<u>Write up</u>
By Kai Richardson
1. Did you complete all required sections and assignment's functional requirements. If not state what is successfully implemented and what is not.
   Yep, all implemented. In Rust, that is.
2. Measure the performance of you code with varying number of threads. Measure the execution time and report it here. Use the "time" utility to do this task.

Threads	Run Time	Comments and Notes
1	real 0m39.435s user 0m23.801s sys 0m1.414s	
2	real 0m22.662s user 0m38.117s sys 0m2.276s	
4	real 0m12.983s user 0m42.078s sys 0m2.431s	
8	real 0m8.894s user 0m46.278s sys 0m4.033s	Most performant!
16	real 0m9.393s user 0m46.612s sys 0m3.981s	Everything from this point only gets higher
32	real 0m9.919s user 0m49.441s sys 0m4.678s	
64	real 0m9.706s user 0m50.699s sys 0m3.894s	Dipped down, maybe the multiple works good.
128	real 0m15.229s user 1m3.985s sys 0m11.085s	
256	real 0m13.136s user 0m50.730s sys 0m8.198s	

3. **It might be useful to plot the above performance in a graph. What did you notice in terms of the performance? Why is the performance not linear?**

Realtime vs Threads
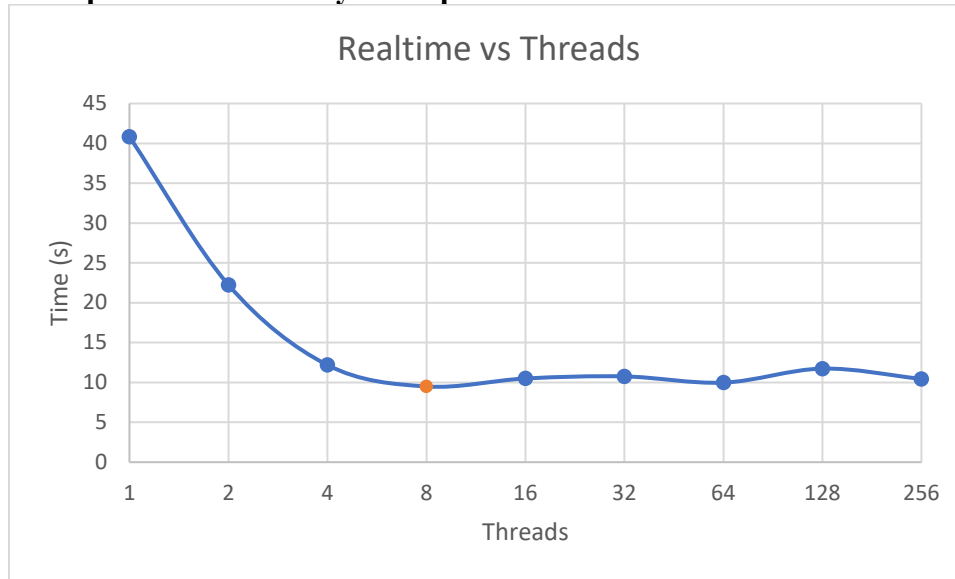
Because the number of virtual threads can only really go as fast as many logical cores to process on, I presume.

4. **How would you increase the performance of your code?**
Increasing the speed of the image decoding library? Better thread allocators with a proper parallel rayon iterators?

After completing this assignment, you should be able to answer the following questions:
1. **What is the limiting factor in programs with a massive number of threads?**
Writing to the output file all at once, competing for processor time, scheduling for actual processing.
2. **What are the protection mechanisms for threads and shared memory?**
You have mutexes, and semaphores generally. You also have shmget/alloc and friends for shared memory between processes.
3. **How to manage many threads without closing them and restarting them from scratch.**
You could implement worker threads that get jobs deployed to them instead of this current method, which would be much better.