# Programming Assignment 1: Due Monday, Sept. 19

*To be done on water.eng.utah.edu (you all have accounts – passwords available if your CS account doesn't work)*

1. Write an average of a set of numbers in OpenMP for a problem size and data set to be provided. Use a block data distribution.

2. Write the same computation in Pthreads.

Report your results in a separate README file.

- What is the parallel speedup of your code? To compute parallel speedup, you will need to time the execution of both the sequential and parallel code, and report
  speedup = Time(seq) / Time (parallel)
- If your code does not speed up, you will need to adjust the parallelism granularity, the amount of work each processor does between synchronization points.
- Report results for two different numbers of threads.

Extra credit: Rewrite both codes using a cyclic distribution

# Programming Assignment 1, cont.

- A test harness is provided in avg-test-harness.c that provides a sequential average, validation, speedup timing and substantial instructions on what you need to do to complete the assignment.
- Here are the key points:
  - You'll need to write the parallel code, and the things needed to support that. Read the top of the file, and search for "TODO".
  - Compile w/ OpenMP: cc –o avg-openmp –O3 –xopenmp avg-openmp.c
  - Compile w/ Pthreads:

    cc –o avg-pthreads –O3 avg-pthreads.c –lpthread
  - Run OpenMP version: ./avg-openmp > openmp.out
  - Run Pthreads version: ./avg-pthreads > pthreads.out
- Note that editing on water is somewhat primitive – I'm using vim. You may want to edit on a different machine and copy to water, but keep in mind that you'll need a fast edit-compile-execute path. Or you can try vim, too. ☺