ATILIM UNIVERSITY

Faculty of Engineering

Department of Computer Engineering

CMPE 431 – OPERATING SYSTEMS

Homework 3

Academic Year: Spring 2019-2020
Date: **02.05.2020**
Due Date: **08.05.2020**
Instructor: Assoc. Prof. Dr. Hürevren KILIÇ
Lab Assistants: Res. Asst. Ersin TİRYAKİ, Res. Asst. İbrahim TARAKCI

- Note: You need to upload your C Code source file as **name_surname_SID_hw3** to Moodle until 23:59 08.05.2020.
- Group study is not allowed. Everyone do his/her homework as an individual.
- **Late homework will NOT be graded.**
- Cheating will NOT be tolerated. If any case of cheating is detected, at any time, **you will get ZERO from your homework.**

In this assignment, you will solve a **multithreaded sorting problem** by implementing a C code. You need to sort a list which includes these following integer values: **{7,12,19,3,18,4,2,6,15,8,13,1}** by **using threads**. First of all, this list of integers is going to be divided into 4 smaller lists of equal size. Four separate sorter threads sort each sublist using **Insertion Sort Algrithm,** so you need **4 sorter threads.** After you sort 4 sublists; you need to merge **first and second sublists by using first merger thread**, then you need to **merge third and fourth sublists by using second merger thread**. Finally, **third merger thread** merges **first half of list** (which is merged by first merger thread) **and second half of list** (which is merged by second merger thread). So you need **3 merger threads**.

Sample Output is following:

```
Before Sorting: 7  12  19  3  18  4  2  6  15  8  13  1
After 1. Sorting Thread Work: 7  12  19
After 2. Sorting Thread Work: 3  4  18
After 3. Sorting Thread Work: 2  6  15
After 4. Sorting Thread Work: 1  8  13
After first Merging Thread Work: 3  4  7  12  18  19
After second Merging Thread Work: 1  2  6  8  13  15
After third Merging Thread Work: 1  2  3  4  6  7  8  12  13  15  18  19

--------------------------------
Process exited after 0.2274 seconds with return value 0
Press any key to continue . . .
```

Pseudocode of Insertion Sort Algorithm is following:

a is an array of size N =>

```
for i from 1 to N
   key = a[i]
   j = i - 1
   while j >= 0 and a[j] > key
      a[j+1] = a[j]
      j = j - 1
   a[j+1] = key
next i
```

Graphical representation of sorting is following: