CST 8244, Winter 2012, Lab 2

Date: Monday 16 January 2012 **Due**: Wednesday 18 January 2012

Objective: To create threads and read data from a file in each thread

Submission: Demonstrate a working program on 23 January 2012. Email all source code, log file, makefile and output of the program to terais@algonquincollege.com before due date. Subject on email should be "`CST 8244, W12, Lab 2`". It is important that you have the subject line correctly typed, your submission could be lost if it is incorrect. Marks are awarded for correct submission guidelines.

Documentation: Document your code with the header and footer provided earlier in a lab. Document all segments of your code where the required feature is implemented; for instance, reading data, computing results, writing data to file, advancing the pointer.

Programming Standard: Each function should perform only one task. As a guideline a function should be less than 25 lines of code.

Preparation:
Study the code in the file `mutex.c` in the thread project with Momentics.
Read QNX help on `pthread_create, pthread_join.` Study sample code in `pthead_create.` Use QNX help, it is an excellent source of information and code samples.

Requirements:
Create a project, call it "`Arith-Lab2`". See instructions on creating a project.

Create four threads in the process. Call each thread `add, subs, mult` and `div.`

In each thread open the data file `arith_data.txt`. It contains two integers on each line separated by a comma, there are a total of 1000 lines. Note: In QNX it does not matter if the file is binary or text. First thread adds the two numbers, accumulates the sum and computes average of all the sum. The second thread subtracts the two numbers, accumulates the difference and computes the average. The other two threads multiply and divide the numbers, accumulate the result and computes the average. Averages

and quotients are computed upto four decimal places. After you create a thread, for example the add thread, call `delay(1000);` Wait for each thread to complete, use `pthread_join()`. Display the results on the screen and write them to a log file, as suggested below. Call the log file `Lab2.log`
Sample.

```
Total of sum:
Average of sum:
Total of difference:
Average of sum:
Total of product:
Average of product:
Total of quotient:
Average of quotient:
```

Creating a project in QNX Momentics IDE.

In QNX Momentics IDE. Select. File->New->Project
In new project wizard under QNX (Not C/C++),
Select QNX C Project
Give a Project name.
Type: Application (default). Click Next
IDE shows an error, it needs a target. This is how you choose a target.
Choose "Build Variants" Tab, Select x86 Little Endian
You should now have skeleton code, with a make file. Compile and run the code and see it working.