



BILKENT UNIVERSITY

ENGINEERING FACULTY

DEPARTMENT OF COMPUTER ENGINEERING

CS 342 Project 2 Report

Mehmet Ege Acıcan

21602186

Table of Contents

Part1: Description, Errors and RunTime Table	2
Part2:Description,Errors and RunTime Table	3

1- Part1: Description, Errors and Runtime Table

On part 1, it has been asked to process N input files and insert the K maximum numbers on a Binary Search Tree and then write the results. This has been asked to be done via threads and Synchronization. In Part A, Mutex locks has been used.

Without synchronization, the threads can enter their respected files at the same time and then allow the output tree to be different. However, with synchronization this problem is handled.

The following section contains the output tables where

K = tree size

N = number of worker threads.

File Size = # of numbers in the file.

The output tables are as follows:

N	Runtime(s)
1	0.004441
2	0.004205
3	0.005606
4	0.046338
5	0.055964

Figure 1: Output table based on the N (k has been hold constant $k = 1000$, filesize has been hold constant $fs = 10000$)

Conclusion: The relation between thread numbers and Runtime is linear

k	Runtime(s)
100	0.001515
500	0.002312
1000	0.004187
5000	0.019114
10000	0.036726

Figure 2:Output table based on k(N has been hold constant $n = 3$, filesize has been held constant on 10000)

Conclusion: The runtime is affected in linear order with k

File Size	Runtime(s)
1500	0.003725
2000	0.004286
3000	0.005059
5000	0.006396
10000	0.036500

Figure 3: Output table based on filesize (N has been hold constant $n = 2$ and k has been held constant on $k = 100$)

Conclusion: There is a linear relation between file size and runtime

2- Part2: Description, Errors and Runtime Table

In part B, the same thing has been asked however this it is being asked to be done with processes instead of threads and this time a shared memory space has been asked to use between the processes. In part B, semaphores has been used.

The absence of synchronization has presented the same situation as it was in part A. the processes can enter their respected files at the same time and then allow the output to be different. However, with synchronization this problem is handled.

The following section contains the output tables where

K = tree size

N = number of processes.

File Size = # of numbers in the file.

The output tables are as follows:

N	Runtime(s)
1	0.004685
2	0.004745
3	0.004895
4	0.004911
5	0.004969

Figure 3: Output table based on the N (k has been hold constant $k = 1000$)

Conclusion: There is a linear relationship between process number(N) and runtime.

K	Runtime(s)
100	0.000777
500	0.002346
1000	0.004639
5000	0.046079
10000	0.146533

Figure 4: Output table based on k (N has been hold constant $n = 1$)

Conclusion: There is a linear relationship between number K and runtime

File Size	Runtime(s)
1500	0.002774
2000	0.003425
3000	0.004994
5000	0.005287
10000	0.008745

Figure 5: Output table based on file size($N=1$, $k = 100$)

Conclusion: There is a linear relation between File Size and Runtime