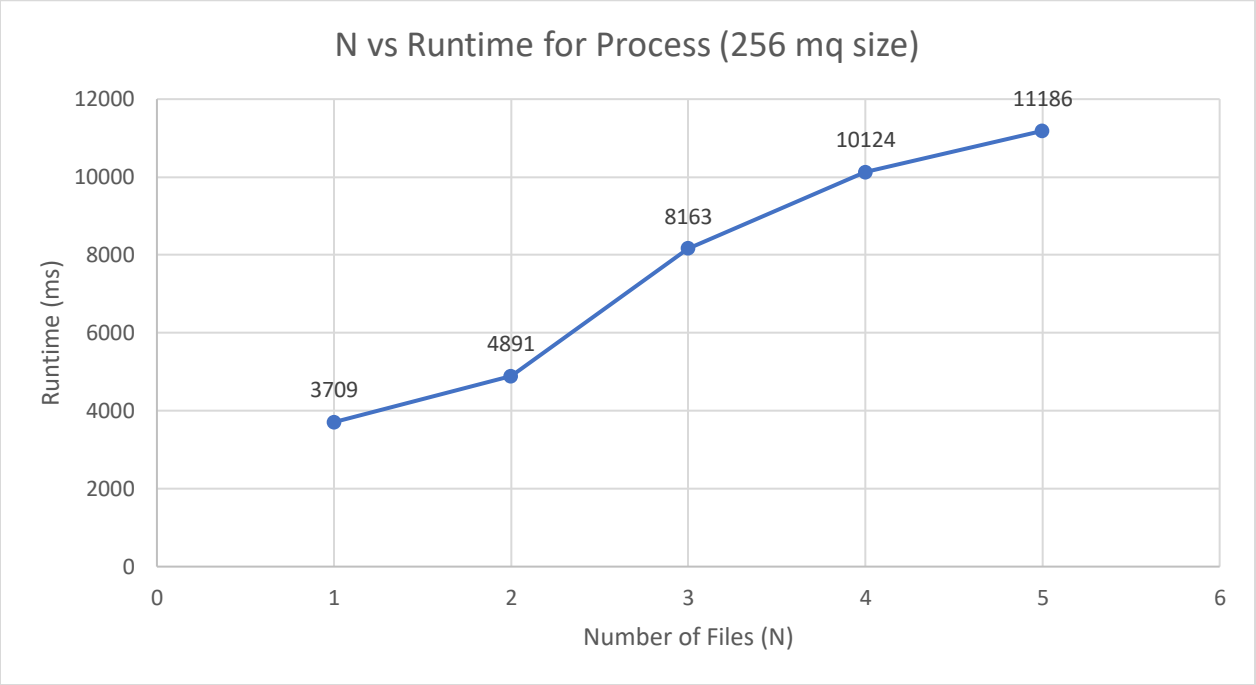
CS-342
Project 1 Report

Mehmet Berk Türkçapar – Ege Ergül
21902570 - 21902240
Section: 1

- Runtime Values for Processes

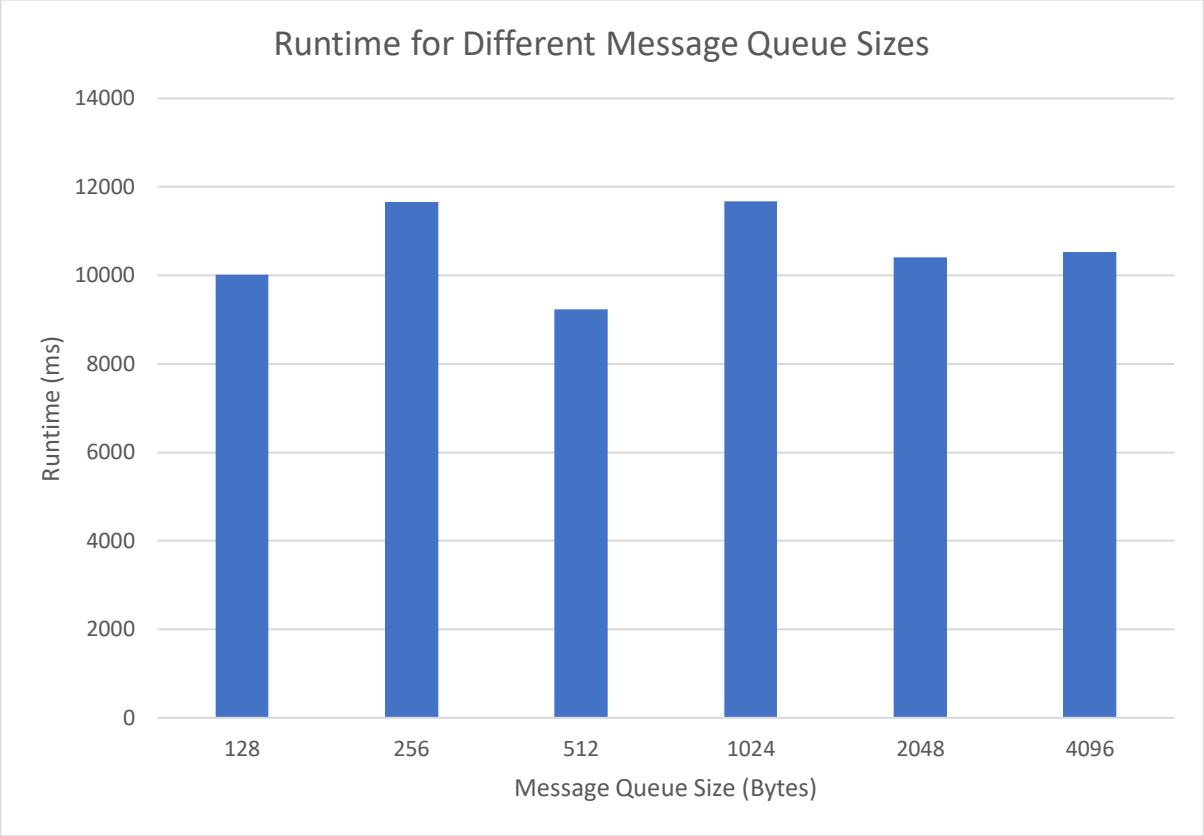
For Different Values of N:

We have experimented for different values of N (number of files). We have executed the program for N = 1, 2, 3, 4, 5. Below graph shows the relation between different values of N and the runtime values for each of these values. These values are obtained from the execution of pword program. For this experiment the message queue size was fixed to 256 bytes.



As it can be seen from the graph as the number of files increased the runtime also increased. As N increases there are more child processes created which means there are more data to be processed. Processes do not use shared memory. For communication they utilize message queues. Since each of these processes use the same message queue, they need to wait for each other to use it. Thus, it is expected that as. the number of files increase the runtime also increase.

For Different Values of Message Queue Size:

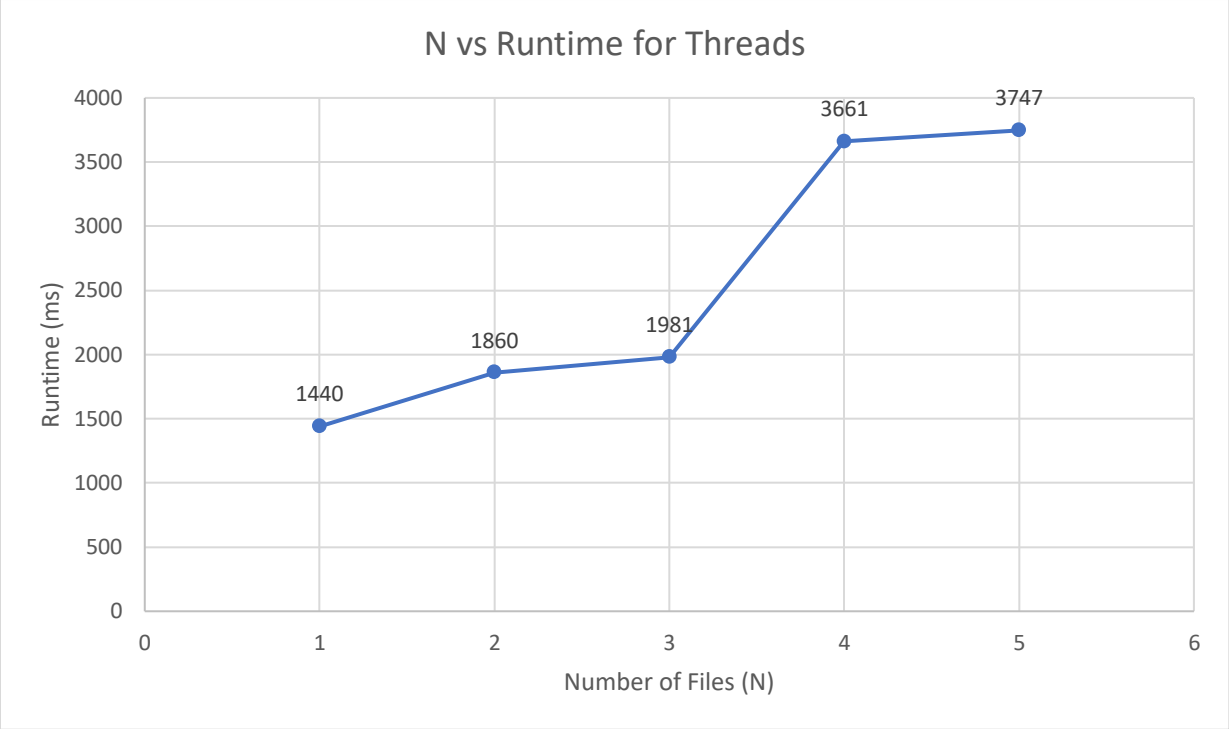


We have experimented for different values of message queue size and fixed the number of files (N) to 3. We have executed the program for 128, 256, 512, 1024, 2048 and 4096 bytes. As it can be seen from the graph the execution time of pword did not change significantly for different message queue sizes. Also, it can be concluded that there is no particular relation between message queue size and execution time.

- Runtime Values for Threads

For Different Values of N:

We have experimented for different values of N (number of files). We have executed the program for N = 1, 2, 3, 4, 5. Below graph shows the relation between different values of N and the runtime values for each of these values. These values are obtained from the execution of tword program.



As it can be seen from the graph as N increased the execution time of the program also increased. This is because each thread uses the same global variable for holding the frequency data. Thus, each thread needs to wait each other to prevent any data errors which may occur as a result of multiple threads trying to change the same part of the memory.

- Threads vs Processes

As it can be seen from the obtained data, threads perform much faster compared to processes. For example, for N = 5 pword's execution time is 11186 milliseconds whereas tword's execution time is 3747 milliseconds. The reason behind this difference is that threads have shared memory while processes don't. Each time a new child process is created the memory block of its parent is copied. However, for threads this is not necessary since they share the same memory.