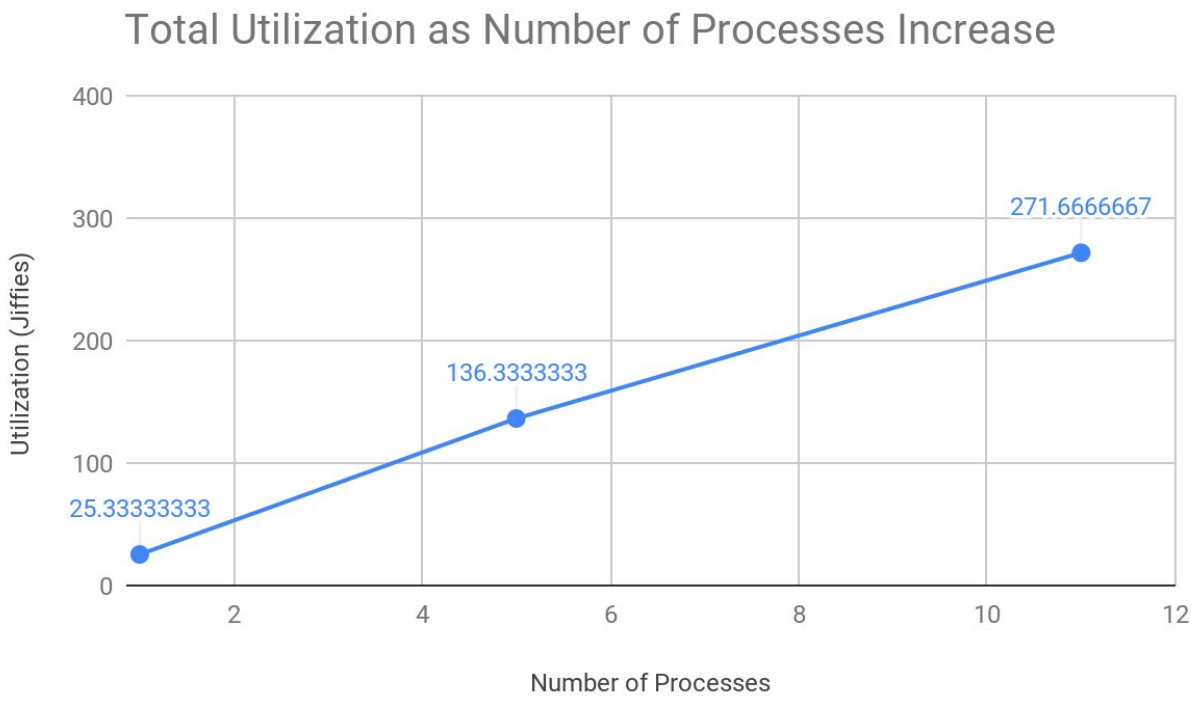
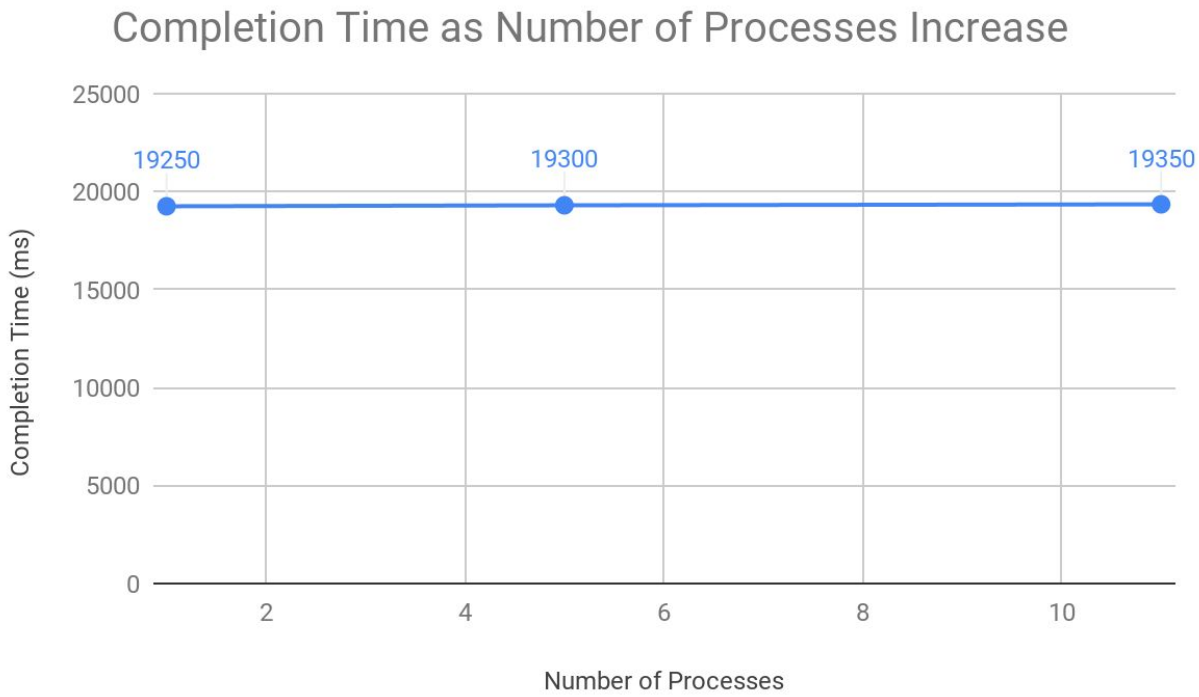
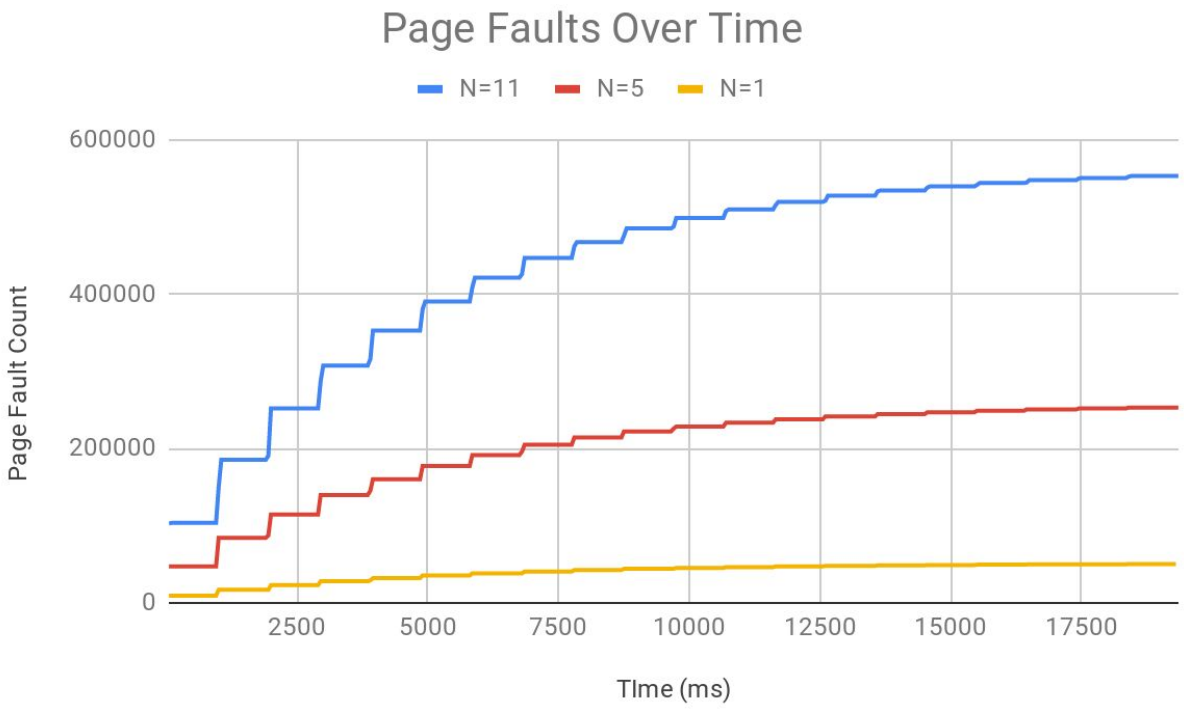
Case Study 2: Multiprogramming







In this case study, we explore the computation time, application finish time, and page fault rate as multiple processes are ran. To record utilization time, the processes were ran 3 times to record their respective utilization times, and then averaged.

As the number of parallel processes increased, computation time barely changed. It was steady around 19.3 seconds. Because of this, CPU utilization was considered to be utime + stime, as all 3 processes ran for a similar total amount of time. The fault amounts varied linearly with the number of processes running, which is to be expected, as more running processes would incur more page faults.

As the number of processes running increased, the CPU utilization increased with a slightly diminishing slope. This is to be expected, as the amount of parallelism of tasks requiring memory increases on a CPU, the CPU utilization will raise with a diminishing slope, and then fall due to thrashing. Here, we were able to document the initial rising slope. In the worker thread, the thread sleeps 1 second after making memory requests, to prevent too many requests from being made to the memory management unit. If that safety sleep was removed, or the number of tasks was further increased, it may be possible to witness the drop in CPU utilization due to thrashing.