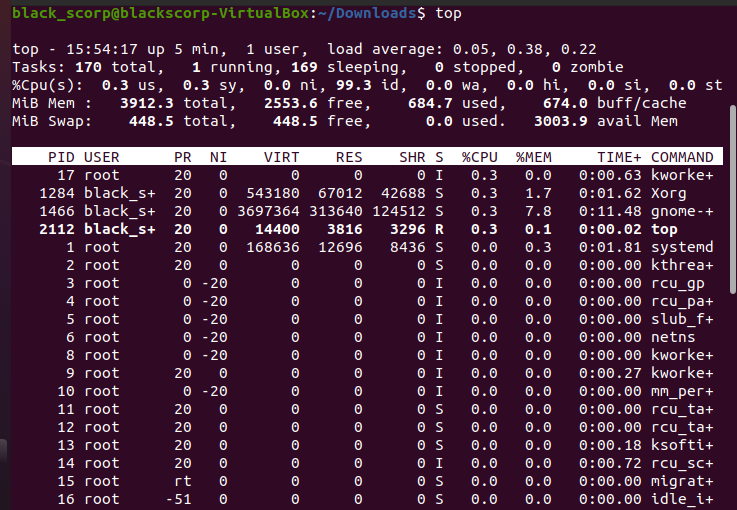


Figure 01 - System with normal state

2. Before such an issue occurs, no process was using too much CPU. (See Figure 01)

3. Then we ran the task using `/opt/tasks/task1.sh`. And that creates a CPU consuming task.

4. After running the task, a process named `sneaky+` started using around 95% of the CPU. (See Figure 02)

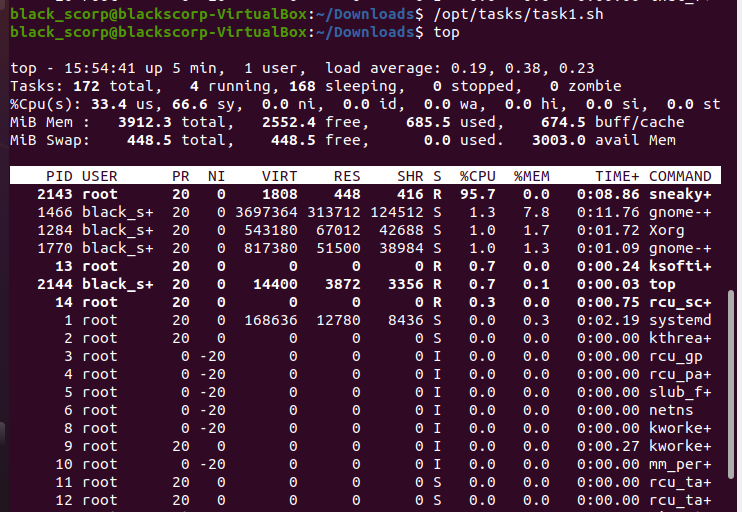


Figure 02 - System with high CPU consumption

## Diagnosis:

We found that the `sneaky+` process was the one using too much CPU. Using its PID, we tried to find the service behind it using `**systemctl status <PID>**` and `**grep**` commands.

## Steps to Fix:

1. After identifying the related systemd service, we stopped it using `**sudo systemctl stop sneaky.service**`.
2. We also disabled it using `**sudo systemctl disable sneaky.service**` to **prevent it from running again.**
3. Finally, we rebooted the system using **sudo reboot** and checked with `**top**` to confirm the problem was solved.

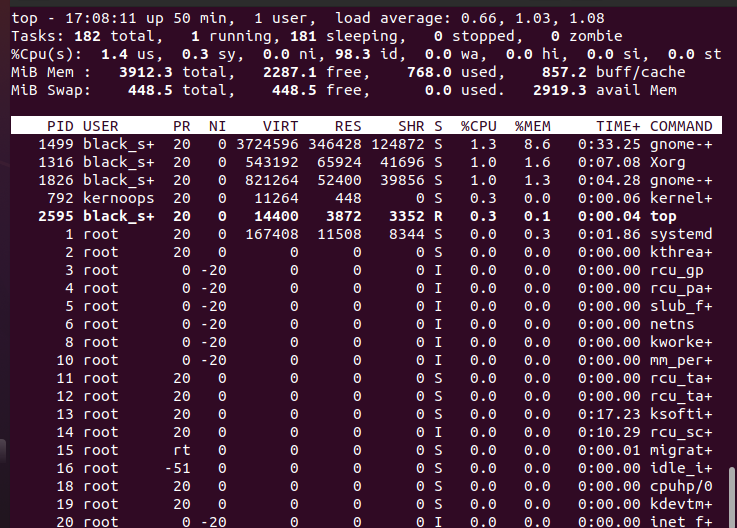


Figure 03 - After resolving the issue

For a Quick solution, we can try **sudo kill -9 <PID>** as well, but it will not prevent the restarting.

## Conclusion:

This method is a simple and effective way to handle background processes that slow down the system.  
Figure 1: System state before running task 1, showing normal CPU usage.  
Figure 2: System state after running task 1, showing high CPU usage by `sneaky+`.

# Theoretical Reflection

In this task, we learned how CPU usage affects system performance. The CPU does all the main processing tasks in a computer. If one program uses most of the CPU, then other programs can become slow. This is what happened with the `sneaky+` process. It was using around 95% of the CPU, so the rest of the system felt very slow.  
A CPU has to handle many tasks. It switches between them using context switching and that makes us feel all are working at the same time but in correct terms, if we only have one CPU with us. It does one thing at a snapshot and then it switch the context and get to another process. If a program uses too much CPU, the scheduler (which manages how tasks are run) gets overloaded. This limits the performance. When a process is called 'CPU-bound', it means it always wants to use the CPU and doesn't wait for anything else. That’s what we saw here.

So in any place where we see a slow down of a system. We can use to check the processes which are consuming much CPU, Memory etc using Top command on Linux systems, in Windows we have a GUI option named task manager to do this and stop any task we want to stop at there itself.

When the system is CPU-bound, you also get resource contention. That means many processes want to use the same resource (CPU), but only one can use it at a time. This causes delay for others. Response time goes up. For example, even opening the file manager or terminal took extra time. That’s why identifying such programs and stopping them is important for system health.

In real systems, this problem can come from faulty scripts or background services. So this task taught us how to observe, analyze, and fix such issues in a simple Linux environment.

In short. If we find a slow down. We can check the processes that consumes much CPU and Memory and so on and Kill them if not mandatory to run. That itself releases much of the resources and other processes can utilize the released resources.

* **I acknowledge the use of ChatGPT in paraphrasing original content (OpenAI, 2025). The prompts and output from ChatGPT are included in Appendix 1**

**Prompts of Task 01**

“I have this linux assignment. So I'm running this inside virtual box with ubuntu. I checked using top command and system seems fine, no process using much CPU.

But then I run the /opt/tasks/task1.sh as it says in instructions. After that, a process called sneaky+ is using like 95% CPU. So it's consuming a lot, and system becomes slow.

I think this is the one I have to fix for Task 1. But I don’t know what to do after that. I mean what commands should I use to find what is causing it and how to stop it permanently.

Can you please give me all necessary steps or commands to fix this sneaky process such as sudo systemctl stop sneaky.service? Also how to find if it’s coming from a service or something. Thanks.”

**Response**

“To fix the sneaky+ process consuming excessive CPU, follow these steps:

Check the process details:

First, get the details of the sneaky+ process. You can use the ps command to check which executable it’s running from.

bash

Copy

ps aux | grep sneaky+

This will slow you the full command and the parent process of sneaky+.

Identify the parent process:

The process might be spawned by a parent process. You can use pstree or top to find the parent.

bash

Copy

pstree -p | grep sneaky+

Or using top to view the parent PID (PPID) and locate it more clearly.

Find the associated service:

If sneaky+ is part of a service, it might be linked to a systemd service or a script running as a daemon. To check if it’s controlled by systemd:

bash

Copy

systemctl list-units --type=service | grep sneaky

If you find it is a service, use the following command to stop and disable it:

bash

Copy

sudo systemctl stop sneaky.service

sudo systemctl disable sneaky.service

Check for cron jobs:

If it’s not part of a service, it might be run periodically by a cron job. Check the cron jobs for root and other users:

bash

Copy

sudo crontab -l

sudo crontab -u root -l

cat /etc/crontab

Check for scheduled jobs:

If it’s not part of the systemd or cron jobs, it could be scheduled in another way (e.g., a script in ~/.bashrc, ~/.profile, or ~/.bash\_profile). Check those files for anything suspicious.

Kill the process (if necessary):

If you want to stop the process immediately without disabling it permanently, use kill:

bash

Copy

sudo kill -9 <PID\_of\_sneaky+>

Permanent removal (if required):

If the process is being started by some script or service that you want to remove permanently, delete the associated script, or disable/remove the service. Check the /opt/tasks/task1.sh file to see if it’s creating or invoking the process, and modify it if necessary.

By following these steps, you should be able to identify and stop the sneaky+ process from consuming excessive CPU and ensure it doesn't restart.”