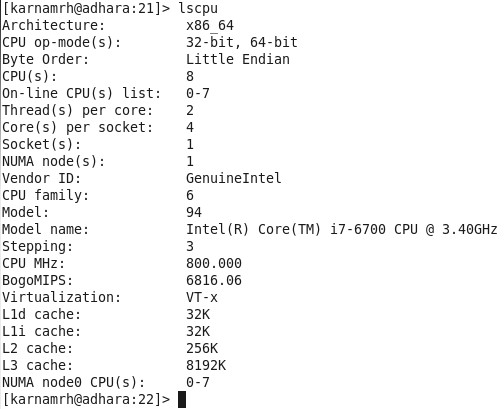
**ASSIGNMENT\_2**

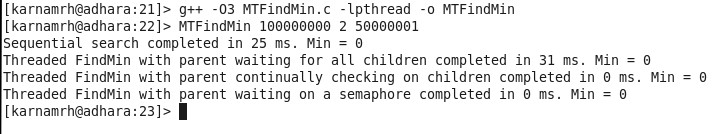
**Name: HARI CHANDANA KARNAM**

**Command to know the system specifications: lscpu**



**Test Case 1:**

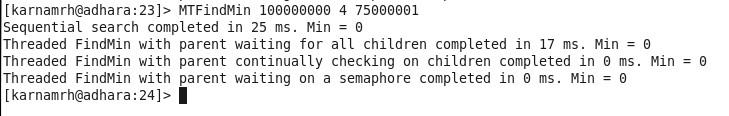
Array size = 100M, T=2, index for zero =50M+1



|  |  |  |  |
| --- | --- | --- | --- |
| **Sequential Search** | **Parent waiting for all children** | **Parent Continuously checking on all**  **children** | **Parent waiting on**  **semaphore** |
| 25ms | 31ms | 0ms | 0ms |

**Test Case 2:**

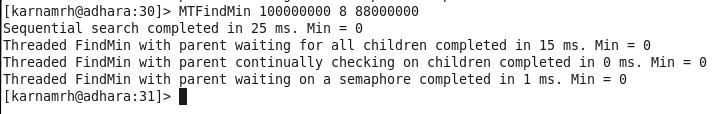
Array size = 100M, T=4, index for zero =75M+1



|  |  |  |  |
| --- | --- | --- | --- |
| **Sequential Search** | **Parent waiting for all children** | **Parent Continuously checking on all**  **children** | **Parent waiting on**  **semaphore** |
| 25ms | 17ms | 0ms | 0ms |

**Test Case 3:**

Array size = 100M, T=8, index for zero =88M

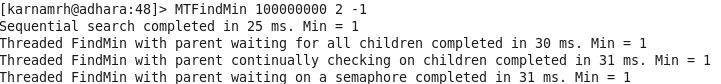


|  |  |  |  |
| --- | --- | --- | --- |
| **Sequential Search** | **Parent waiting for all children** | **Parent Continuously checking on all**  **children** | **Parent waiting on**  **Semaphore** |
| 25ms | 15ms | 0ms | 1ms |

**Test Case 4:**

Array size = 100M, T=2, index for zero =-1 (no zero)

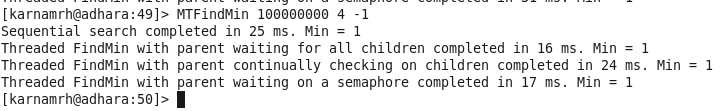
**Proves that there is no use even if there are max cores but min threads.**



|  |  |  |  |
| --- | --- | --- | --- |
| **Sequential Search** | **Parent waiting for all children** | **Parent Continuously checking on all**  **children** | **Parent waiting on**  **Semaphore** |
| 25ms | 30ms | 31ms | 31ms |

**Test Case 5:**

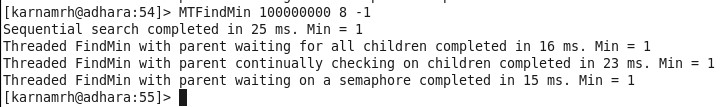
Array size = 100M, T=4, index for zero =-1 (no zero)



|  |  |  |  |
| --- | --- | --- | --- |
| **Sequential Search** | **Parent waiting for all children** | **Parent Continuously checking on all**  **children** | **Parent waiting on**  **semaphore** |
| 25ms | 16ms | 24ms | 17ms |

**Test Case 6:**

Array size = 100M, T=8, index for zero =-1 (no zero)



|  |  |  |  |
| --- | --- | --- | --- |
| **Sequential Search** | **Parent waiting for all children** | **Parent Continuously checking on all**  **children** | **Parent waiting on**  **Semaphore** |
| 25ms | 16ms | 23ms | 15ms |

**Conclusion:**

**Sequential Search without multithreading:(Worst Case):**

It takes same time for all the test cases and takes maximum time to compute the minimum.

**Parent waiting for all children:**

Performance increases when the number of threads increased. But even if one of the children is done finding zero, parent still waits for other children to complete the task.

**Parent continuously checking on all children:**

Performance is good when there is zero in array because it terminates all the children if one of the children finds zero. But performance becomes bad when there is no zero in the array because parent wastes CPU cycles while checking the child thread continuously.

**Parent Waiting on Semaphore:(Best Case)**

Parent has been put on waiting state till the threads find the minimum value. So, Parent does not waste CPU cycles. From the first three cases we can see that this method took almost 0ms to find zero.