Threads in C
Lab exercises

EPL222 – Lab5

Lab exercise 1

- Write a program that sums a large array of integers using multiple threads
  - The size of the array and the number of threads should be given as command line arguments
  - The array size should be perfectly divided by the number of threads (ArraySize % #Threads =0)
- A global sum should be maintained
  - Whenever a thread finishes summing up its portion of the array it should update the global sum
  - The program should wait for the threads to do their job!

Lab exercise 2

- Write a program that “counts”
  - You must have at least two threads that increment a “count” variable in a loop
    - Each thread will continue until it loops TCOUNT times
    - The number of threads as well as TCOUNT should be command line arguments
    - In each loop each thread will print its id and the new count
  - You must have another thread that watches the value of “count” and when “count” reaches a predefined limit it modifies the count by adding 1000
  - All threads should print when they call a synchronization function (lock, unlock, wait or signal)
  - The limit should be defined as:
    (int) TCOUNT * NoTheads * 0.6
  - The main program prints the final value of count.