Nick Wu

Intro to OS

Lab 7

1. C program attached
2. Threads are typically more efficient and lighter than process-based concurrency. The communication and synchronization between threads allow for easy sharing of memory. Threads can also be managed and created within a single process. The main issue with threads is the possibility of data corruption and race conditions due to the characteristic of shared memory. Thus, synchronization mechanisms like mutexes and semaphores need to be used to coordinate the threads access to shared memory.  
     
   On the other hand, using processes and pipes for concurrency provides certain advantages. Each process has its own memory space, eliminating the complication of explicit synchronization mechanisms. Pipes allow processes to communicate through an isolated channel which means one process crashing does not affect others. However, creating and managing processes can be more resource-intensive compared to threads. Processes also have limited sharing capabilities, requiring explicit data transfer between them.