CS594  Scientific Computing for Engineers
Homework 7
Due March 30, 2011

Do the exercises below on mordor8.eecs.utk.edu. You may to choose to use either Fortran or C.

1. Compile and run the Hello World example code.
   
   ```
   gcc -fopenmp omp_hello.c -o hello
   OR
   gfortran -fopenmp omp_hello.f -o hello
   ```
   hello

   Take a moment to examine the source code and note how OpenMP directives and library routines are being used.
   a. How many threads were used? Why?

   b. Set the number of threads to 4 by means of the OMP_NUM_THREADS environment variable.
      ```
      setenv OMP_NUM_THREADS 4
      ```
      Re-run the example code and notice the output. Run several more times. Do you always get exactly the same output? Why or why not?

2. Compile and run the workshare1 example with four threads. Pipe the output through the sort utility to make it easier to see how loop iterations were scheduled across the team of threads. Run the program a few more times and view the output. Then edit the program to change the dynamic scheduling to static (leaving the chunk size at 10). Run the revised program a few times and view the output.

   a. Explain the differences between the output from the two versions of the program.

   b. Reflect on possible performance differences between dynamic and static scheduling.

3. Compile and run the workshare2 example with 16 threads. Run the program several times and observe any differences in output. How many threads do any work? Is which threads do the work deterministic or non-deterministic?
4. Compile and run the matrix multiply example with 4 threads.
   
   ```
   gcc -fopenmp omp_mm.c -o matmult
   or
   gfortran -fopenmp omp_mm.f -o matmult
   ```

   matmult

   The output shows which thread did each iteration and the final matrix result. Now run the program again, but sort the output to clearly see which threads execute which iterations.

   ```
   matmult | sort | grep Thread
   ```

   Explain what causes the particular distribution of loop iterations.

5. Modify the Hello program to obtain information about your OpenMP environment. Using the appropriate OpenMP routines/functions, have the master thread query and print the following:
   - The number of processors available
   - The number of threads being used
   - The maximum number of threads available
   - If you are in a parallel region
   - If dynamic threads are enabled
   - If nested parallelism is supported

6. Each of the provided programs omp_bug[n].c,f contains a bug. For each of the four buggy programs, find the problem and fix it to produce a correctly working program.