Lab #3: Java Threads

Overview

The purpose of this lab is to introduce threading in Java and to analyze the performance characteristics of a divide-and-conquer approach to problem decomposition.

Part 1: Quicksort

Write a single-threaded integer quicksort routine (you may copy this code from any reliable resource) in Java.

Part 2: Parallel Quicksort

Write a multi-threaded integer quicksort routine (do not copy this from any resource).

At each call, the invoking thread will partition the subarray, spawn a new thread to recurse on one portion, and recurse on the other portion.

At a specified watermark (this should be easily modifiable), the algorithm will sort the subarray using insertion or selection sort (you will need to write this routine, but are free to view reliable resources; just be sure that the routine you use supports sorting a subarray) instead of further partitioning the array.

Part 3: Analysis

Compare the performance of the two sorts on large arrays of random integers. Be sure to seed the random number generating consistently so that each sort is using the same array of values.

Adjust the watermark value to determine the effects of task granularity on the multi-threaded sort.

As part of your lab submission, you must write a paper detailing these experiments and the results of tweaking the array size and the watermark value.

Handin

Submit, using `handin`, your source code and paper to `458_lab3` on vogon.