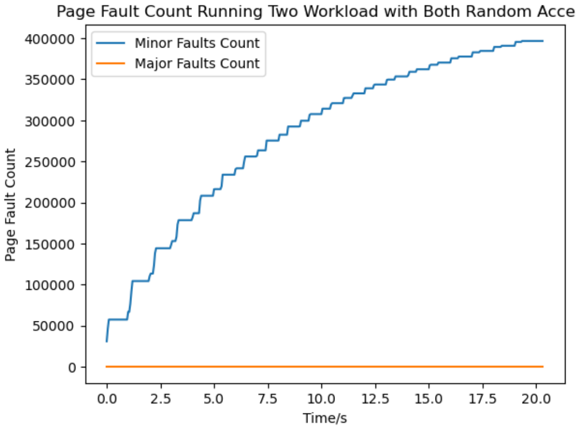
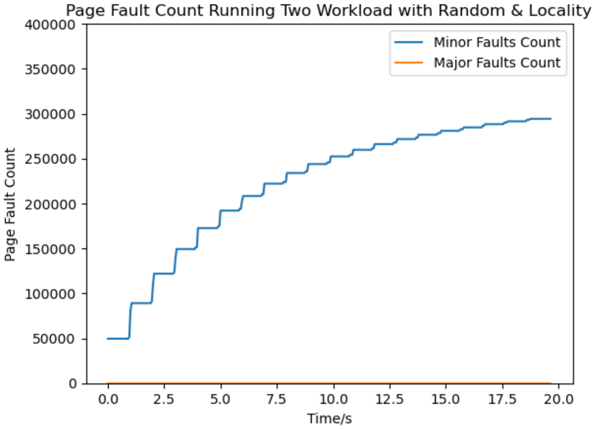
5.1 Case Study 1: Thrasing and Locality.

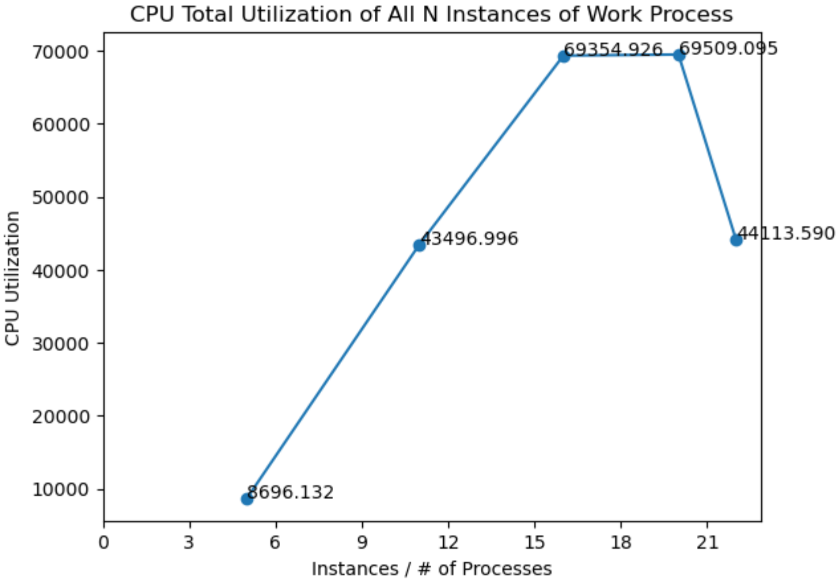


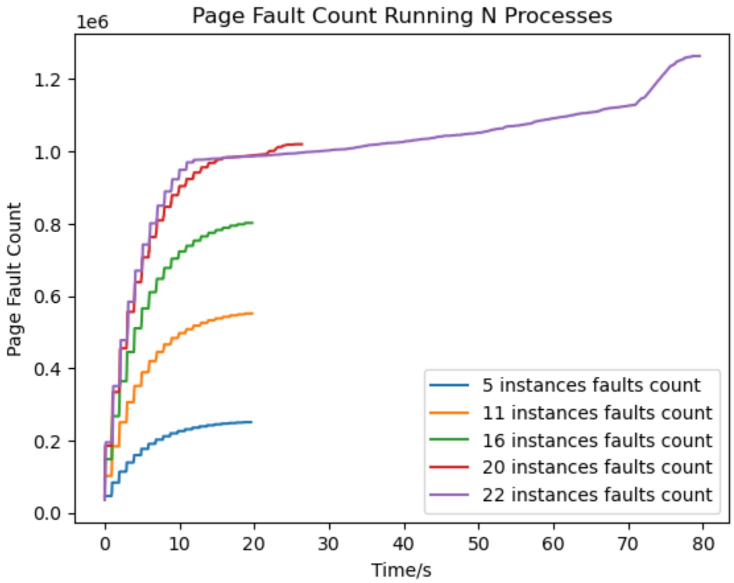


From the two graphs, it's apparent that there are no major page faults in either case. Regarding minor page faults, the first case, which involves two random access processes, exhibits more minor page faults than the second one, which involves one random and one locality-based access. This indicates that random access patterns result in more page faults because the page fault replacement policy is less effective at reducing page faults when the access pattern is random.

The completion time is essentially the same for both cases, around 20 seconds, because there are no major page faults to necessitate disk I/O.

5.2 Case Study 2. Multiprogramming





From these two graphs, it can be seen that before n is less than or equal to 16, the CPU utilization grows almost linearly. During this phase, there are only minor page faults and no major page faults. The completion time is consistent at around 20 seconds. However, afterwards, when n increases to 20 and 22, the CPU utilization does not continue to increase linearly but instead decreases, indicating that thrashing occurs. Major page faults appear at this point, and the CPU is idle as the page fault handler is caught in a continuous cycle of replacing pages. The completion time is no longer 20 seconds but has increased.