Andrew Plum

Professor Wilder

CS 240

9/30/22

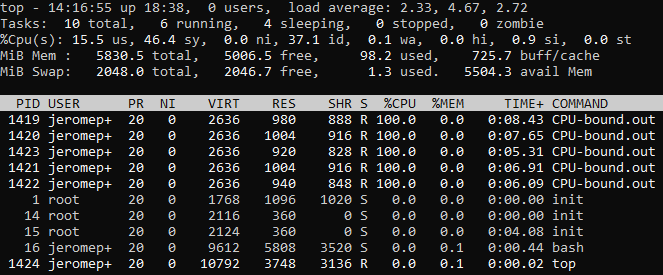
Homework #2 Report

Answers to questions where the processes ran on my local computer (4 cores):

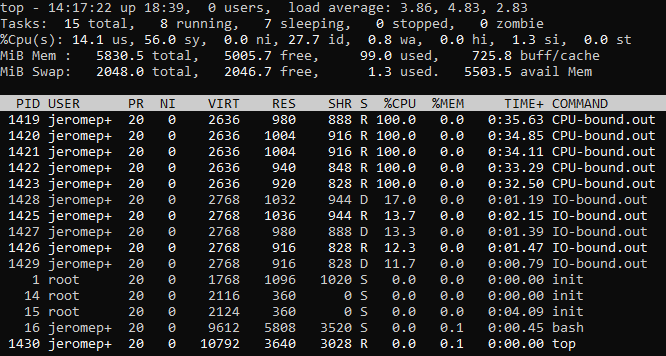
1. a) What I noticed about the CPU-bound processes is that all of them had the same process priority and same CPU utilization of 100%. b) I ran 5 instances of the CPU-bound program, and I observed a CPU utilization of 100% for all 5 processes.
2. a) When observing the mix of processes, I noticed all 5 of the CPU-bound processes had 100% of CPU utilization, and that the IO bound processes had significantly less CPU utilization despite having the same process priority. In screenshot 2, the CPU-utilization ranged from 11.7% to 17.0% and they were relatively the same in screenshot 3. When comparing screenshots 2 and 3, you can also see that the CPU-bound processes were experiencing significantly more CPU time than the IO-bound processes. b) I ran 5 instances of the IO-bound program and observed a CPU utilization that ranged from 11.7% to 17.0% in screenshot 2 and 11.3% and 15.0% in screenshot 3.

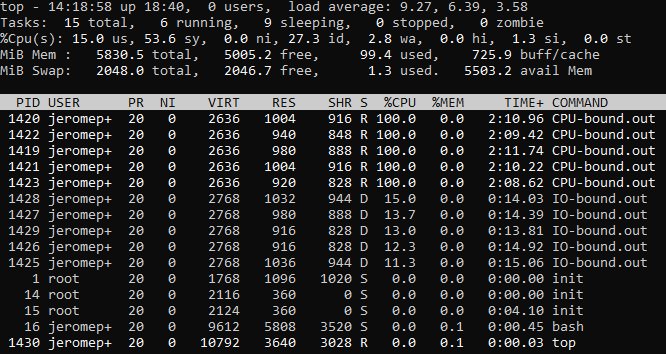
Answers to questions where the processes ran on the computer science course servers (cs240a):

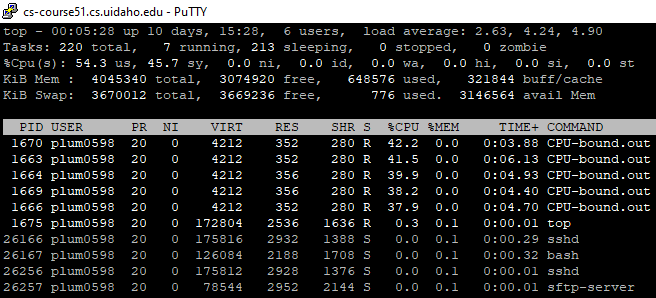
1. a) What I noticed about the CPU-bound processes when they were ran on the course servers was that they all had lower CPU utilizations than the CPU-bound processes on my local computer. b) I ran 5 instances of the CPU-bound program, and I observed a CPU utilization that ranged from 37.9% to 42.2%. This information can be seen in screenshot 4.
2. a) When observing the mix of processes, I noticed all 5 of the CPU-bound processes had a higher percent of CPU utilization than the IO-bound processes even though both kinds of processes had the same process priority. b) I ran 5 instances of the IO-bound program and observed a CPU utilization of anywhere from 0% to 0.7% while the CPU-bound processes were running concurrently and a CPU utilization from 1.3% to 1.7% when the IO-bound processes were running without the CPU-bound processes. The CPU-bound processes significantly decreased the efficiency of the IO-bound processes, but the IO-bound processes were slow regardless.

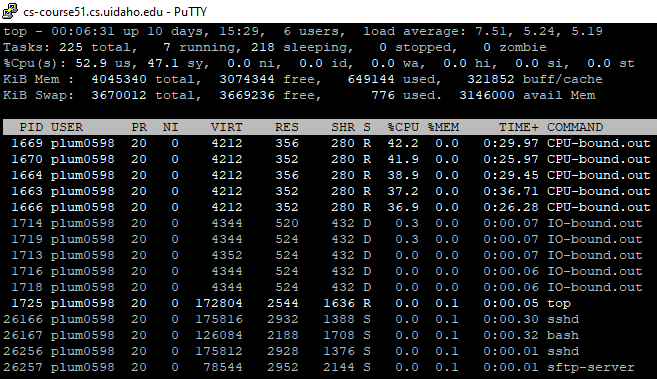
Screenshots of processes ran on my local computer (4 cores):

Screenshot 1: This screenshot was taken at the beginning of the time frame of running the processes. There are 5 CPU-bound processes running on my local computer. Notice all 5 CPU-bound processes have a CPU utilization of 100%.

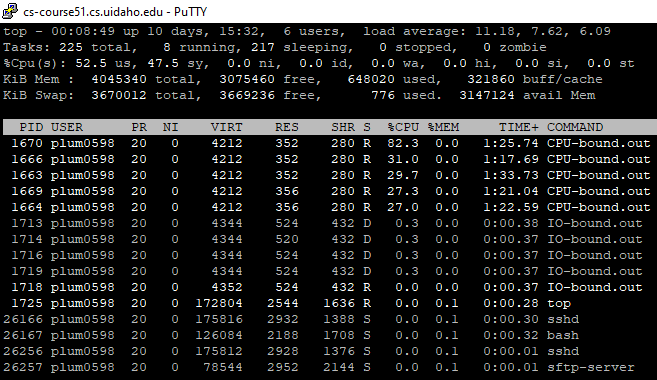
Screenshot 2: This screenshot was taken after the IO-bound processes were started. There are 5 CPU-bound processes and 5 IO-bound processes running on my local computer. This screenshot is taken relatively early in the time frame when they were ran. The CPU utilization is significantly higher for the CPU-bound processes than for the IO-bound processes.

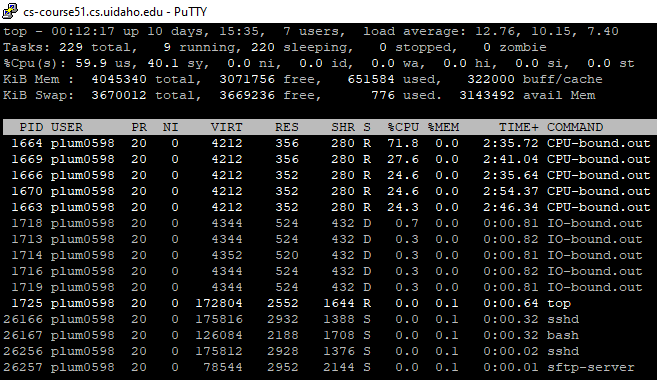
Screenshot 3: There are 5 CPU-bound processes and 5 IO-bound processes running on my local computer. This screenshot is taken relatively later than the last screenshot in the time frame that they were ran. Notice that since screenshot 2, the CPU time for the CPU-bound processes has increased significantly more than the CPU time for the IO-bound process. The CPU utilization is still significantly higher for the CPU-bound processes than for the IO-bound processes.

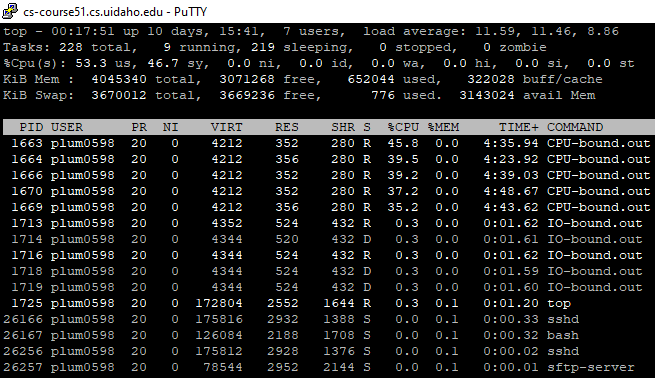
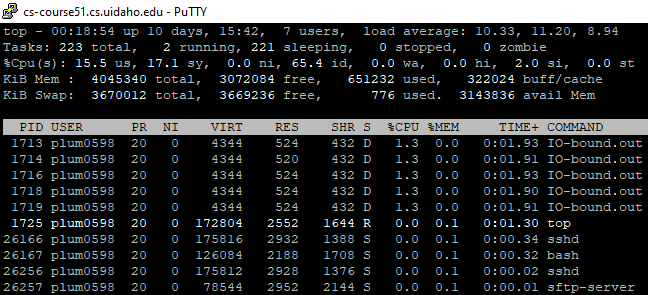
Screenshots of processes ran on computer science course servers (cs240a):

Screenshot 4: 5 CPU-bound processes running on the course servers. This screenshot of top was near the beginning of running the processes. The CPU utilization was similar among the 5 processes and ranged from 37.9% to 42.2%.

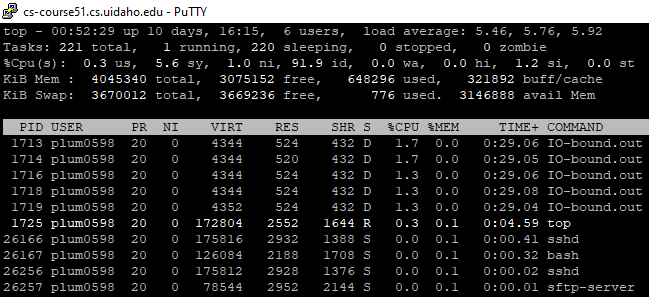
Screenshot 5: This was still early in the time frame of running the processes. The 5 CPU-bound processes are still running in addition to 5 IO-bound processes. The CPU utilization was very low for the IO-bound processes. This was likely due to the wait times related to IO access.

Screenshot 6: This screenshot was taken halfway between the beginning and the middle of the time frame. Sometimes one of the processes’ CPU utilization would increase dramatically. Here you can see the process with the PID 1670 have a CPU utilization 82.3%.

Screenshot 7: This screenshot was taken nearly at the halfway point of the time frame of running the processes. Notice the CPU time for the CPU bound processes has increased significantly more than the CPU time for the IO-bound processes since screenshot 5. The IO-bound processes have had significantly less CPU time due to the wait times associated with IO access. You may notice there was also another dramatic increase in the CPU utilization for the CPU-bound process with the PID 1664.

Screenshot 8: This screenshot was taken nearly at the end of the time frame of the CPU-bound processes running. It is still the case that the IO-bound processes have had significantly less CPU time than the CPU-bound processes.

Screenshot 9: This screenshot was taken immediately after the CPU-bound processes finished. Here you can see the CPU utilization of the IO-bound processes increase significantly relative to what they were previously in screenshot 8.

Screenshot 10: This screenshot was taken relatively soon after screenshot 9. Here the CPU time has increased significantly compared to the time frame before screenshot 8 when both the CPU-bound processes and the IO-bound processes were both running. Even though the CPU utilization isn’t super high, because it was higher than what it was previously, the IO-bound processes have had a significant increase in CPU time.

Conclusion:

IO-bound processes were significantly more inefficient with CPU utilization than the CPU-bound processes. This is because when you introduce any sort of IO instructions into a program, the processor has very long wait times before it can process things again because the IO devices are simply that much slower than the processor. It is also observable that the CPU-bound processes also significantly decrease the efficiency of the IO-bound processes, but the IO-bound processes were slow regardless.

Because the IO-bound processors were more inefficient with CPU utilization than the CPU-bound processors, I can conclude IO access limits a program’s ability to fully utilize the CPU. This implies that if you want to increase the efficiency of the programs you write, then you should try to limit the number of IO instructions used in the program and only use them when absolutely necessary. Although this experiment does not examine this, it would be good to explore whether IO operations are as efficient when used in different ways. For example, would it be just as efficient to leave a file open and then close it once fully done with it in the program as it would be to constantly open and close a file every time you make some change to it? I am going to try to test this on my own to see if there is a significant difference.

Overall, this experiment was very useful, as it has led to an increased understanding of how computational operations differ from IO operations. I also learned what a makefile was and how it works (I was never introduced to one previously), and I figured out how to use the windows subsystem for Linux so that I could run the programs I wrote locally on my computer. It has also raised some interesting questions for me to explore in the future.