Project - 3

Design Document

Vignesh Kulothungan ([vxk111430@utdallas.edu](mailto:vxk111430@utdallas.edu))

Date: 18th March, 2014

**Algorithm:**

**Main Process (Thread 0):**

* Reads input from command line.
* Opens “semainit.dat” file and initializes the semaphores with the values read from the file.
* Opens the “input.dat” file (received via command line) and reads the first number and stores in “N”.
* Then reads “N” numbers from the file and stores it in the input array.
* Then creates (N/2 – 1) threads. The threads created will all execute the “sortArray” routine. The thread number is passed as an argument to each thread.
* The main process (thread 0) will execute the same instructions in “sortArray” and also implements the semaphore implementation.
* Four semaphores are used to achieve synchronization here.
* Two semaphores are used before start of each phase and two semaphores are used after the end of each phase to achieve synchronization between threads.
* All threads will do a sem\_post on semaphore “semStrCnt” and will wait on semaphore “semStr”. The main process will check the semaphore value of “semStrCnt”, when the value is equal to (N/2 – 1), it will do a sem\_post on semaphore “semStr” (N/2-1) times to unblock all threads waiting at the start of each phase.
* All threads will execute the algorithm in parallel, reach the end and will do sem\_post on semaphore “semEndCnt” and wait of semaphore “semEnd”.
* Main thread will reach end, check the semaphore value of “semEndCnt”. When it is equal to (N/2-1), it will do sem\_post on semaphore “semEnd” (N/2-1) times to unblock all threads waiting at the end of the phase.
* This ensures all threads start and end at the same time.
* After executing all log2(N) stages and log2(N) phases, the main thread does a “pthread\_join” and waits for all (N/2-1) threads to exit. When all threads exit, parent will print the sorted array.
* Main thread will read the next number from the “input.dat” file, if it is zero, it exits. Else, repeats all steps.

**Threads (N/2 -1):**

* All threads execute the “sortArray” routine. Threads are differentiated with the thread number sent by the Main process as input argument.
* Before starting each phase, all threads will sempost on semaphore “semStrCnt” and will wait on “semStr” semaphore. They all will be unblocked by the Main thread, once all threads reach this start point.
* Before exiting each phase, all threas will sempost on semaphore “semEndCnt” and will wait on “semEnd” semaphore. They all will be unblocked by the Main thread, once all threads reach this exit point.
* If the mode is “-o” (Observation mode), all threads will print the array at the end of each phase, including the Main Thread.

**Semaphores:**

* Number of semaphores used is 4.
* Two used at the start of each phase and two at the end of each phase.
* All semaphores are initialized to value “0”.