Mark Piland

CST235

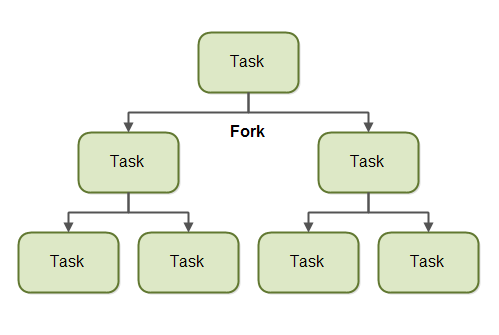
March 19 2018

Completed in collaboration with Robbie Evans III

**Week 6 CLC Assignment**

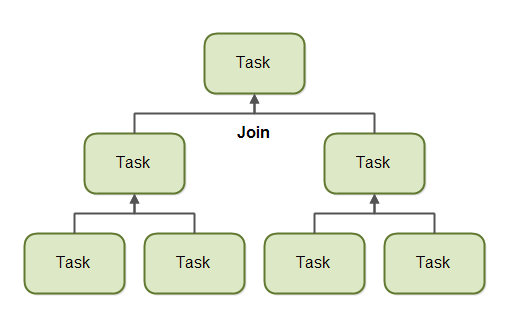
**Question 1. Explain Fork and Join in Java**

The fork and join is a framework that was presented in Java 7. The framework provides tools to help speed up parallel processing by attempting to use all available processor cores on the computer which is accomplished **through a divide and conquer approach** (Baeldung, 2017). First, the framework **“forks”** the task, recursively breaking the task into smaller independent subtasks until they are simple enough to be executed asynchronously or concurrently as illustrated in the diagram bellow (Jenkov, 2015).



In addition, when the task splits itself into subtasks, each subtask can be executed in parallel by different processors, or on different threads on the same Central Processing Unit (CPU). It is important to note that a task splits itself up into subtasks if the work the task is large. “There is an overhead to splitting up a task into subtasks, so for small amounts of work this overhead may be greater than the speedup achieved by executing subtasks concurrently” (Jenkov, 2015).

After forking, the **“join”** part begins, in which the results of all subtasks are recursively joined (merged) into a single result, or in the case of a task which returns void, the program simply waits until every subtask is executed (Baeldung, 2017). The diagram bellow illustrates the joining process after forking.



In addition, for effective parallel execution of the above-mentioned techniques, the fork/join framework uses a pool of threads called the *ForkJoinPool (*designed to work well with fork-and-join task splitting. The ForkJoinPool located in the *java.util.concurrent* package)which manages worker threads of type *ForkJoinWorkerThread*. The *worker* threads executes only one task at a time, and the *ForkJoinPool* does not create a separate thread for each single subtask. Therefore, each thread in the pool has its own double-ended queue which stores tasks.

**References**

B. (2017, July 20). Guide to the Fork/Join Framework in Java. Retrieved March 20, 2018, from <http://www.baeldung.com/java-fork-join>

Jenkov, J. (2015, March 02). Java Fork and Join using ForkJoinPool. Retrieved March 20, 2018, from <http://tutorials.jenkov.com/java-util-concurrent/java-fork-and-join-forkjoinpool.html>