Operating Systems Fall 2013
Homework Problem 1

1. Assuming that the execution of threads on your computer is time-sliced[1], what is the cost of this time-slicing in terms of the time required for the operating system to perform the time-slicing? To find out, write a procedure that does work performing a lengthy computation, such as the repeated addition shown in the function *tfunc* listed below. Then compare the time taken by a single thread doing all the work with the time taken by ten threads each doing one-tenth of the work.

    a. What can you say about the cost of time-slicing?

    b. How does the number of processors on your computer affect your result? (to answer this you may have to either analyze what happens if only one CPU is being used for all of the threads, or experiment on a computer with fewer, or more, CPUs).

    To implement a solution, develop a program that will take as arguments the number of threads executing simultaneously, and the number of iterations performed in *tfunc* for each thread. You should try this program with the number of threads equal to the number of processors on your computer. You should then let the number of threads equal to 10 or 20 times the number of processors. The number of iterations should be very large so that the program will run for 10 or more seconds.

    Discuss and interpret your observations with respect to the questions above.

```
void *tfunc(void *arg) {
   int n = (int)arg;
   int i;
   int j;
   volatile int x;

   for (i=0; i<n; i++) {
         /* this loop is the "work" that each thread does */
         x = 0;
         for (j=0; j<1000; j++)
               x = x+j;
   }
   return(0);
}
```

[1] What does "timed-sliced" mean with respect to thread execution?