CPRE 388 – Embedded Systems II: Mobile Platforms
FALL 2017, HOMEWORK 4
100 Points

Date Assigned – Oct 18, 2017
Date Due – Nov 1, 2017, 11.00 PM

Submission – Through Blackboard ONLY as typed WORD or PDF for theory questions. Please write in your own words if you use one of the resources below. Do not cut and paste from any website or pdf document. Upload zipped project folder for coding exercise. Individual work only. Do not share code.

The below are only basic specifications. You are free to improve the app with creative additions.

1. ThreadPoolExecutor is a utility class which reduces the effort required for creation, managing, and deletion of multiple threads which may run parallel instances of the same task. For example, multiple thread instances may run averaging filter on 8x8 blocks of a large BMP image file.

   ThreadPoolExecutor (Reference b.) class can be configured with different options such as thread pool size, keep-alive times, queuing strategies etc.

   (i) Design an *intuitive* UI which provides selectable options to the user to configure the ThreadPoolExecutor class. (initialization of the class). Your UI will be graded on the quality, perceived user experience (such as font sizes, spacing, good use of screen space, use of variety of UI widgets etc.). Using Preference fragments may be a good option for some portions of the UI. [45]

   Below is an example reference implementation. However, you may NOT reproduce the same UI or use the implementation in the resource below for this homework.

   http://androidahead.com/2017/02/23/threadpoolexecutor-on-android-a-practical-example/

   (ii) Develop a backend for the UI, by creating a controller (Activity) which will create and launch threads using the ThreadPoolExecutor class configuration selected by the user in Part a. Your tasks (Runnable to be executed in the thread) may be simple for loops to keep the thread busy (choose any appropriate number of loop iterations). Update the UI, periodically, to show the total number of tasks which have been completed, number of threads actively executing task and any other dynamic state as needed. [55]

   You may submit all of the above as a single working app.

References

a. https://developer.android.com/training/multiple-threads/index.html
b. https://developer.android.com/reference/java/util/concurrent/ThreadPoolExecutor.html
c. https://developer.android.com/training/multiple-threads/create-threadpool.html
d. https://developer.android.com/training/multiple-threads/run-code.html