Kaushik Mahadevan

Prof. Kim

CSIT345

23 March 2023

HW 2



* Input and output involve reading and writing to a file. Because accessing a resource from multiple threads multiple threads leads to a concurrency problem (critical section), and because processing files is not easily parallelized, it makes the most sense to use 1 thread for each operation.
* For the CPU bound portion of the program, the work can be split among each of the 4 cores. With 1 to 1 mapping, 1 thread can be assigned to each processor core and so 4 threads should be used for this problem.

1. 1. There are 6 unique processes created. First fork -> 2 processes total, 2nd fork -> 3 processes total, 3rd fork -> 6 processes total.
   2. One thread created for each process after the 2nd call to fork(), so two threads are created. One school of thought is that each process spins up its own thread, so there are a total of 6 threads per process + 2 newly created threads, so it could be 8, but I don’t think that’s what this question is asking for.