



## Assignment 2

CS375 Operating System

August 15, 2020

Group Member		
No.	Student Name	Student ID
1	Mohammad falah altawil	118256
2	Ahmad nazeq quraan	108055



## Assignment 2

CS375 Operating System

August 15, 2020

### Part I - Multithreaded / Task parallelism

- a. Write a multithreaded program that calculates various statistical values for a list of numbers (try small/large/very large numbers). This program has to create three separate worker threads. One thread will determine the average of the numbers, the second will determine the maximum value, and the third will determine the minimum value. For example, suppose your list contains the Integers 90 81 78 95 79 72 and 85, The program will report the following: The average value is 82 The minimum value is 72 The maximum value is 95 The variables representing the average, minimum, and maximum values will be stored globally. The worker threads will set these values, and the parent thread will output the values once the workers have exited.

### Solution:

Code : In file named "part 1 a with thread"

This is a screen for run code .

```
<terminated> MultiThread [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (Aug 15, 2020, 9:16:13 PM – 9:16:39)
Enter the number of elements :7
Enter the elements :
90 81 78 95 79 72 85
take time with thread in durationInNano : 62904300
take time with thread in durationInMillis : 62
The Avarage Value :82.85714285714286
The Minimum Value : 72
The Maximum Value : 95
```



## Assignment 2

CS375 Operating System

August 15, 2020

### Part I - Multithreaded / Task parallelism

- b. Write a multithreaded program that outputs prime numbers. This program should work as follows: The user will run the program and will enter a number as an input. The program will then create a separate thread that outputs all the prime numbers less than or equal to the number entered by the user.

#### Solution:

Code : In file named "part 1 b with thread"

This is a screen for run code .

```
<terminated> Prime [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (Aug 15, 2020, 10:03:22 PM – 10:03:31 PM)
Prime number you choose: 20
[ 2 3 5 7 11 13 17 19

take time with thread in durationInNano : 7659900
take time with thread in durationInMillis : 7
```



## Assignment 2

CS375 Operating System

August 15, 2020

### Part I - Multithreaded / Task parallelism

- c. Solve the previous problems without using threads. Use the time function with each part (a, b and c) and show, for each one, what is the exact execution time? Is there any difference? Why? You have to answer these questions in your report.

### Solution:

For Choice a : Code : In file named "part 1 a without thread"

This is a screen for run code .

```
<terminated> MultiThread [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (Aug 15, 2020, 9:51:41 PM – 9:51:57 PM)
Enter the number of elements : 7
Enter the elements : 90 81 78 95 79 72 85
take time in durationInNano : 6200
take time in durationInMillis : 0
The Average Value :82
The Minimum Value : 72
The Maximum Value : 95
```

For Choice b : Code : In file named "part 1 b without thread"

This is a screen for run code .

```
<terminated> Prime [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (Aug 15, 2020, 10:00:35 PM – 10:00:45 PM)
Prime number you choose: 20
2 3 5 7 11 13 17 19

take time in durationInNano : 449700
take time in durationInMillis : 0
```



## Assignment 2

CS375 Operating System

August 15, 2020

### Part II – Multithreaded / Data parallelism

- a. write a multithreaded sorting program that works as follows: A list of integers is divided into Four smaller lists of equal size. Four separate threads (which we will term sorting threads) sort each sub-list using a sorting algorithm of your choice. (You may use any built in sorting function). The Four sub-lists are then merged by a Fifth thread — a merging thread — which merges the Four sub-lists into a single sorted list. Because global data are shared cross all threads, perhaps the easiest way to set up the data is to create a global array. Each sorting thread will work on one 1/4 of this array. A second global array of the same size as the unsorted integer array will also be established. The merging thread will then merge the Four sub-lists into this second array. Graphically, this program is structured similarly according to Figure 4.20 if we intend to partition the array into two parts. This programming project will require passing parameters to each of the sorting threads. In particular, it will be necessary to identify the starting index from which each thread is to begin sorting. Refer to the instructions in Project 1 for details on passing parameters to a thread. The parent thread will output the sorted array once all sorting threads have exited.

### Solution:

Code : In file named “part 2 with thread”

This is a screen for run code .

```
Problems @ Javadoc Declaration Console
<terminated> Sorting [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (Aug 15, 2020, 10:21:22 PM – 10:21:24 PM)
list numbers
2
5
8
9
12
15
18
19

take time with thread in durationInNano : 16143400
take time with thread in durationInMillis : 16
```



## Assignment 2

CS375 Operating System

August 15, 2020

### Part II – Multithreaded / Data parallelism

- b. Solve the previous problem without using threads. Use the time function with each part (a and b) and show, for each one, what is the exact execution time? Is there any difference? Why? You have to answer these questions in your report.

### Solution:

Code : In file named “part 2 without thread”

This is a screen for run code .

```
<terminated> Sorting [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (Aug 15, 2020, 11:27:02 PM – 11:27:04 PM)
list numbers not sorted
5
2
9
8
15
12
19
18
list numbers sorted
2
5
8
9
12
15
18
19

take time in durationInNano : 23345900
take time in durationInMillis : 23
```