Alejandro Paz
Parallel Computing

Map Reduce

What problems did you encounter completing the assignment and how did you overcome them?

My first issue was figuring out how to properly iterate over the files first. I overcame this by using p.iterate in a for loop. One more issue I ran into that I was able to fix quickly was I forgot to release the lock after acquiring it.

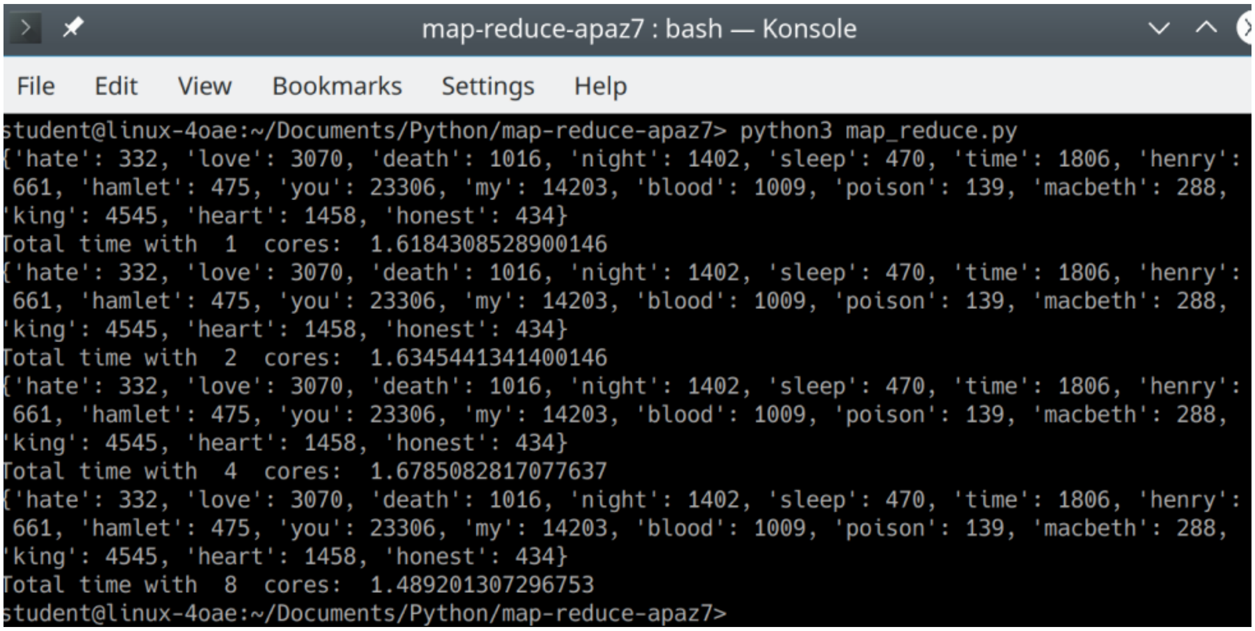
Any problems you weren't able to overcome or any bugs still left in the program?

None that I noticed.

About how long it took you to complete the assignment

This project took me about 2 days to fully finish. The first day was to simply get an idea of how I wanted to go about completing the project while the second day was the implementation of it.

Performance measurements (given in seconds) for 1, 2, 4, and 8 threads



```
student@linux-4oae:~/Documents/Python/map-reduce-apaz7> python3 map_reduce.py
{'hate': 332, 'love': 3070, 'death': 1016, 'night': 1402, 'sleep': 470, 'time': 1806, 'henry':
 661, 'hamlet': 475, 'you': 23306, 'my': 14203, 'blood': 1009, 'poison': 139, 'macbeth': 288,
'king': 4545, 'heart': 1458, 'honest': 434}
Total time with  1  cores:  1.6184308528900146
{'hate': 332, 'love': 3070, 'death': 1016, 'night': 1402, 'sleep': 470, 'time': 1806, 'henry':
 661, 'hamlet': 475, 'you': 23306, 'my': 14203, 'blood': 1009, 'poison': 139, 'macbeth': 288,
'king': 4545, 'heart': 1458, 'honest': 434}
Total time with  2  cores:  1.6345441341400146
{'hate': 332, 'love': 3070, 'death': 1016, 'night': 1402, 'sleep': 470, 'time': 1806, 'henry':
 661, 'hamlet': 475, 'you': 23306, 'my': 14203, 'blood': 1009, 'poison': 139, 'macbeth': 288,
'king': 4545, 'heart': 1458, 'honest': 434}
Total time with  4  cores:  1.6785082817077637
{'hate': 332, 'love': 3070, 'death': 1016, 'night': 1402, 'sleep': 470, 'time': 1806, 'henry':
 661, 'hamlet': 475, 'you': 23306, 'my': 14203, 'blood': 1009, 'poison': 139, 'macbeth': 288,
'king': 4545, 'heart': 1458, 'honest': 434}
Total time with  8  cores:  1.489201307296753
student@linux-4oae:~/Documents/Python/map-reduce-apaz7>
```

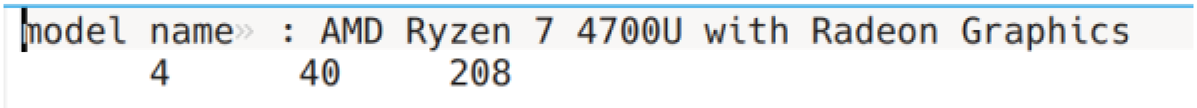
A short analysis of why the program behaves as it does with an increasing number of threads

The more threads you have all working to accomplish a task then faster your program will complete. With only one thread in play, finishing the count of words across multiple files will take longer than when you have multiple threads that could all count the number of words at the same time.

Any observations or comments you had while doing the assignment

It seems that the number of threads used doesn’t really have much of a difference regarding the time it takes to count the words across files. With 1, 2, and 4 threads the times are very similar and with 8 the time gets a little bit better but not much. I wonder if this is due to my implementation of the solution.

Output from the cpuInfoDump.sh program



```
model name	: AMD Ryzen 7 4700U with Radeon Graphics
      4      40     208
```