Module 2: Critical Thinking

Ryan Thompson

Colorado State University - Global

CSC 507

Dr. Farr

17 October 2024

**Introduction**

Process management utilities are used to perform various tasks related to controlling, monitoring, and interacting with running processes. Every operating system provides support for their own type of process management tools. Windows will normally have their GUI based Task Manager, whereas Linux tends to lean toward terminal based process management tools such as htop. Both process management tools provide similar features and functionality. Therefore either one can be used to display hardware resource usage, view running processes, and ending problematic processes, etc.

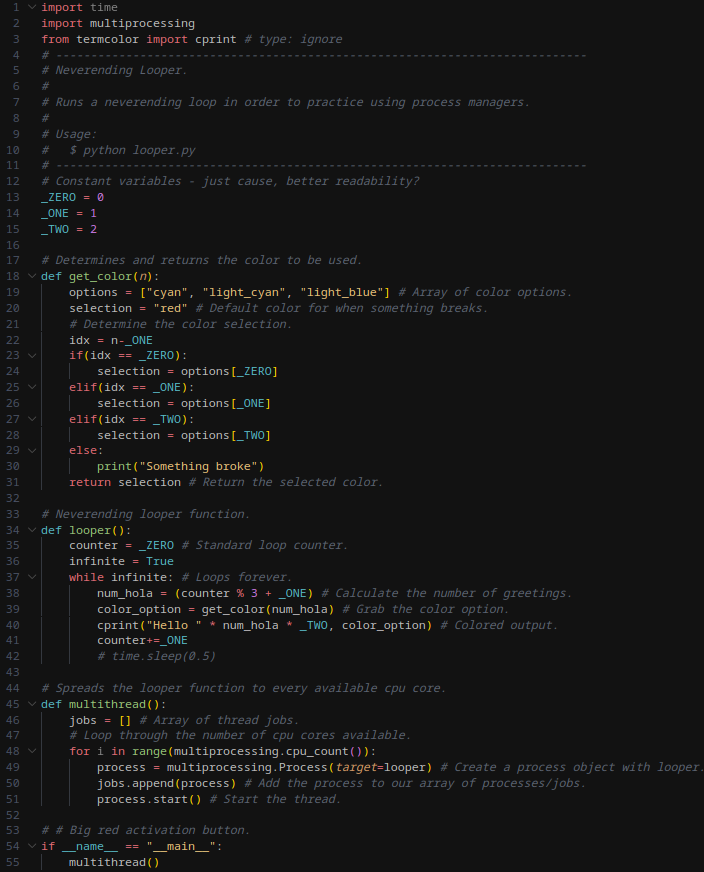
**Methodology**

In this instance we will be running a python script with a never ending loop in order to witness the load it has on the computer’s resources. This will be witnessed by locating the process within a process manager. Then we will use the process manager to end the process, thus stopping the never ending loop from hogging all the resources. The entire process will be completed both in Windows 10 and Linux Debian 12 operating systems.

To conduct this experiment, five steps will be executed in both operating systems. The first step will be to analyze the starting resource usage with a process management tool. Next the Python script will be started, at which point the processes will be located within the process management tool. While the Python script is still running its never ending loop, the resource usage will be reanalyzed for any changes. Once the changes have been documented the processes will be terminated using the process management tool. Finally, once all the desired processes have been ended, a final analysis of the resource usage will be conducted.

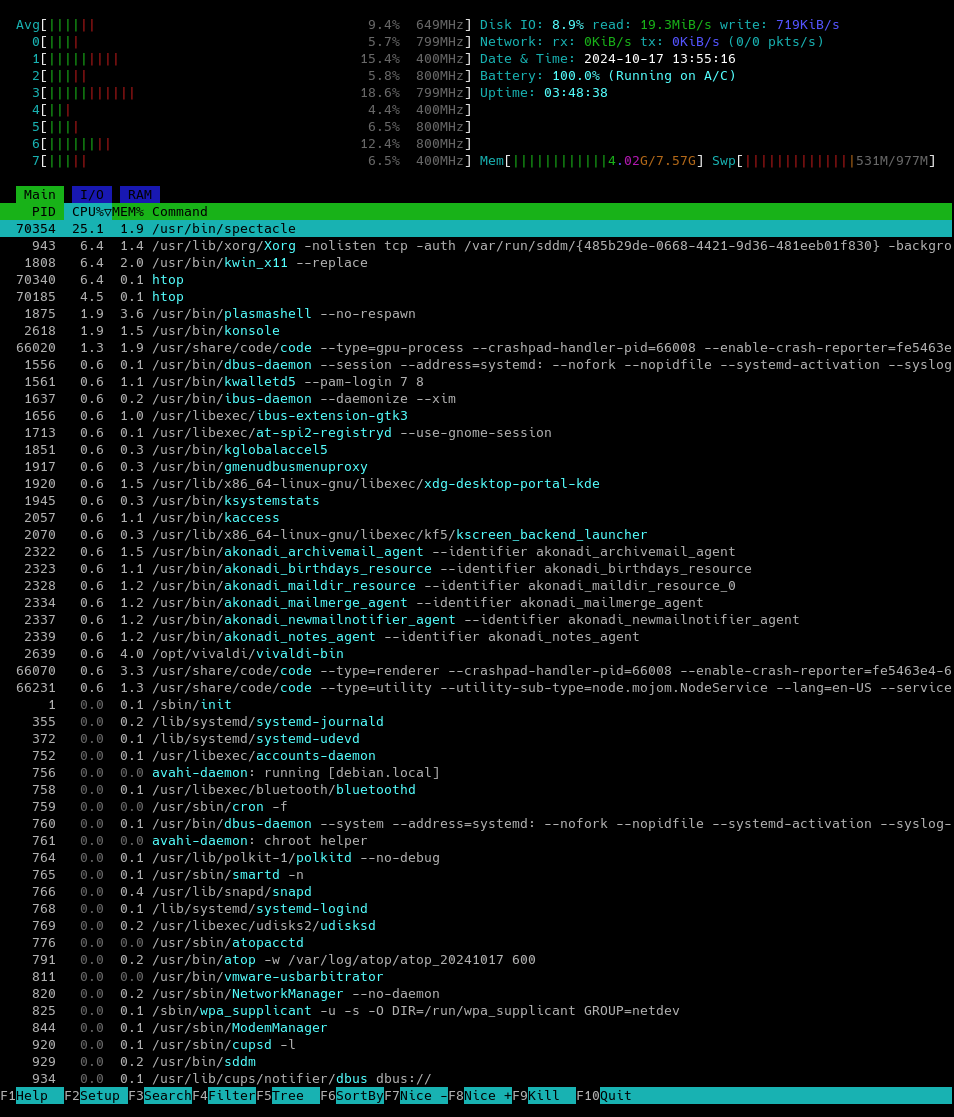
**Experiment**

In the first screenshot listed below, the Python script to run a never ending loop can be viewed. The script makes use of all the available CPU cores in order to hopefully see how the processes can impact hardware resources.



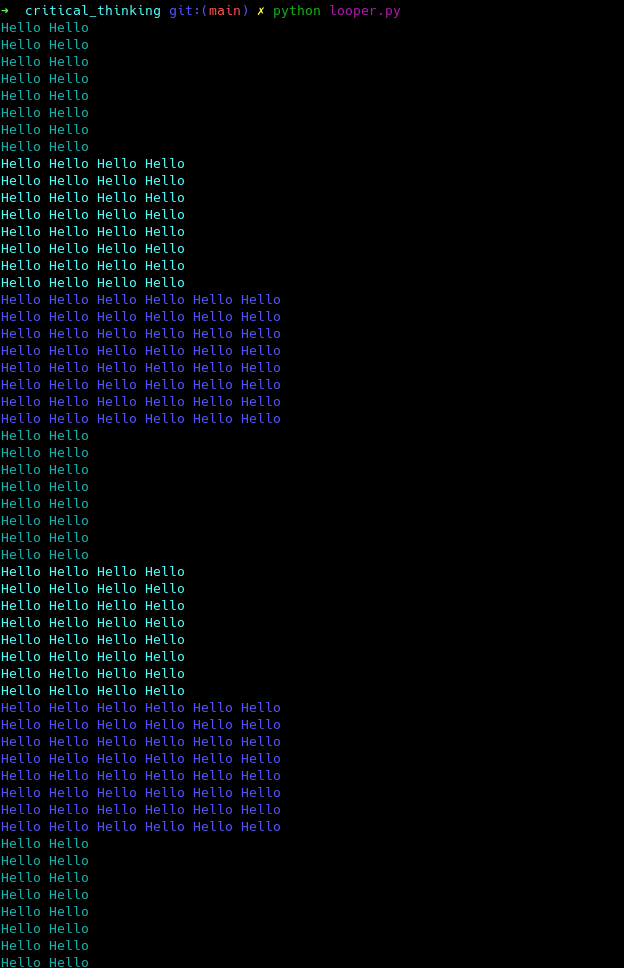
*Python source code*

**Linux Debian 12**

To start the experiment with Linux, the htop process management tool is used to analyze the computer’s baseline resource usage. Within htop, the processes will be sorted by CPU usage in descending order. As you can see in the screenshot below the CPU usage is relatively low. Specifically, the average CPU usage is at 9.4%.

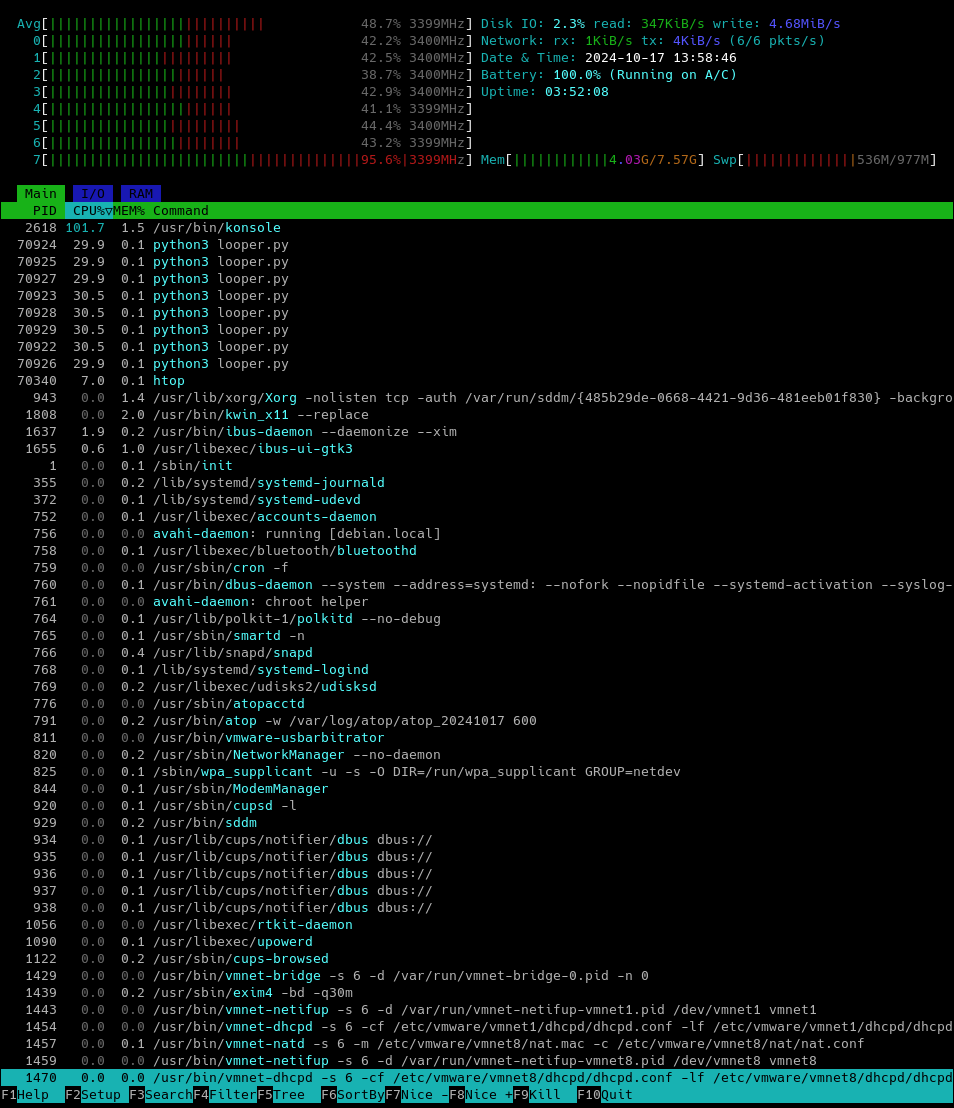
*Initial htop results*

Here the Python program is started in order to run its never ending loops in a multithreaded fashion. Now we will use the process management tool, htop*,* again to witness how the resource usage has changed.



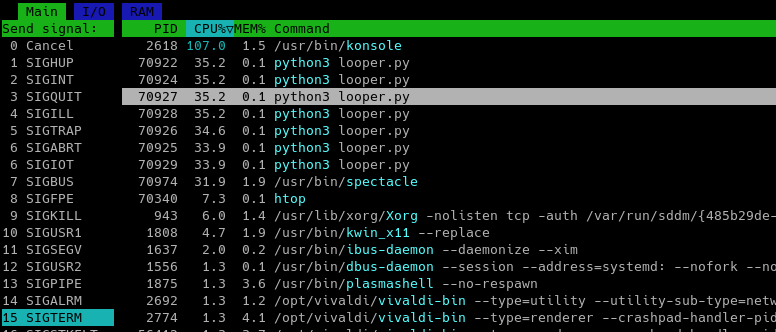
*Python script output*

Here a significant increase in the resource usage can be seen. The average CPU usage has shot up to 48.7%. Furthermore, you can see each thread running our script via the command *python3 looper.py* is taking up a large chunk of resources*.*



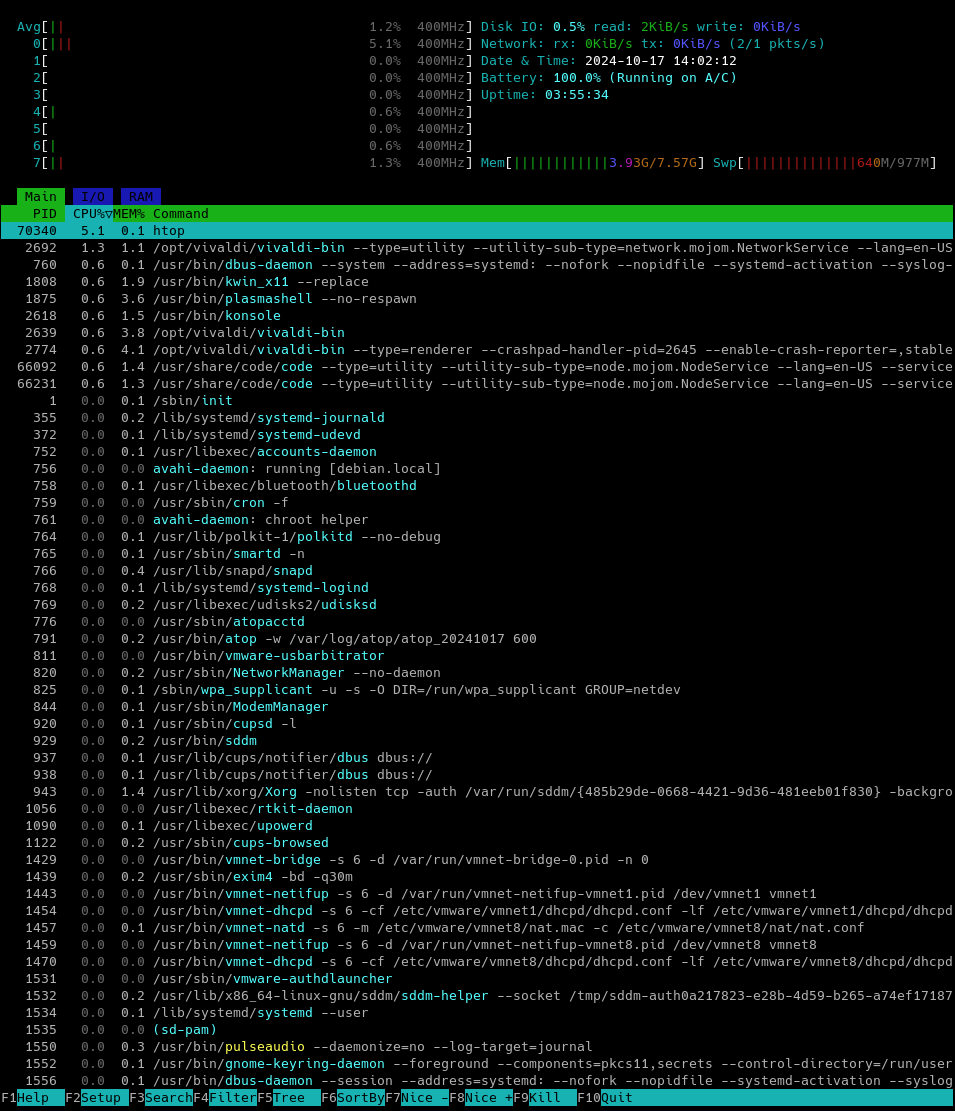
*Htop output while running the Python script.*

In this step the, htop tool will be used to end all the processes running the command *python3 looper.py*. This is done by selecting the desired process and pressing F9 to send a kill command. Since the processes are sorted by CPU usage, the processes in question are all conveniently located towards the top. If the processes were not sorted, one could also try using the search feature by pressing F3.



*Sending the kill command to the desired processes*

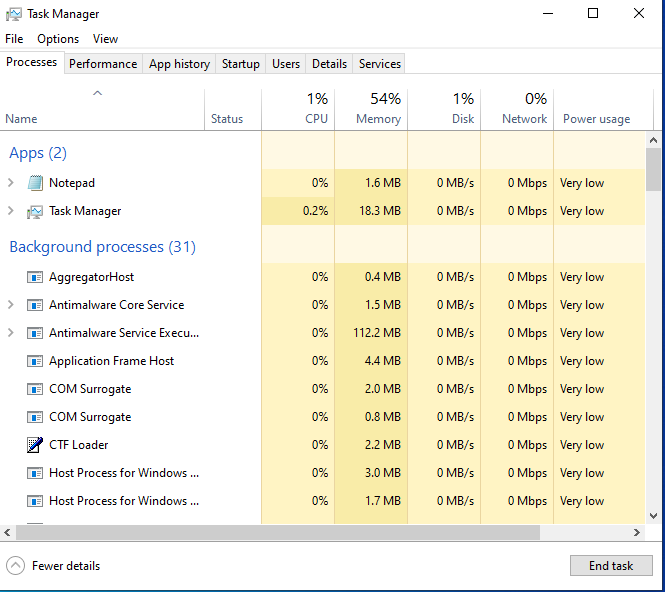
After sending the kill command to the processes running the never ending loop, the processes have ended and the resource usage has significantly decreased. In the screenshot below, the average CPU usage is down to 1.2%.



*Htop output after ending all desired processes*

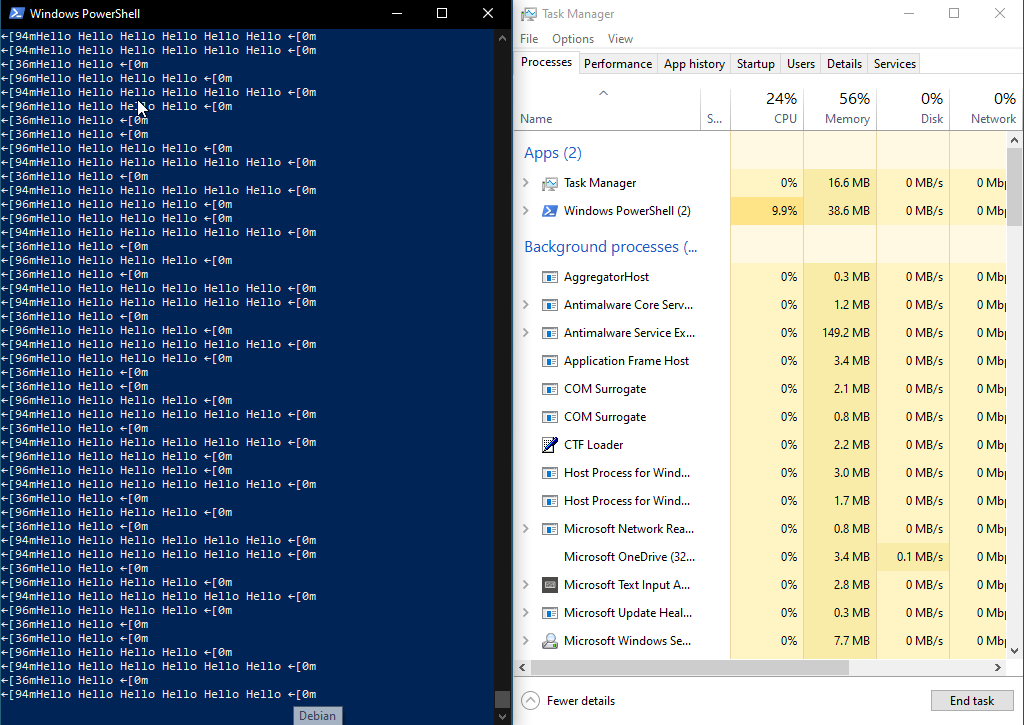
**Windows 10**

To start the experiment with Windows, the process management utility, Task Manager, will be used to get the computer’s baseline resource usage. Here the CPU usage can be seen starting at 1%. This will be compared to the results when the Python script is executed.



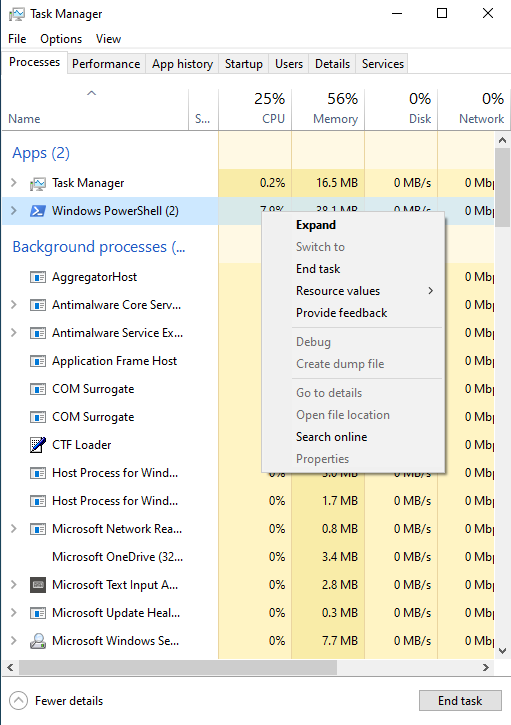
*Task Manager - Starting point*

In this section the Python script can be seen executing within a Windows PowerShell terminal. While running the script, the resource usage can be seen to increase. The CPU in particular has had a significant increase, while the RAM has had a mild increase. The CPU increased from 1% to 24% simply by running the Python script.



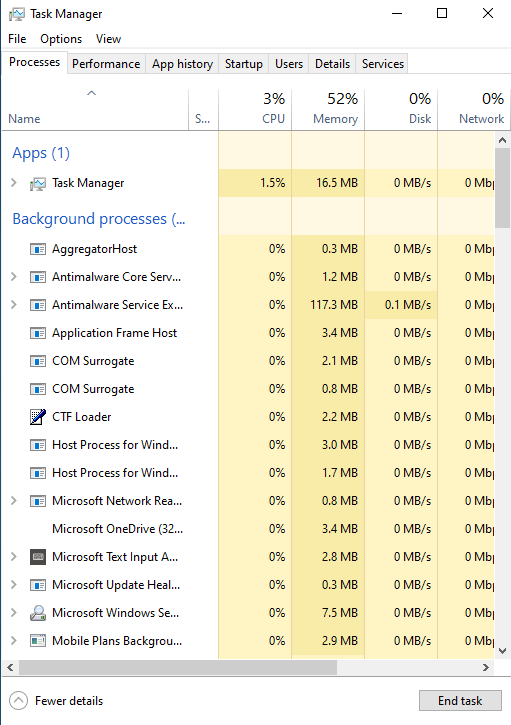
*Windows’ PowerShell & Task Manager*

Now that the resource usage has been analyzed, the task or processes need to be ended. This is done by simply clicking on the desired task and clicking the “end task” option. After ending the desired task(s), the Task Manager will be used again to analyze the hardware resources.



*Using the Task Manager to end processes*

Finally, the desired processes have been terminated and the resource manager, Task Manager, is showing the PowerShell process is gone and the resource usage has gone down significantly. Specifically, the CPU usage has dropped down to 3%, after ending the processes.



*Final Task Manager results*

**Conclusion**

While both operating systems perform similarly when given the same task, there are some minor differences to consider. For example, Windows is more of a GUI based system whereas Linux has more of a mixture of GUIs and terminal based tools. During this experiment Windows’ Task Manager competed against Linux’s htop process manager. Even though they both performed roughly the same, I found myself more drawn to Linux’s htop. I preferred htop because it provided more information on the processes as well as offering more customization. Overall, both are good operating systems taking slightly different approaches towards similar goals.

**References**

Grob, S. (2016). An introduction to htop: A dynamic task manager for Linux. Journal of Open Source Software, 1(3), 56. https://doi.org/10.21105/joss.00056

Microsoft. (2021). Task Manager in Windows 10: A guide to system performance monitoring. Microsoft Documentation. https://support.microsoft.com/en-us/help/12445/windows-task-manager-what-it-is-and-how-to-use-it

Simpson, D. (2020). Comparing system monitoring tools in Linux and Windows: htop vs. Task Manager. International Journal of Computer Science & Information Technology, 12(4), 99-105. https://doi.org/10.1016/j.ijcsit.2020.04.010