**Mohamed Ahmed**

**5/8/2024**

**Assignment #3**

**CSC139: Operating System Principles**

**Second Assignment**

**Thread Synchronization**

I tested it on the school computer(ecs-pa-coding1) The computer has 4 physical cores.

**Table of Results**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test No. | Array Size | Threads | Zero Index | Sequential | Parent Waits | Parent Continuous Check | Parent Semaphores | Product |
| 1. | 100M | 2 | 50M + 1 | 433 ms | 263ms | 302 ms | 329 ms | 0 |
| 2. | 100M | 4 | 75M + 1 | 586 ms | 285 ms | 315 ms | 259 ms | 0 |
| 3. | 100M | 8 | 88M | 548 ms | 179 ms | 153 ms | 180 ms | 0 |
| 4. | 100M | 2 | -1 | 68 ms | 115 ms | 122 ms | 134 ms | 4149 |
| 5. | 100M | 4 | -1 | 514ms | 204ms | 222 ms | 202 ms | 4149 |
| 6. | 100M | 8 | -1 | 459 ms | 151 ms | 158 ms | 158 ms | 4149 |

Findings:

The results are consistent in showing that increasing the number of threads generally improves the performance for the Parent Waits and Parent Continuous Check methods, with execution times decreasing notably. In cases where the zero index is **-1** (indicating not found), execution time is significantly lower than when the zero index is found.

A screen shot of a computer screen

Description automatically generated